Title
Developing an Adaptive Strategy for Connected Eco-Driving Under Uncertain Traffic and Signal Conditions

Authors
Hao, Peng
Wei, Zhensong
Bai, Zhengwei
et al.

Publication Date
2020

DOI
10.7922/G2F18WZ1

Data Availability
The data associated with this publication are available at: https://doi.org/10.6086/D11H3P

Developing an Adaptive Strategy for Connected Eco-Driving Under Uncertain Traffic and Signal Conditions

January 2020

A Research Report from the National Center for Sustainable Transportation

Peng Hao, University of California, Riverside
Zhensong Wei, University of California, Riverside
Zhengwei Bai, University of California, Riverside
Matthew J. Barth, University of California, Riverside

National Center for Sustainable Transportation
UCR College of Engineering- Center for Environmental Research & Technology

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. NCST-UCR-RR-20-03	**2. Government Accession No.** N/A	**3. Recipient's Catalog No.** N/A
4. Title and Subtitle Developing an Adaptive Strategy for Connected Eco-Driving Under Uncertain Traffic and Signal Conditions		**5. Report Date** January 2020 **6. Performing Organization Code** N/A
7. Author(s) Peng Hao, PhD, https://orcid.org/0000-0001-5864-7358; Zhensong Wei, https://orcid.org/0000-0003-3523-5689 Zhengwei Bai, https://orcid.org/0000-0002-4867-021X Matthew J. Barth, PhD, https://orcid.org/0000-0002-4735-5859		**8. Performing Organization Report No.** N/A
9. Performing Organization Name and Address University of California, Riverside Bourns College of Engineering – Center for Environmental Research & Technology 1084 Columbia Avenue Riverside, CA 92507		**10. Work Unit No.** N/A **11. Contract or Grant No.** USDOT Grant 69A3551747114
12. Sponsoring Agency Name and Address U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology 1200 New Jersey Avenue, SE, Washington, DC 20590		**13. Type of Report and Period Covered** Final Report (July 2018 – June 2019) **14. Sponsoring Agency Code** USDOT OST-R
15. Supplementary Notes DOI: https://doi.org/10.7922/G2F18WZ1 Dataset DOI: https://doi.org/10.6086/D11H3P		

16. Abstract

The Eco-Approach and Departure (EAD) application has been proved to be environmentally efficient for a Connected and Automated Vehicles (CAVs) system. In the real-world traffic, traffic conditions and signal timings are usually dynamic and uncertain due to mixed vehicle types, various driving behaviors and limited sensing range, which is challenging in EAD development. This research proposes an adaptive strategy for connected eco-driving towards a signalized intersection under real world conditions. Stochastic graph models are built to link the vehicle and external (e.g., traffic, signal) data and dynamic programing is applied to identify the optimal speed for each vehicle-state efficiently. From energy perspective, adaptive strategy using traffic data could double the effective sensor range in eco-driving. A hybrid reinforcement learning framework is also developed for EAD in mixed traffic condition using both short-term benefit and long-term benefit as the action reward. Micro-simulation is conducted in Unity to validate the method, showing over 20% energy saving.

17. Key Words Eco-Approach and Departure, Connected Vehicles, reinforcement learning, energy, mixed traffic	**18. Distribution Statement** No restrictions.		
19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 62	**22. Price** N/A

NCST

About the National Center for Sustainable Transportation

The National Center for Sustainable Transportation is a consortium of leading universities committed to advancing an environmentally sustainable transportation system through cutting-edge research, direct policy engagement, and education of our future leaders. Consortium members include: University of California, Davis; University of California, Riverside; University of Southern California; California State University, Long Beach; Georgia Institute of Technology; and University of Vermont. More information can be found at: ncst.ucdavis.edu.

Disclaimer

Acknowledgments

This study was funded, partially or entirely, by a grant from the National Center for Sustainable Transportation (NCST), supported by USDOT through the University Transportation Centers program. The authors would like to thank the NCST and USDOT for their support of university-based research in transportation, and especially for the funding provided in support of this project.

Developing an Adaptive Strategy for Connected Eco-Driving Under Uncertain Traffic and Signal Conditions

A National Center for Sustainable Transportation Research Report

January 2020

Peng Hao, Zhensong Wei, Zhengwei Bai, and Matthew J. Barth

Center for Environmental Research & Technology, University of California, Riverside

NCST

[page intentionally left blank]

TABLE OF CONTENTS

List of Tables

List of Figures

Developing an Adaptive Strategy for Connected Eco-Driving Under Uncertain Traffic and Signal Conditions

EXECUTIVE SUMMARY

The eco-approach and departure (EAD) application for signalized intersections has been proved to be environmentally efficient in a Connected and Automated Vehicles (CAVs) system. In the real-world traffic, traffic conditions and signal timings usually appear to be dynamic and uncertain. The traffic-related information received from sensing or communication devices is highly uncertain due to the limited sensing range and varying driving behaviors of other vehicles. Meanwhile, when the host vehicle is approaching an actuated signalized intersection which are widely deployed in the U.S., the remaining time of the current signal phase indicated by the SPaT message will be updated dynamically according to vehicle actuation. This uncertainty increases the difficulty to predict the actual queue length of the downstream intersection or actual remaining time in a phase using signal phase and timing (SPaT) information. It further brings great challenge to derive an energy efficient speed profile for vehicles to follow.

This research proposes an adaptive strategy for connected eco-driving towards a signalized intersection under real world conditions including uncertain traffic and actuated signal condition. A graph-based model is created with nodes representing dynamic states of the host vehicle (distance to intersection and current speed) and indicator of queue status or signal status and directed edges with weight representing expected energy consumption between two connected states. Then a dynamic programing approach is applied to identify the optimal speed for each vehicle-queue-signal state iteratively from downstream to the upstream. The uncertainty can be addressed by formulating stochastic models when describing the transition of queue-signal state. For uncertain traffic conditions, numerical simulation results show an average energy saving of 9%. It also indicates that energy consumption of a vehicle equipped with adaptive EAD strategy and a 100m-range sensor is equivalent to a vehicle with conventional EAD strategy and a 190m-range sensor. To some extent, the proposed strategy could double the effective detection range in eco-driving. For the actuated signals, the numerical simulations with real world SPaT data show that the proposed method is robust and adaptive to varying signal conditions, and achieves 40% energy savings when the vehicle arrives in the red time, and 8.5% energy savings when the vehicle arrives in the green time compared to other baseline methods. We also proposed a hybrid reinforcement learning (HRL) framework to develop eco-driving strategies using onboard sensor only when the historical traffic data are not available. Microsimulation experiments in Unity shows that the proposed HRL-based ego-vehicle can traverse trough a signalized intersection with eco-driving strategy under mixed traffic conditions, reducing 12%-47% energy consumption comparing with baseline methods and 1.2%-6.9% travel time. The proposed framework can also be readily implemented to other types of vehicles by replacing the energy-reward function and vehicle dynamic model.

1. Introduction

The growing transportation activities has been not only substantially enhancing the mobility of people and goods, but also producing more greenhouse gas (GHG) emissions and consuming a large amount of energy. In 2016, it is estimated that transportation sector has accounted for the largest portion (28%) of total U.S. GHG emissions, with 83% of the gas emitted by light-duty vehicles and medium- and heavy-duty trucks [1]. According to the statistics from U.S. Department of Energy, the energy consumption of transportation has kept increasing since 2012, reaching 28.2 quadrillion BTU (British thermal unit) and a share of 28.8% of U.S. total energy consumption by end-use sector in 2017 [2]. The increasing GHG emissions and energy consumption has drawn tremendous attention of government and researchers, and a series eco-driving projects and applications have come up throughout the years to improve the efficiency of the transportation system. Taking advantage of both vehicle-to-everything (V2X) and autonomous driving technology, connected and automated vehicle (CAV) emerges as one of the transformative solutions to the current challenges in sustainable transportation, such as traffic congestion, air pollution, and energy consumption [1], [2]. Connected vehicles (CVs) has shown the capability to improve traffic mobility and energy efficiency via vehicle-to-vehicle (V2V) or vehicle-to- infrastructure (V2I) communication. Meanwhile, automated vehicles (AVs) equipped with sensing technology (e.g., camera, Lidar, radar, etc.) and artificial intelligent (AI) technology would recognize the environment and subsequently perform proper actions by fully or partial automation. Urban traffic near the signalized intersection is one good scenario to utilize the advantages of CV-AV fusion, as the ego-vehicle may have extensive interaction with traffic signal timing and non-surrounding vehicles, which is imperceptible from on-board sensors but detectable via V2X communications.

Lee et al. proposed a cooperative vehicle intersection control (CVIC) system to enable the interaction between vehicles and infrastructure to optimize the efficient intersection operation [4]. According to the study conducted by Guler et al., the increase in the penetration rate of CV in low demand traffic can significantly reduce the average delay of passing intersections by applying connected platooning and adaptive signal control [5]. Elhenawy et al. proposed a game-theory-based control algorithm for cooperative adaptive cruise control (CACC) based AVs, which reduces travel time and delay under uncontrolled intersection [6]. In Europe, started from 2010, the project eCoMove has developed a transport energy efficiency system based on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I/I2V) communication, where real-time data can be shared among the vehicles and traffic controllers supporting a more fuel-saving traffic system [3]. In the U.S., Application for the Environment: Real-Time Information Synthesis (AERIS) research program established by the Intelligent Transportation Systems (ITS) Joint Program Office (JPO) in 2014 has developed 18 Connected Vehicle (CV) applications in 5 Operational Scenarios, among which Eco-Approach and Departure (EAD) at Signalized Intersections has been proven to be an effective application in decreasing fuel consumption and emissions [4].

The EAD application in the host CV can calculate the most energy efficient speed profile and guide the vehicle to pass the target traffic signal in an eco-friendly manner after collecting the

Basic Safety Message (BSM) from other CVs and Signal Phase and Timing (SPaT) information transmitted from the roadside equipment unit [5]. Besides the SPaT messages and traffic condition (number of queued vehicles or queue length) that serve as a main requirement for the application, other types of information such as geographic data (road map and grade) and vehicle dynamics also contribute to the calculation of an ideal speed profile. In the real-world traffic, as shown in Figure 1, signal timing and traffic conditions usually appear to be dynamic and uncertain. For example, when a CV is approaching an actuated signalized intersection, the remaining time of the current signal phase indicated by the SPaT message will be updated dynamically. Meanwhile, the traffic-related information received from other CVs and radar is also highly uncertain due to the limited sensing range and varying driving behaviors of other vehicles. Therefore, the future signal timing and traffic condition of the downstream intersection is hard to predict, which brings challenges to develop applicable EAD models.

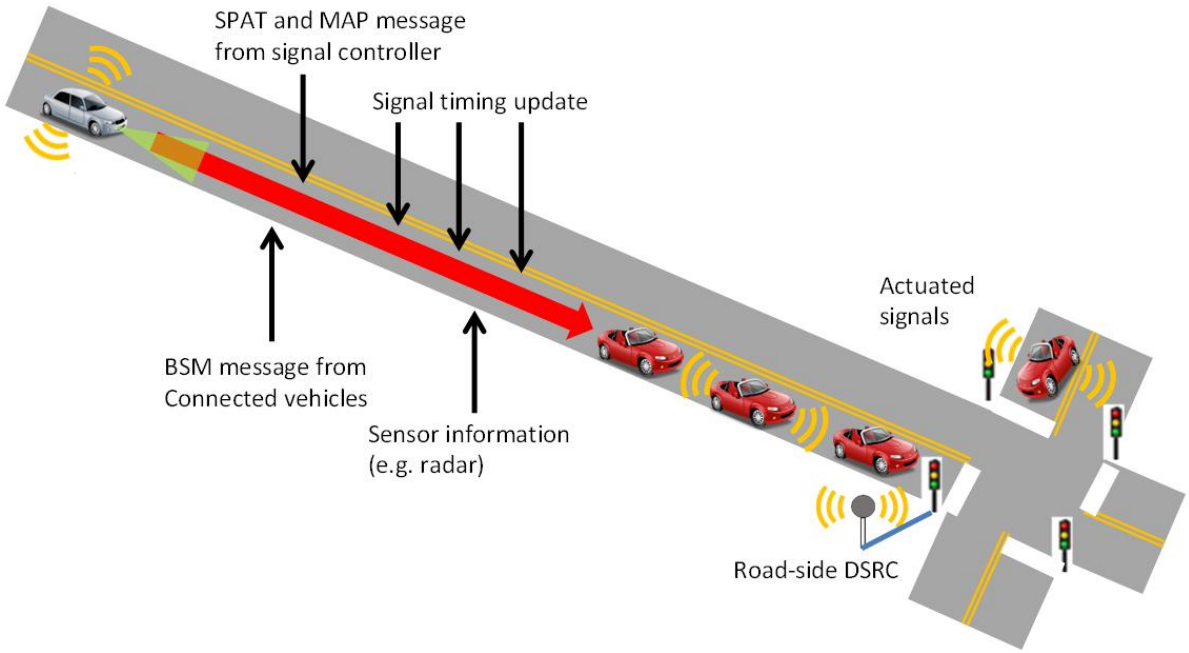


Figure 1. Dynamic information in connected eco-driving

The EAD application was initially developed under fixed-timing signal control, which 12% reduction on fuel consumption and CO2 emissions has been validated in microscopic simulation models [6]. Later studies also made no-preceding traffic or fixed-timing signal assumptions to avoid the uncertainty in the traffic condition [7-8]. He et al. obtained the speed profile by solving a multi-stage optimal function and put the queue information into constraints [9], Ye et al. estimated the end of queue based on the predicted preceding vehicle trajectories, with an assumption under congested urban traffic scenario such that a preceding vehicle could always be detected after SPaT messages are received [10]. All the above studies were conducted under the assumption that either queue does not exist or is fully predictable before trajectory planning. If the radar does not have enough sensing range to detect the preceding vehicle after signal information is received, those studies will not be able or will be less effective to design an optimal speed profile for drivers or longitudinal controller to follow. Meanwhile, most existing

EAD applications were developed for fixed-time signals. Actual signals, which are widely deployed in the U.S., receive less attention due to the high uncertainty in phase extension and skipping caused by vehicle actuation. This uncertainty increases the difficulty to predict the actual remaining time in a phase using signal phase and timing (SPaT) information. It further brings great challenge to derive an energy efficient speed profile for vehicles to follow.

To find the optimal solution to adapt the uncertain information, we propose a prediction based adaptive connected eco-driving system. The proposed system analyzes the possible upcoming traffic and signal scenarios based on historical data and live information stream collected from communication and sensing devices, and then choose the most energy-efficient solution that minimize the expectation of the energy consumption of all possible following actions. The objectives of the proposed research include:

1. Design an adaptive connected eco-driving strategy to optimize the energy consumption at the intersection considering preceding traffic and queues, using sensor detection and historical traffic data.
2. Develop an adaptive connected eco-approach and departure strategy that is applicable to actuated signals, using historical SPaT data.
3. Develop a multi-sensor based online connected eco-driving strategy using Reinforcement Learning for conditions with no external data (e.g., historical traffic or signal data)
4. Implement the algorithms in micro-simulation and test it under various traffic/sensing conditions.

The rest of this report is organized as follows. In Section 2, we present fundamentals on adaptive eco-driving, including the variables, the problems and the basic models. In Section 3, we show the models and numerical simulation results for uncertain traffic. In Section 4 we design the adaptive eco-driving strategy for actuated signals. In Section 5 and 6, we develop the Reinforcement Learning based connected eco-driving strategy and conduct micro-simulations to validate it. In Section 7 we conclude the report with discussions on future research.

2. Fundamentals in Adaptive Eco-Driving

In this section, we introduced the proposed connected eco-driving framework that could accommodate the host vehicle, traffic and signal timing information. Fundamental concepts, including variables and rules, are defined in this section to better support the mathematical modeling for this system.

2.1. Problem statement

In the proposed adaptive connected eco-driving system, nine types of information are fed into the algorithms to derive the most energy-efficient solution for the equipped vehicle. Their definitions are as follows:

1. Distance to intersection (D): the road distance from the current GPS location to the stop line.
2. Vehicle speed (V): the current speed of the vehicle, measured by on-board diagnostics (OBD) devices or GPS devices.
3. Time (t): current time stamp.
4. V2I communication range (C): this application works for both Dedicated Short-Range Communications (DSRC) or C-V2X. The V2I communication range is usually limited by the technology. As the connected eco-driving application start to work when the vehicle is within the communication range, we can assume $D \leq C$.
5. SPaT information: when a CV approaches within V2I communication range, it could receive SPaT information and know the phase status and the start and end time of the current phase.
6. Onboard sensor range (S): The maximum reliable forward detection range of the onboard sensor (e.g., radar, lidar or camera). Usually this range is less than the V2I communication range C, i.e., $S<C$.
7. Distance to the preceding vehicle (R): The measured distance to the preceding vehicle at the same lane. If there is no vehicle detected within the sensor range S, $R = -1$.
8. Speed of the preceding vehicle (U): The speed of the preceding vehicle at the same lane. If there is no vehicle detected within the sensor range S, $U = -1$.
9. The prior distribution of the queue length Q (in vehicle number), summarized from the historical traffic data. The prior probability of $P(Q=q)$ is a pre-defined function $f(q)$. The cumulative probability, $P(Q \leq q)$, is defined as $F(q)$.

We assume the sensors only report the states of the adjacent preceding vehicle (if any). Based on the information from sensors, there are three circumstances: 1. No preceding vehicle within the sensor range; 2. A stop preceding vehicle detected; and 3. A moving preceding vehicle detected. Base on the range that sensor can reach and the distance to the intersection, following cases are considered when we estimate the queue length or queue length distribution:

Case 1.1. No preceding vehicle within the sensor range (*U* = -1) and the vehicle is close to the intersection (*D*≤*S*): the queue length is 0.

Case 1.2. No preceding vehicle within the sensor range (*U* = -1) and the vehicle is far from the intersection (*D*>*S*): the possible queue length can vary from 0 to *D*-*S*. The conditional probability of the queue length can be formulated as:

$$P(Q = q|U = -1, D > S) = \frac{f(q)}{F(g(D-S))} \quad (1)$$

Here we use a function *g* to convert the queue length in distance into queue length in vehicle number:

$$g(y) = \left\lfloor \frac{y - l_{veh}}{l_{jam}} \right\rfloor + 1 \quad (2)$$

where l_{veh} is the average length of vehicle, l_{jam} is the average jam spacing (measured from vehicle font to vehicle front). We select the integer part of the value.

Case 2.1. A stop preceding vehicle is detected (*U* = 0) and the vehicle is close to the intersection (*D*<*R*): the preceding vehicle should be a queued vehicle at the downstream intersection if two intersections are closely spaced. In this case, the queue length for the study intersection is 0.

Case 2.2. A stop preceding vehicle is detected (*U* = 0) and the vehicle is far to the intersection (*D*≥*R*): the queue length in distance can be determined as *D*-*R*. The queue length in vehicle number is then calculate as $\frac{D-R-l_{veh}}{l_{jam}} + 1$.

Case 3.1. A moving preceding vehicle is detected (*U* > 0) and the vehicle is close to the intersection (*D*≤*R*): the preceding vehicle is traveling in the downstream of the stop line. In this case, the queue length is 0.

Case 3.2. A moving preceding vehicle is detected (*U* > 0) and the vehicle is far to the intersection (*D*>*R*): the possible queue length in distance can vary from 0 to *D*-*R*. The conditional probability of the *Q* can be formulated as:

$$P(Q = q|U > 0, D > R) = \frac{f(q)}{F(g(D-R))} \quad (3)$$

The above cases can be categorized into three types: A. No-queue cases (Case 1.1, 2.1, and 3.1); B. Deterministic queue cases (Case 2.2); and C. Non-deterministic cases (Case 1.2 and 3.2). In the following sections, we will develop eco-driving strategies for each type.

2.2. No-queue cases

The no-queue cases are the basic scenarios for eco-approach and departure application. In Hao et al [11], a graph-based trajectory planning algorithm was developed to solve the optimal solution for eco-approach and departure. In that paper, we assign a unique 3-D coordinate (*t, D, V*) to describe the dynamic state of the vehicle, which corresponds to the nodes in the graph.

The edges in the graph represent the movement of the vehicle, i.e., state transition from one-time step to the next. The cost on edge as the energy consumption during this state transition process. To formulate this graph model, we discretize the time and space into fixed time step Δt and distance grid Δd. The vehicle speed domain is therefore discretized with $\frac{\Delta x}{\Delta t}$ as the step. The energy consumption minimization problem is converted into a problem to find the shortest path from the source node *Vs* (*t, D, V*) to the destination node *Vd* (*T, 0, V'*) in the directed graph, where *t*, *D* and *V* are the current time, distance and speed of the vehicle. *T* is the target passage time at the stop line. For the red time arrival scenario, *T* can be identified as the start of the green time plus a buffer time, i.e., $T = T_g + \tau_b$. *V'* is the target speed when the vehicle passes the stop line. The Dijkstra's algorithm [12] is then applied to solve this single-source shortest path problem. This method shows good performance in energy efficiency but takes relatively long computational time in creating the graph and solving it.

In this project, to achieve higher computational efficiency and better compatibility with stochastic models, we reformulate this problem into a dynamic programming approach. The objective of this problem is defined as follows:

Give any initial state (*t, D, V*), find the optimal valid action that minimize the expected total cost over the rest of the path to the target state (*T, 0, V'*).

Here we say the transition from State 1 and State 2 is a "valid" action if they satisfy:

1. Time at State 2 is consecutive with time at State 1: $t_2 = t_1 + \Delta t$;
2. Consistency on distance and speed: $D_2 = D_1 + V_1 \Delta t$
3. Speed constraint: $V_2 = V_1 + x\Delta t$ and $V_{min} \leq V_2 \leq V_{max}$, where V_{min} and V_{max} are the minimum and maximum speed allowed.
4. Acceleration constraint: $V_2 = V_1 + x_1 \Delta t$, $(a_{min} \leq x_1 \leq a_{max})$, where a_{min} and a_{max} are the maximum deceleration rate and maximum acceleration rate.

Then we say State 1 is the valid parent state of State 2, and State 2 is the valid child state of State 1. Based on the criteria above, given state (*t, D, V*), the valid actions are included in the set of

$$\{t + \Delta t, D - V\Delta t, V + x\Delta t | a_{min} \leq x \leq a_{max} \text{ and } V_{min} \leq V + x \leq V_{max}\} \quad (4)$$

The acceleration rate *x* is therefore the key variable to define a valid action. According to the powertrain model in [11] and [13], the acceleration is also important in energy estimation for any type of vehicle or powertrain. We can formulate a powertrain-specific function $H(V, x, \Delta t)$ to represent the cost that the study vehicle varies its speed from *V* to $V + x\Delta t$ in Δt time.

We then use $M(t, D, V)$ to represent the minimum total cost at state (*t, D, V*), which corresponds to a series of optimal valid action from the initial state to the final state. This problem is then formulated in an iterative way as follows:

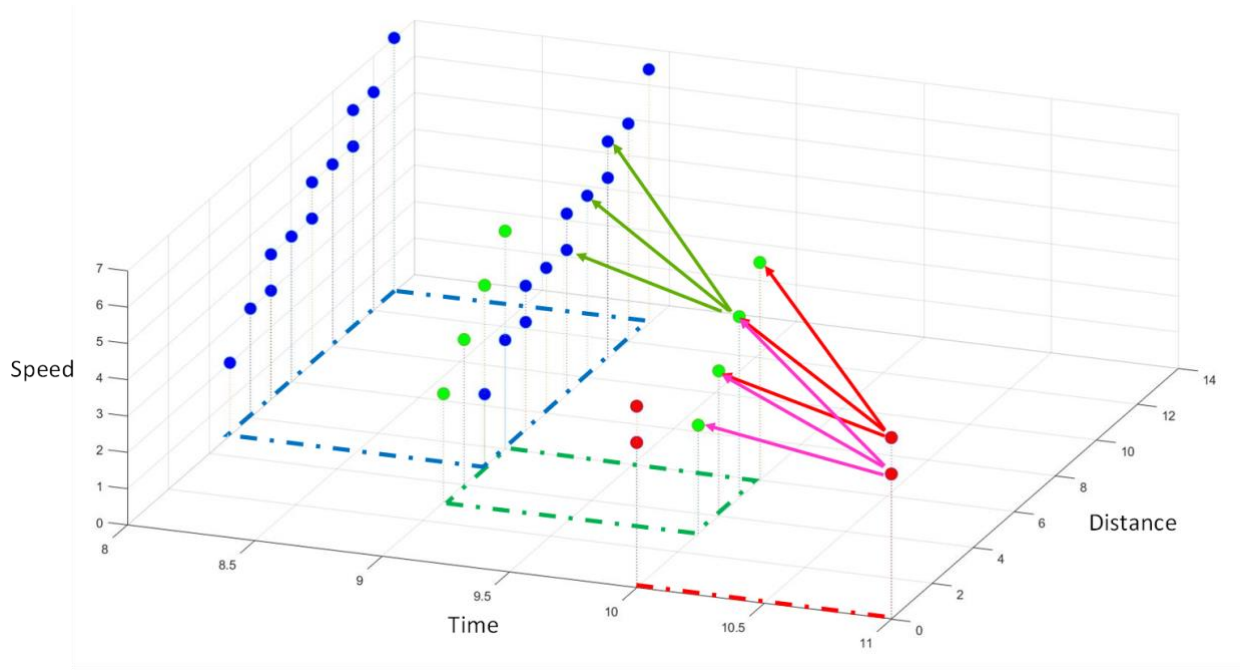
$$M(t,D,V)=\min_{x}\left(H(V,x,\Delta t)+M(D-V\Delta t,V+x\Delta t,t+\Delta t)\right) \quad s.t. a_{min}\leq x\leq a_{max} \quad V_{min}\leq V+x\leq V_{max} \tag{5}$$

We also define the values of the boundary states at or beyond the stop line. If the vehicle arrives at the stop line at the target time with target speed, $M(T,0,V')=0$. For other cases, e.g., 1) if the vehicle exceeds the stop line ($\mathrm{d}<0$); 2) if the vehicle arrives at the stop line at other time ($\mathrm{d}=0, t\neq T$); or 3) the vehicle arrives at the stop line with other speed ($\mathrm{d}=0, V\neq V'$), the total cost function is set to infinity, i.e., $M(t,D,V)=+\infty$.

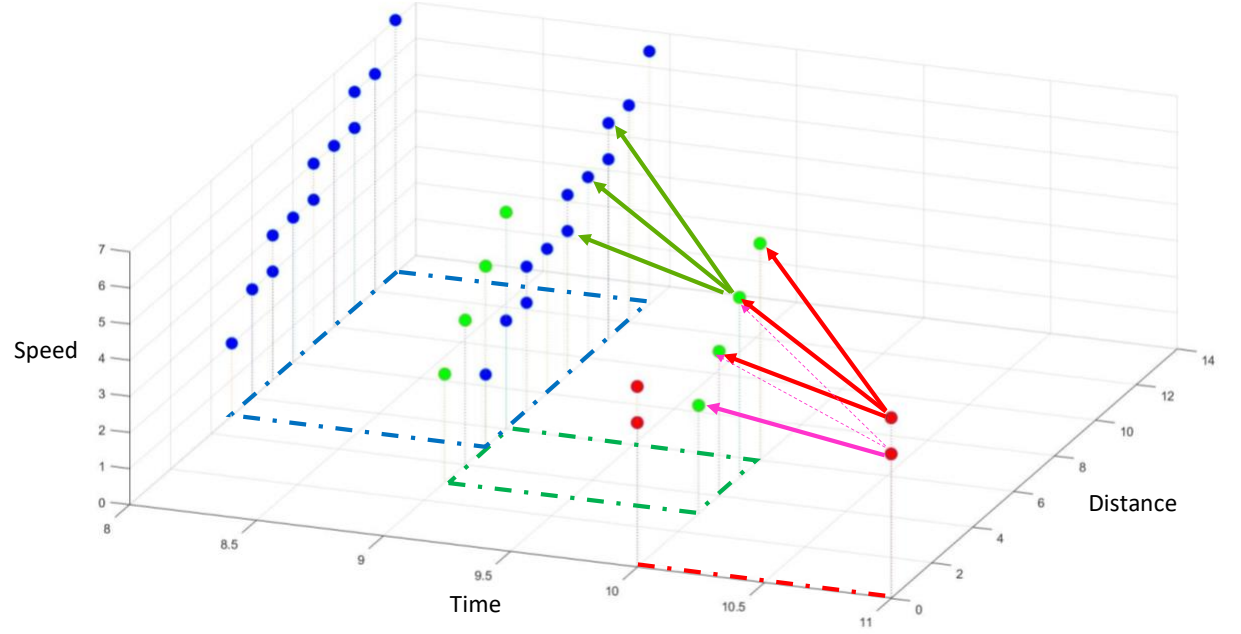
Based on all the assumptions above, this problem is formulated into a multiple-source single-destination shortest path problem. It can be solved using a variation Dijkstra algorithm in which two nodes are linked only if their time sates are consecutive. The pseudo codes below describe the algorithm. Here we use $X(t,D,V)$ to record the optimal acceleration rate at state (t,D,V).

Initialize the M values of all states with $+\infty$, i.e., $M(t,D,V)=+\infty, X(t,D,V)=0, \forall t,D,V$.
Set $M(T,0,V')=0$.
For $t=T:-\Delta t:T_{min}+\Delta t$
  For each (t,D,V)
    Find all the valid parent states of (t,D,V), i.e., $(t-\Delta t, D+V\Delta t-x\Delta t, V-x), \forall x$
      For each valid action *x*
        If $M(t,D,V)+H(V-x,x,\Delta t)<M(\tau-\Delta t, D+V\Delta t-x\Delta t, V-x)$
          Update $M(t-\Delta t, D+V\Delta t-x\Delta t, V-x)=M(t,D,V)+H(V-x,x,\Delta t)$
          Update $X(t-\Delta t, D+V\Delta t-x\Delta t, V-x)=x$
Return $M(t,D,V)$ and $x(t,D,V)$

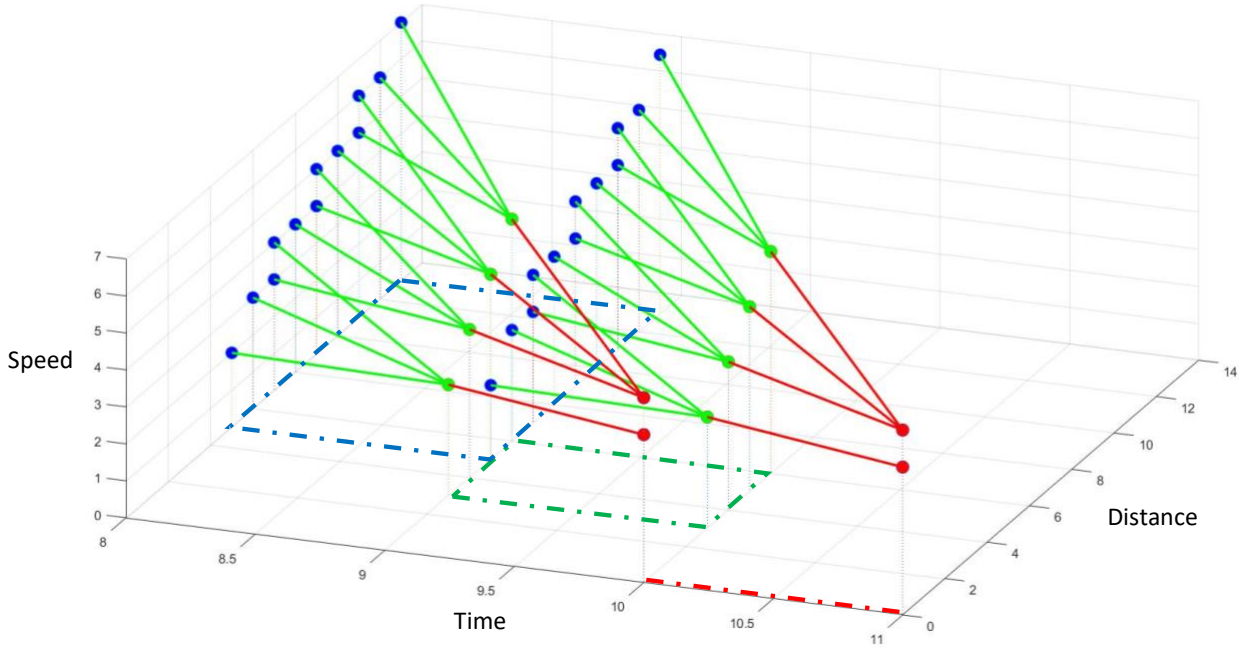
Figure 2 illustrates a simple example of this algorithm. We use blue, green and red dots to show the states in three consecutive time stamps. Figure 2(a) shows the process to find all the valid parent states of each state, using two red states and one green states as examples. Figure 2(b) shows that if one state has two or more valid child states, the optimal valid action corresponds to the one with lower *M* value. Figure 2(c) illustrates all the optimal valid actions for the blue and green states based on the proposed algorithm.



(a) The process to find all the valid parent states



(b) The process to identify the optimal valid actions



(c) All the optimal valid actions

Figure 2. A graph-based illustration of the proposed algorithm

2.3. Deterministic queue cases

The interaction of preceding traffic makes the connected eco-driving problem more realistic, and more challenging. There are several scenarios that the vehicle might met if it is trying to pass the signalized intersection after receiving the SPaT information. For example, if the current signal is green and the preceding vehicle is detected to be moving, the host vehicle could follow the preceding vehicle with an eco-adaptive cruise control strategy (which is not in the scope of this report). If the current signal is green and the preceding vehicle is detected to stop, then the estimated time that vehicle should arrive at the intersection could be calculated from the starting time of the current green phase with additional queue-discharging time calculated by the stop location and shockwave theory. If the current signal is red then the preceding vehicle is most likely to be detected to a stop (or on the way to a stop) at some time during the trajectory, and the sensing range together with the distance between the preceding vehicle and host vehicle restrict the distance of eco-driving. For all the cases except for the first one, the preceding vehicle's stop location (i.e., the end of queue) is crucial to determine the optimal speed profile for the host vehicle as it affects the location and time when eco-driving could start and finish.

If the end of queue is detected by sensors, the queue length in distance can be determined as *D-R*, the difference between the distance to the intersection and the range of sensor. The queue length in vehicle number, *Q*, is $\frac{D-R-l_{veh}}{l_{jam}} + 1$. As the queue discharging process would provide additional delay to the study vehicle when performing eco-driving, the target time *T*

under deterministic queue cases can be considered as a function of queue length *Q*. We use h_{sat} to represent the average saturation flow headway, τ_{SLT} to represent the start-up lost time, and T_b to represent the buffer time for the EAD vehicle to guarantee safety. The queue-aware target time is then expressed as:

$$T(Q) = \tau_{SLT} + h_{sat}Q + T_b \tag{6}$$

The perceived queue length *Q* is a new state variable in addition to time, distance and speed. We can revise the objective function in the previous section to accommodate it in the optimization problem.

$$\begin{aligned} M(t,D,V,Q) = \min_{x} & \left(H(V,x,\Delta t) + M(D - V\Delta t, V + x\Delta t, t + \Delta t, Q)\right) \\ & s.t. a_{min} \le x \le a_{max} \\ & V_{min} \le V + x \le V_{max} \end{aligned} \tag{7}$$

For certain queue length *Q*, if the vehicle arrives at the stop line at the corresponding queue-aware target time with target speed, $M(T(Q),0,V',Q) = 0$. For other cases, the *M* values are set to infinity.

One may notice that in this section, although we add a new dimension in the planning space, the new states introduced by different values of Q are more like parallel universes which do not interact with each other. In the next section, we will show how those universes interact with each other under uncertain traffic conditions.

3. Adaptive Eco-Driving Strategy Under Uncertain Traffic Condition

Based on the framework developed in Section 0, we propose an iterative approach to adapt the uncertain queue information so that the vehicle could start eco-driving even when entering the sensing range without knowing the current queue information. The previous section shows how to create the speed profile after detecting the end of queue based on the information acquired from I2V/V2V communication and onboard sensors. In this section we will derive the speed profile starting from the receiving of the SPaT messages to the detection of the end of queue, through analyzing the signal information and potential traffic condition based on historical data (queue distribution). The most energy-efficient solution can be then derived from minimizing the expectation of the energy consumption of all possible actions after combining the two phases.

3.1. Non-deterministic cases

If the onboard sensing cannot reach the stop line or is blocked by a moving preceding vehicle, the actual queue length at the intersection remains uncertain. In this case, the prior queue length distribution would impact the optimal strategy the vehicle should take when approaching the intersection. In the example shown in Figure 3, we create a scenario to explain why the prior queue length distribution is a significant factor in eco-driving strategy design. Assume the vehicle is 105 m far from the intersection, and the sensor range is 100 m. The vehicle is approaching the intersection with 5 m/s as the speed. As $D>S$, the queue condition remains uncertain (i.e., the queue length can be 1 or 0) until the vehicle gets closer in the next second. Then the vehicle is in a dilemma: to keep the current speed or decelerate to 4 m/s. The cost of the speed keeping action is 1, and the cost of the deceleration action is 0.5. Note that the costs are not the actual energy consumption in eco-driving. We just use them as a simple example to address the significance of prior queue information. In the figure we show the different residual cost (i.e., M value at next time step) under combinations of two strategies and two queue condition.

1. If keep the current speed and there is no queue, the vehicle can pass the intersection with constant speed with the residual cost of 1.
2. If decelerate and there is no queue, the deceleration is unnecessary. The vehicle has to use more energy to recover the speed. The residual cost is 2.
3. If decelerate and there is a queue, the deceleration is necessary as the vehicle has to slow down to bypass the queue. The residual cost is 3.
4. If keep the current speed and there is queue, the vehicle has to slow down harder in the rest of the path to avoid the queue. The residual cost is 4.

Then, if the intersection has less traffic, say the prior probability of no-queue case is 80% and that of queue case is 20%, the expected cost of speed keep strategy is lower than the deceleration strategy according to the figure. In contrast, if the intersection has more traffic, say the prior probability of no-queue case is 20% and that of queue case is 80%, the expected

cost of speed keep strategy is higher than the deceleration strategy. That means, the optimal strategy the vehicle should take is dependent to the prior traffic information.

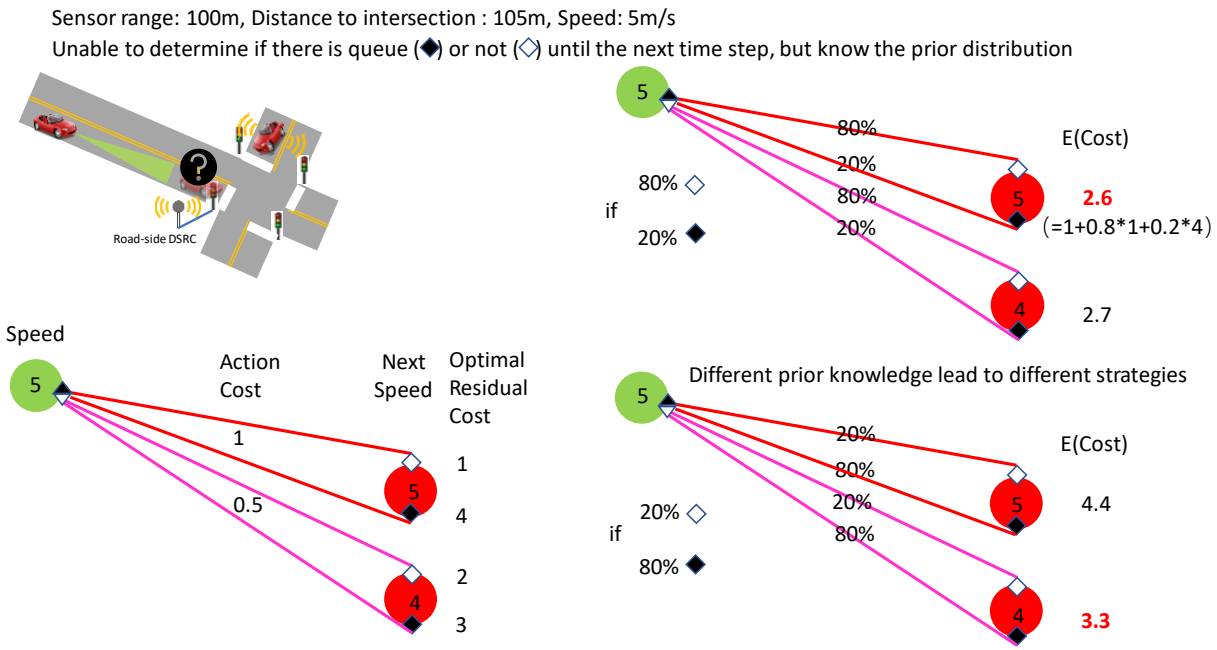


Figure 3. Adaptive strategy for uncertain traffic condition

For non-deterministic cases (Case 1.2 and 3.2), we define the queue state *Q* as -1. We first discuss Case 1.2 in which *D*>*S*. At certain non-deterministic state $(t, D, V, -1)$, the conditional probability that the actual queue length *q* is $\frac{f(q)}{F(g(D-S))}$ according to equation (1). We can define the potential queue pool for the current state as {0, 1, 2, … $g(D-S)$}. In the next time step, the vehicle precedes with distance of *V*. There might be two possible scenarios:

Scenario 1. No new detection by next time step. The vehicle is still under non-deterministic state $(D - V\Delta t, V + x\Delta t, t + \Delta t, -1)$, the conditional probability of *Q*=*q* becomes $\frac{f(q)}{F(g(D-S-V\Delta t))}$. The potential queue pool for becomes {0, 1, 2, … $g(D - S - V\Delta t)$}. As $g(D - S - V\Delta t) \leq g(D - S)$, some potential queue lengths are removed from the pool. The prior probability of this scenario, $\mu_{-1} = \frac{F(g(D-S-V\Delta t))}{F(g(D-S))}$.

Scenario 2. The queue is detected by next time step, or the sensing range exceeds the stop line without finding any queue. The vehicle switches to no-queue or deterministic state $(D - V\Delta t, V + x\Delta t, t + \Delta t, Q)$. The queue length *q* can be any value in the set of $\{g(D - S - V\Delta t) + 1, \dots, g(D - S)\}$. The prior probability of queue length *Q*=*q*, $\mu_q = \frac{f(q)}{F(g(D-S))}$.

The objective function is then formulated as follows:

$$M(t,D,V,-1)=\min_{x}\Big(H(V,x,\Delta t)+\mu_{-1}\bar{M}_{-1}+\sum_{q=g(D-S-V\Delta t)+1}^{g(D-S)}\mu_q\bar{M}_q\Big) \quad s.t. a_{min}\le x\le a_{max},\ V_{min}\le V+x\le V_{max} \tag{8}$$

where $\bar{M}_{-1}=M(D-V\Delta t,V+x\Delta t,t+\Delta t,-1)$, and $\bar{M}_q=M(D-V\Delta t,V+x\Delta t,t+\Delta t,q)$

In equation (8), $H(V,x,\Delta t)$ is the immediate cost of the action x, $\bar{M}_{-1}$ is the residual cost if the vehicle is still under non-deterministic state by next time step and $\bar{M}_i$ is the residual cost if the queue is detected as Q_i by next time step. The sum of probability μ_{-1} and μ_i's equals to 1.

The model for Case 3.2 is similar. We can just replace S with R to formulate the same optimization function as equation (8). The pseudo code to solve the problem in (8) is shown below.

Initialize M values of all states with $+\infty$, i.e., $M(t,D,V,Q)=+\infty, X(t,D,V,Q)=0, \forall t,D,V,Q$.
Set $M(T(Q),0,V',Q)=0$.
For $t=T(Q_{max}):-\Delta t:T_{min}+\Delta t$
  For each t,D,V
    For $q=0:1:Q_{max}$
      Find all the valid parent states of (t,D,V,q), i.e., $(t-\Delta t,D+V\Delta t-x\Delta t,V-x,q)$, and $(t-\Delta t,D+V\Delta t-x\Delta t,V-x,-1),\forall x$
      For each valid action *x*
        Let $\bar{M}_q=M(t,D,V,q),\ \mu_q=\frac{f(q)}{F(g(D+V\Delta t-x\Delta t-S))}$
        If $M(t,D,V,q)+H(V-x,x,\Delta t)<M(\tau-\Delta t,D+V\Delta t-x\Delta t,V-x,q)$
          Update $M(t-\Delta t,D+V\Delta t-x\Delta t,V-x,q)=M(t,D,V,q)+H(V-x,x,\Delta t)$
          Update $X(t-\Delta t,D+V\Delta t-x\Delta t,V-x,q)=x$
    Find all the valid parent states of $(t,D,V,-1)$, i.e., $(t-\Delta t,D+V\Delta t-x\Delta t,V-x,-1)$
    Let $\bar{M}_{-1}=M(t,D,V,-1),\ \mu_{-1}=\frac{F(g(D-S))}{F(g(D+V\Delta t-x\Delta t-S))}$
    For each valid action *x*
      If $H(V-x,x,\Delta t)+\sum_{q=-1}^{Q_{max}}\mu_q\bar{M}_q<M(\tau-\Delta t,D+V\Delta t-x\Delta t,V-x,-1)$
      Update $M(\tau-\Delta t,D+V\Delta t-x\Delta t,V-x,-1)=H(V-x,x,\Delta t)+\sum_{q=-1}^{Q_{max}}\mu_q\bar{M}_q$
      Update $X(t-\Delta t,D+V\Delta t-x\Delta t,V-x,-1)=x$
Return $M(t,D,V,q)$ and $x(t,D,V,q)$

3.2. Simulation and numerical results

3.2.1. Simulation setup and baseline methods

Numerical simulations are conducted in MATLAB to test the proposed method and compare with the baseline. Table 1 lists the assumptions for all the simulations:

Table 1. Simulation assumptions and parameters

D_0	Initial Distance	300 m
T_0	Initial Time	0 s
h_{sat}	Saturation headway	2 s
s_{jam}	Jam spacing	5 m
V_0, V'	Initial and final speed of host vehicle	13 m/s
V_{max}	Maximum speed	18 m/s
V_{min}	Minimum speed	0 m/s
a_{max}, a_{min}	Maximum and minimum acceleration	2 m/s2
Q	Range of queue length	[0, 20]
Δd	Distance step	1 m
Δt	Time step	1 s
Δv	Speed step	1 m/s

We create multiple baseline strategies to validate the proposed algorithms. The first strategy is ideal EAD method. The ideal trajectory for absolute minimum energy consumption can be derived when the actual queue length is known (i.e., perfect information) at the beginning of the simulation. This strategy can only be achieved if all vehicles are connected to share their positions to the study vehicle. Besides the ideal method, baseline EAD methods (Baseline_k) are setup for comparison: Assuming the queue length to be Q_k, the vehicle first follows the ideal trajectory of the assumed Q_k length, then change to the corresponding strategy after detecting the real queue length. These baselines are the methods given the same information as the proposed method except the historical queue distribution is missing. Note that if k is 0, Baseline_0 corresponds to the scenario when the vehicle follows the existing EAD strategy with no-queue assumption until the sensor detects preceding traffic.

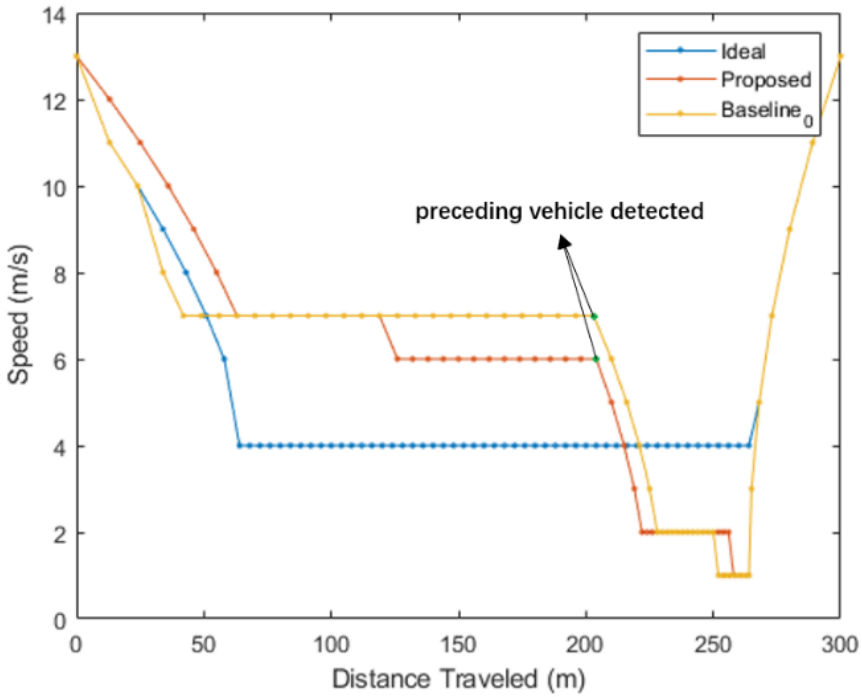
Note that for some baseline methods, there might not exist a solution, for example: the vehicle first travels under at an assumption of a large queue length, but actual queue length is small, therefore the vehicle will first travel at a relatively lower speed due to the assumed long delay and couldn't reach the traffic signal at required time after the queue is detected. In these cases, a delayed time (T') can be calculated as the minimum extra time that vehicle is given to finish the trajectory with predefined final speed V'. This delayed time increases the risk to miss

following green windows in the downstream signals. It will also force the following vehicles to slow down and result in an extra energy and fuel consumption to the system. To make a fair comparison, we use a penalty term to quantify the impact of the delay as the additional amount of energy (ΔM) that the delayed vehicle will consume to catch up with the fast vehicle with the same final speed.

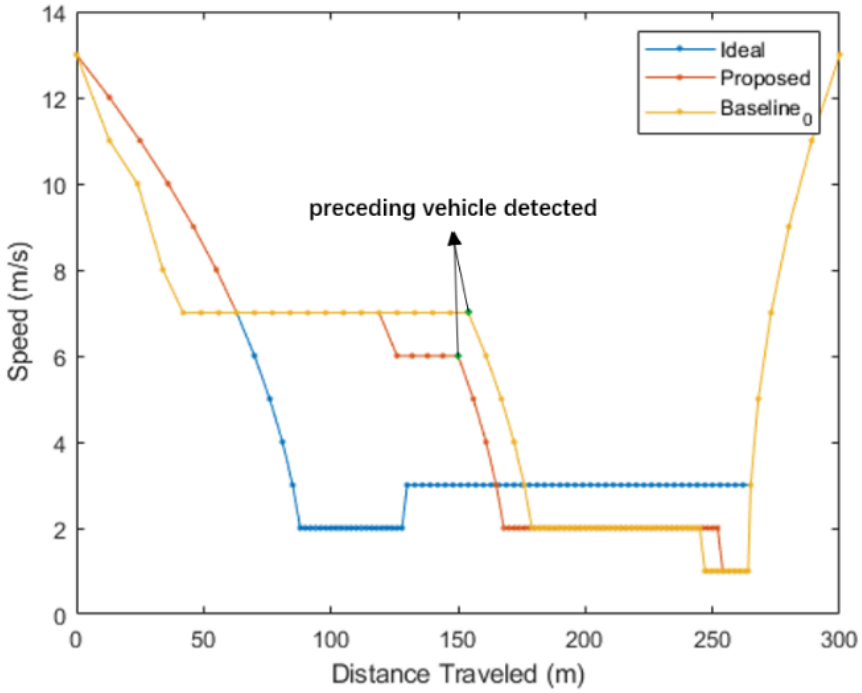
Therefore, all the methods including ideal, proposed and baseline can be evaluated with energy consumption and the result is shown in the following subsections. In the following subsection, we show some sample trajectory among different methods to address the significance of the proposed method.

3.2.2. Sample trajectories

First, two sample trajectories of the vehicle approaching traffic signal with different queue lengths derived from each method are shown in Figure 4. For the baseline method, zero vehicle is assumed to be waiting by the traffic signal, i.e., Baseline_0 is used. The other assumptions include: sensing distance S is 50m, Start of green time T_g is 40s and the queue length Q follows an uniform prior distribution between {0, 20}.



(a) Actual queue length Q = 10



(b) Actual queue length Q = 20

Figure 4. Speed profile of proposed against baseline and ideal method

Figure 4 compares the speed profile of proposed method against baseline and ideal methods with 10 and 20 as the actual queue length. Note that Baseline_0 and proposed method result in the same trajectory in the two plots before preceding vehicle getting detected (point labeled with green). Compared to the baseline method, the proposed method spends shorter time driving at higher constant speed, which saves 2.28% (top) and 2.17% (bottom) total energy respectively.

3.2.3. Results with varying actual queue length

We then compare the energy consumption among different methods for varying actual queue length. All the parameters are set up as follows: sensing distance S is 100m, start of green time T_g is 40s, and the queue length Q follows an uniform prior distribution between {0, 20}. As shown in Figure 5, the proposed method has a lower energy consumption than the baseline methods for most Q and only has a slightly larger energy consumption compared to ideal method. To compare with all the possible baseline methods, since distribution of Q is uniform, the average energy consumption, calculated as the average cost value of all Q's is shown in Table 2. The proposed method reduces the energy consumption by 3.35% (Baseline_0) and 8.88% (average of all 21 baselines), and is 2.24% higher than the ideal energy consumption.

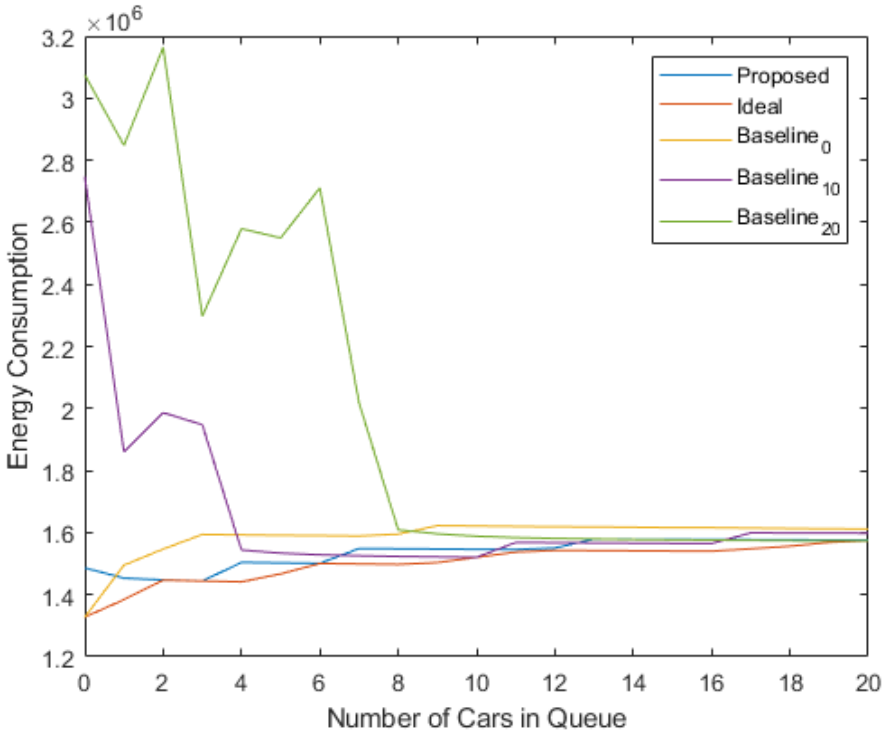


Figure 5. Energy comparison of proposed against baseline and ideal method in terms of different queue length (Unit: J)

Table 2. Average energy consumption comparison among three methods (Unit: 10^6 J)

Method	Energy	Method	Energy
Ideal	1.5011	Baseline10	1.6682
Proposed	1.5354	Baseline11	1.6998
Baseline0	1.5869	Baseline12	1.6235
Baseline1	1.5500	Baseline13	1.6506
Baseline2	1.5444	Baseline14	1.6727
Baseline3	1.5417	Baseline15	1.7176
Baseline4	1.5424	Baseline16	1.7365
Baseline5	1.5646	Baseline17	1.7949
Baseline6	1.5932	Baseline18	1.8532
Baseline7	1.6156	Baseline19	1.9315
Baseline8	1.6066	Baseline20	1.9908
Baseline9	1.6216		

3.2.4. Results with varying phase duration

In Figure 6, we compare the energy consumption among different methods when varying T_g, between 20s and 60s. For the ideal method, as shown in Figure 5, The energy consumption is monotonically increasing due to the more frequent acceleration and deceleration during a longer travel time. The proposed method shows a better performance than baseline methods when $T_g \geq 22$s. The worse performance for small T_g is caused by the high acceleration and speed of the vehicle that tries to arrive at the traffic signal at required time. The energy consumption tends to reach the same value as T_g increases among all methods.

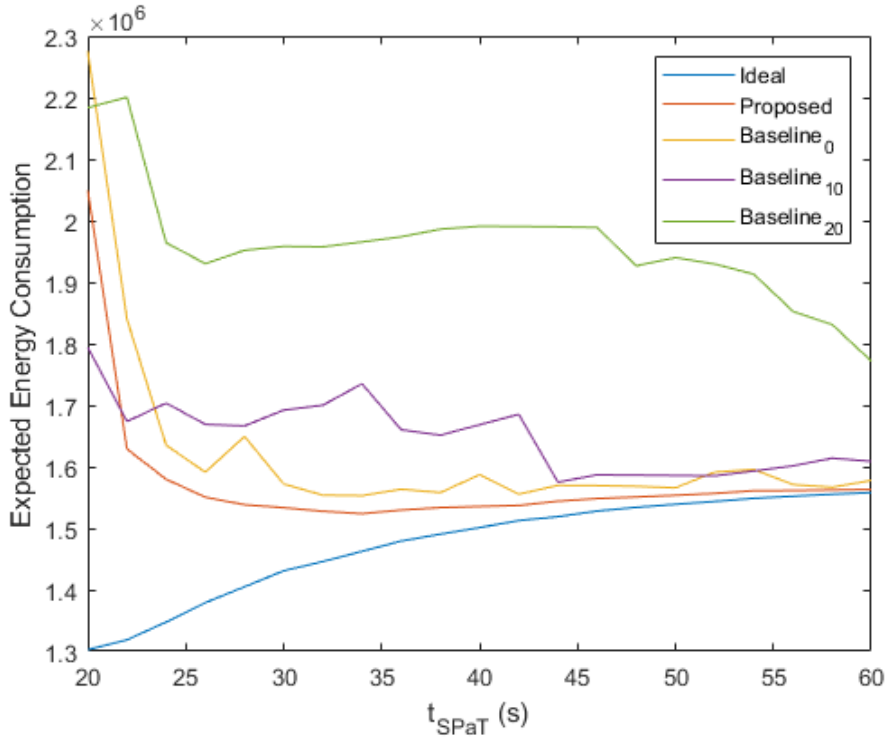


Figure 6. Energy consumption comparison of proposed against baseline and ideal method in terms of different phase duration (Unit: J)

3.2.5. Results with varying sensing range

In this subsection, we will compare the energy consumption among different methods for varying sensing range *S*. This simulates the various sensing ranges of all kinds of sensors when there is a queue in front of the host vehicle. The same parameters in above subsections are adopted, but multiple sensing ranges are tested.

As we can see from Figure 7, the proposed method always outperforms the baseline method. The average energy consumption of ideal method stays the same for all sensor range since the queue length is set to be known from the beginning. For both baseline methods and proposed method, the energy consumption gradually decreases as *S* increases, since the distance that queue is known gets longer and more trajectories can result in absolute minimum energy consumption. A detailed results table is shown in Table 3. It indicates that energy consumption of a vehicle equipped with adaptive EAD strategy and a 100m-range sensor is equivalent to a vehicle with conventional EAD strategy and a 190m-range sensor. To some extent, the proposed strategy could double the effective detection range in eco-driving.

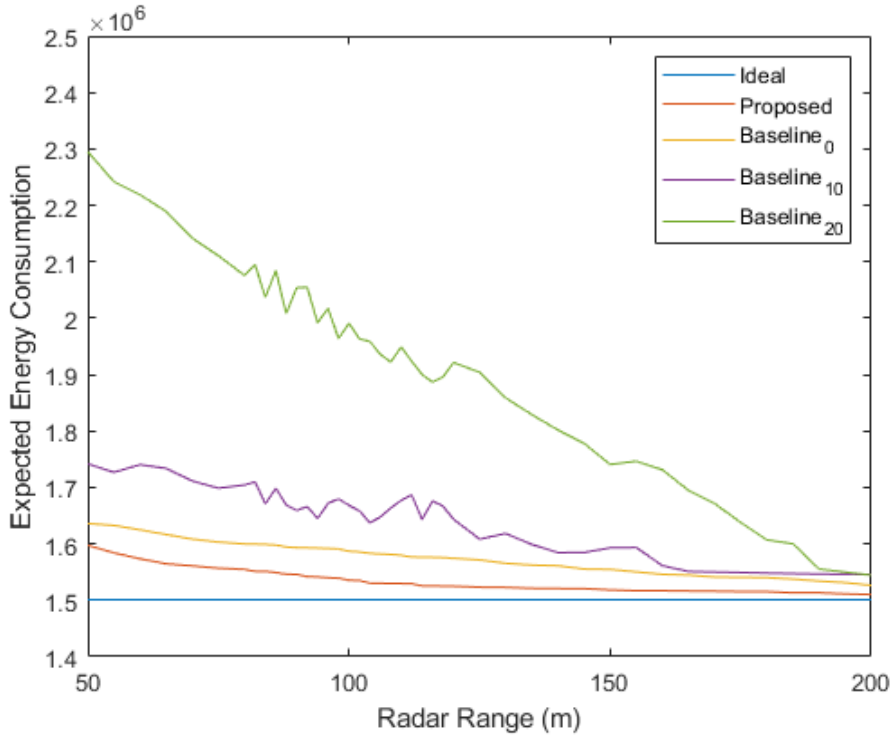


Figure 7. Energy consumption comparison of three methods in terms of different sensor range

Table 3. Energy comparison between three methods for different sensing range (Unit: 10^6 J)

Range	Ideal	Proposed	Baseline0	Baseline10	Baseline20
50	1.5011	1.5973	1.6360	1.7419	2.2949
60	1.5011	1.5735	1.6249	1.7403	2.2186
70	1.5011	1.5610	1.6085	1.7115	2.1418
80	1.5011	1.5549	1.5998	1.7043	2.0755
90	1.5011	1.5461	1.5933	1.6593	2.0540
100	1.5011	1.5354	1.5869	1.6682	1.9908
110	1.5011	1.5297	1.5797	1.6770	1.9489
120	1.5011	1.5250	1.5744	1.6438	1.9214
130	1.5011	1.5228	1.5655	1.6184	1.8585
140	1.5011	1.5209	1.5612	1.5846	1.8018
150	1.5011	1.5183	1.5547	1.5930	1.7407
160	1.5011	1.5170	1.5460	1.5614	1.7313
170	1.5011	1.5161	1.5411	1.5502	1.6715
180	1.5011	1.5154	1.5402	1.5482	1.6071
190	1.5011	1.5133	1.5340	1.5466	1.5554
200	1.5011	1.5103	1.5265	1.5458	1.5446

3.2.6. Results with different queue distribution

In this subsection, we will verify the capability of proposed method for a different queue distribution. Gaussian distribution is selected to generate queue length pattern. We set Q~N(10, 4) with other parameters the same as 3.2.3. Table 4 shows the comparison of expected energy consumption among different methods. The proposed method reduces the energy consumption by 4.14% ($Baseline_0$) and 3.56% (average 21 baselines) and is 1.88% higher than the ideal consumption.

Table 4. Energy comparison among three methods for Gaussian queue distribution (Unit: 10^6 J)

Method	Energy	Method	Energy
Ideal	1.5141	Baseline10	1.5693
Proposed	1.5431	Baseline11	1.5887
Baseline0	1.6070	Baseline12	1.5491
Baseline1	1.5695	Baseline13	1.5617
Baseline2	1.5624	Baseline14	1.5765
Baseline3	1.5552	Baseline15	1.5931
Baseline4	1.5433	Baseline16	1.6028
Baseline5	1.5604	Baseline17	1.6476
Baseline6	1.5619	Baseline18	1.6893
Baseline7	1.5487	Baseline19	1.7634
Baseline8	1.5444	Baseline20	1.8183
Baseline9	1.5479		

4. Adaptive Eco-Driving Strategy for Actuated Signals

In this section, we developed an adaptive EAD strategy for human drivers or automated vehicle controllers to minimize the expected energy consumption when passing an actuated signalized intersection. The historical SPaT data are applied to calculate the probability that one signal state (including phase status, time in the phase, minimum and maximum time-to-change) transfers to another state. On top of that, a graph-based model is created with nodes representing dynamic states of the host vehicle (distance to intersection and current speed) and signal state (passing time and estimated minimum and maximum time-to-change) and directed edges with weight representing expected energy consumption between two connected states. Then a dynamic programing approach is applied to identify the optimal speed for each vehicle-signal state iteratively from downstream to the upstream. The proposed data-driven method is applicable to any types of actuated signals without knowing the exact control rules for the signal. Real world SPaT data collected from the intersection of Wilmington Avenue and E Carson Street in Carson, CA is applied in the simulation, which has shown that the proposed method is robust and adaptive to varying signal conditions, and achieves 40% energy savings when the vehicle arrives in the red time, and 8.5% energy savings when the vehicle arrives in the green time compared to other baseline methods.

4.1. Problem statement

When a CV approaches an actuated signalized intersection and establish communication with the signal controller via DSRC or C-V2X, it could receive signal phase and timing (SPaT) information and know the status of current traffic signal with the phase status, phase starting time, current time in the phase and estimated minimum and maximum time-to-change for the current phase. Using the received SPaT information, distance to the traffic signal (D) and the current speed (V), the proposed method can derive the optimal speed profile from the preconstructed energy graph for minimum expected energy consumption. The host vehicle will then follow the suggested speed to achieve an eco-driving behavior.

Since the actuated signals actively respond to the traffic, the SPaT pattern can be different at every cycle. The uncertainty increases the difficulty of deriving an energy efficient speed profile. Most existing eco-driving methods give certain assumptions to the actuated SPaT information. Some of them assume that the minimum time-to-change, or maximum time-to-change, or both usually converge to the similar value that is close to the real time-to-change when the phase comes to an end [8]. The method will then derive an energy efficient speed profile based on the estimated time-to-change using the SPaT information. However, in the real-world traffic, the assumed time-to-change convergence might be slow or inaccurate. And when the time-to-change starts to get accurate enough for the algorithms, it might be too late to start eco-driving.

In order to solve the problem of the inaccurate minimum and maximum time-to-change SPaT information, instead of seeing them as time indicators, we use them purely as parameters along with the passing time, distance to the traffic signal and the current speed. The subsection

below describes how we process the real SPaT information and use them as the parameters for the nodes in the graph.

4.2. Statistical model using actuated SPaT data

The SPaT data applied in the algorithm were collected from the north bound of the intersection between Wilmington Avenue and E Carson Street in Carson, CA. The data were preprocessed so that only the 9am – 12pm time period between Dec 11th, 2018 to May 2nd, 2019 was included. The certain time period enables less plan variation and uncertainty in the graph construction, which makes the suggested speed more accurate for energy saving.

The three parameters in the SPaT data are elapsed time in the current phase(sec), estimated minimum total time for the current phase(sec), and estimated maximum total time for the current phase(sec) respectively. The latter two parameters are calculated using the passing time in the current time plus the minimum and maximum time-to-change. All the time parameters are rounded to integers to decrease the number of nodes in the SPaT graph described in the next paragraph.

A directional SPaT graph is constructed to calculate the probability of one SPaT state changing to the next. The node of the graph represents a specific SPaT combination. A directional edge is connected between two nodes if the current SPaT state has changed to the next in the collected SPaT data, and the weight of the edge represents the frequency of this state change. After the SPaT graph is constructed, the probability of one state changing to the next can be calculated using the weight of the edge divided by the total weight of the outgoing edges.

For certain vehicle dynamic state (time, distance, time) and SPaT state W = (elapsed time in the current phase, estimated minimum total time, estimated maximum total time), the objective function is then formulated as follows:

$$M(t,D,V,W)=\min_{x}\left(H(V,x,\Delta t)+\sum \mu_{W\to W'}\bar{M}_{W'}\right) \quad s.t. a_{min}\le x\le a_{max},\ V_{min}\le V+x\le V_{max} \tag{9}$$

Where W' is the possible SPaT state in the next time step, $\bar{M}_{W'}=M(D-V\Delta t,V+x\Delta t,t+\Delta t,W')$ is the residual cost if the next SPaT state is W', and $\mu_{W\to W'}$ is the probability that the the next SPaT state is W'. The sum of probability $\mu_{W\to W'}$ equals to 1.

For the graph construction of the phase for red light, we defined the start of green phase after the end of each red phase as the final state for the graph and formulate the remaining trajectory of the vehicle using rules. Similar state definition is created for the green light phase. The yellow phase (y_1) is created as the first second of the yellow light after the green light phase. For the states containing y_1, if $\frac{d_{TL}}{v}\le 3$, the vehicle will simply cross the road at its current constant velocity. If $\frac{d_{TL}}{v}>3$, the vehicle will enter the red phase time and apply the red light graph for the energy efficient driving. A time proportional energy penalty is given to each state to count for the time lost and encourage an efficient intersection passing.

4.3. Numerical experiment and results

Simulations are conducted in MATLAB to test the proposed method and compare with the baseline. Table 5 below shows the assumptions for all the simulations in the red and green light phase.

Table 5. Simulation assumptions and parameters

D_0	Initial Distance	300 m
T_0	Initial Time	0 s
V'	Final speed of host vehicle	13 m/s
V_{max}	Maximum speed	18 m/s
V_{min}	Minimum speed	0 m/s
a_{max}, a_{min}	Maximum and minimum acceleration	2 m/s2
Δd	Distance step	1 m
Δt	Time step	1 s
Δv	Speed step	1 m/s

4.3.1. Results for red-time arrival

For the case that the vehicle approaches the intersection in the red time, the baseline driver models is developed as follows: when the host vehicle enters the study zone, the vehicle will first accelerate to the maximum speed using constant acceleration of 2 m/s^2, then decelerate at -2 m/s^2 after reaching the safety distance. The safety distance is defined as the shortest distance the vehicle needs to stop at the intersection with the maximum deceleration. If the traffic signal changes to green phase in this process, the vehicle will immediately accelerate with the maximum acceleration and pass the intersection as soon as possible.

To compare the energy consumption between the proposed and baseline method, a total of 5000 historical SPaT messages are tested with different phase-entering time and initial velocity. If the final speed V' between the two methods are different, an extra energy is added to the one with lower speed to quantify the speed gap as form of energy. Table 6 shows the average energy consumption (10^6 J) between two methods (proposed, baseline) over all available SPaT messages with different phase-entering time and initial velocity.

Table 6. Simulation results for red-time arrival (Unit: 10^6 J)

Phase-entering time (s) / Initial velocity (m/s)	0		20		40		60	
5	**1.75**	3.42	**1.65**	3.44	**1.67**	3.43	**1.87**	3.51
9	**1.62**	3.22	**1.61**	3.24	**1.52**	3.21	**1.59**	3.27
13	**1.54**	2.38	**1.57**	2.40	**1.47**	2.37	**1.48**	2.43
17	**1.45**	1.53	**1.46**	1.56	**1.43**	1.53	**1.44**	1.56

As can be seen from the table, the energy consumption decreases as the initial velocity increases and the proposed method always outperforms the baseline method. A 5.23% ~ 52.65% energy saving can be achieved using the proposed method. The average energy saving is over 40%.

4.3.2. Results for green-time arrival

For the case that the vehicle approaches the intersection in the green time, the baseline driver is designed as follows: if the current speed of the vehicle is less than 13m/s, the vehicle will accelerate to 13 m/s using an acceleration of 1 m/s^2 and pass the intersection. If the current speed is higher than 13m/s, the vehicle will decelerate to 13 m/s using an acceleration of -1 m/s^2 and pass the intersection. And the vehicle will keep its current speed if it is driving at 13m/s.

A total of 5000 historical SPaT messages are tested with different phase-entering time and initial velocity for the comparison between two baselines and proposed method. A similar energy-time-velocity transformation is used to quantify the speed gap as form of energy. Table 7 shows the average energy consumption (10^6 J) between two methods over all available SPaT messages with different phase-entering time and initial velocity.

Table 7. Simulation results for green-time arrival (Unit: 10^6 J)

Phase-entering time (s) / Initial velocity (m/s)	0		5		15		25	
5	**4.47**	4.61	**4.56**	4.66	**8.16**	8.22	**6.27**	7.01
9	**3.93**	3.96	**3.97**	4.01	**7.18**	7.72	**6.24**	6.91
13	**3.31**	3.31	**3.32**	3.36	**6.97**	7.08	**6.21**	6.26
17	**2.68**	3.02	**2.63**	3.10	**2.80**	6.85	6.25	**6.14**

As can be seen from the table, the proposed method outperforms the baselines for most cases except for the 25s phase-entering time at 17m/s initial speed. For the 25s phase-entering time, the remaining green time in the current usually is not enough for the vehicle to pass. Therefore, a relative conservative strategy in baseline would save some unnecessary effort. In average, 8.5% energy savings can be achieved when the vehicle arrives in the green time compared to the baseline method.

5. Reinforcement Learning Based Connected Eco-Driving

In Section 3 and 4, we developed statistical models to perform adaptive connected eco-driving based on external historical data (e.g., historical traffic or signal data). However, if those data are not archived or accessible from the traffic operators, those strategies may fail to work. In Section 5 and 6, we will discussion the Connected Eco-Driving approach that only relies on the onboard sensors of the host vehicle, by utilizing the tremendous generalization power of deep learning (DL) [14] and reinforcement learning (RL) [15], which not rely on specific models or rules. Particularly, reinforcement learning has demonstrated its significant power in dealing with policy learning tasks by itself without predefined human rules or models in a complex environment [16], [17], including transportation. Owing to the great capability of RL, many researchers are trying to apply RL algorithms into autonomous driving tasks. A deep RL-based autonomous driving framework was proposed by Sallab et al. to enable automatic lane- keeping with interaction with simple traffic [18]. Desjardins et al. proposed an RL-based cooperative adaptive cruise control (CACC) method by utilizing V2V information, which can result in efficient behavior in CACC [19]. Shalev-Shwartz proposed an RL-based safe driving model, which enables multi-agents to merge smoothly in a double-merge scenario [20]. For signalized intersection scenario, Chen et al. proposed a hierarchical RL-based driving behavior control model, which enables the vehicle to basically interact with the traffic signal (i.e., stop or go with different phase) [21]. Most existing RL-based algorithms focused on lane-keeping, CACC, merging and traffic-signal interaction, but few RL algorithms are used in intersection-based eco-driving strategy to the authors' knowledge. One reason may be that RL algorithms are good at solving single logical task while the intersection-based eco-driving has at least three different logical tasks:

(1) Energy efficiency, which requires the vehicle to drive through the intersection with less energy consumption.

(2) Intersection interaction, which requires the vehicle to interact properly with the traffic signal.

(3) Traffic interaction, which requires the vehicle to interact with traffic without colliding into each other.

In this study, to further explore the eco-driving strategy of CAV under realistic mixed connected traffic around a signalized intersection, we proposed a hybrid reinforcement learning (HRL) framework to learn long-term driving strategies. The key contributions of this research includes (1) we proposed an HRL framework for a logically complex task, like eco- driving with signalized intersection; (2) an innovative long- short term reward algorithm is proposed, which provides the RL model with the ability to learn complex driving strategy from conflicting factors (i.e., speed and energy); (3) The traffic environment is considered as mixed traffic where the other vehicles are human-driven (i.e., without connectivity) and have a different dynamic model; (4) In order to make the mixed traffic more realistic, intelligent driver model (IDM) is applied in building the mixed traffic; and (5) a multi-sensor-based RL- network is proposed, which enables the ego-vehicle to interact properly with the mixed traffic.

5.1. Problem statement

The main purpose of this study is to design an RL-based framework to conduct an eco-driving strategy for vision-based CAV under mixed traffic in the signalized intersection. This problem can be formulated as an optimal policy learning task with three main goals of the policy: (1) to save energy consumption, (2) to reduce the travel time, and (3) to safely interact with traffic and signal. The proposed RL-based framework has five key compositions and the connections between these compositions and the traffic situation in this paper are shown as follows:

1. Agent: the ego-vehicle which can perceive the environment via front camera and V2I-based SPaT information;
2. Environment: traffic environment which includes various kinds of vehicles and signalized intersection;
3. Policy: the proposed eco-driving policy;
4. Action reward: the short-term benefit (i.e., speed reward, energy consumption) of taking action right at this moment and the long-term benefit (i.e., travel time, total energy consumption) of the journey;
5. Action-value function: the function to determine which action is the best choice at the next moment to achieve a long-term optimal result.

These connections illustrate that the issue in this research can be well interpreted by the RL framework. RL framework is established based on Markov Decision Process (MDP) which is a mathematical framework for decision making via the interaction between a learning agent and its environment in terms of state, actions and rewards. In this research, the ego-vehicle (i.e., agent) interacts with the environment (i.e., mixed traffic and signalized intersection). To be specific, the traffic environment, agent observation, agent actions are discussed as follow.

5.1.1. Traffic environment

The traffic environment includes three main parts: the ego-vehicle, other vehicles, and a five-lane signalized intersection.

In order to make the proposed environment more similar to the real traffic, we designed a different kind of other vehicles and different start phase time of the traffic light. To be specific, the other vehicles are divided into five kind vehicles which have different dynamic model and behavior strategy. To make the other vehicles more realistic, we applied the intelligent driver model (IDM) to the other vehicles' longitudinal control method. For the latitudinal control, we designed different rates for other vehicles to change the target lane. The detail of the description of other vehicles is shown in Table 8.

Table 8. The description of the dynamic model of the vehicles

Vehicle	MAX_ACC	MAX_DEC	GAP_ACC	HW_ACC	V_TAR	LAT_RATE
EV1	6.0 m/s2	6.0 m/s2	3 m	1.5 s	13.8 m/s	0.3
EV2	5.0 m/s2	4.5 m/s2	3 m	1.5 s	12.5 m/s	0.2
EV3	3.0 m/s2	5.0 m/s2	2 m	1.2 s	11.1 m/s	0.2
EV4	3.0 m/s2	3.0 m/s2	3 m	1.5 s	9.72 m/s	0.1
EV5	2.0 m/s2	1.5 m/s2	5 m	1.5 s	8.33 m/s	0.1

In the table, MAX_ACC, MAX_DEC, GAP_ACC, HW_ACC, V_TAR, and LAT_RATE represent the maximum acceleration, the maximum deceleration, the minimum distance to the front vehicle, Safe time headway, Desired velocity and the rate of change target lane separately.

For the traffic light, every time the simulation start, the initial phase and time are randomly selected in the entire cycle duration. Specifically, the duration of greed time, yellow time, red time and all-red time phases are set as the 20s, 3s, 40s, and 1s respectively.

5.1.2. Agent

The proposed method is applicable to any type of vehicles given its dynamic and powertrain characteristic. As an example, the ego-vehicle is modeled as an electric CAV in this research. The maximum acceleration and deceleration are set as 3m/s2 and -3m/s2 respectively. Furthermore, the observation input, energy consumption model (ECM) and action output are defined as follow.

For the input observation, in this study, the perception information comes from three main parts: (1) V2I communication-based signalized traffic light information which includes the current phase state and the duration time; (2) the on-board sensor which includes three radars (left distance d_l, right distance d_r and front distance d_f) and front camera (image size 320*160, 50fps); and (3) on-board diagnosis which include the ego-vehicle speed v_e and acceleration value a_e.

Due to the multiple-input data, in order to enhance the learning performance, we find an efficient way to decrease the complexity of the input observation without losing too much information. For the radar data, we define three variables which are the forward warning w_f, the left warning w_l and the right warning w_r. The definition of these variables is shown as follow:

$$w_d = 3 + (v_e - v_f)^2/2a_e$$
$$w_f = \begin{cases} 0, d_f > w_d \\ 1, d_f <= w_d \end{cases}$$
$$w_l = \begin{cases} 0, d_l > w_l \\ 1, d_f <= w_l \end{cases}$$
$$w_r = \begin{cases} 0, d_r > w_r \\ 1, d_r <= w_r \end{cases} \quad (10)$$

Where w_d, v_f, w_l, w_rrepresent the forward warning threshold distance, forward vehicle velocity, left-warning threshold distance and right-warning threshold distance separately. Specifically, $w_l, w_r$are both set as 2m. Thus, the observation is composed of a 12-dimensional vector $\{t_g, t_y, t_r, w_f, w_l, w_r, w_c, d_r, d_f, v_f, v_e, a_e\}$ where t_g, t_y, t_r, d_rrepresent the duration time of green light, yellow light, red light, and the remain distance.

For the energy consumption model, we applied one of our previous work in which an energy consumption model was proposed and calibrated by real-world driving data from a 2013 NISSAN LEAF [22]. The original energy consumption model is shown below.

$$E = -3.037 - 0.591v\,cos(\alpha) - 1.047 \times 10^{-3}v^3 - 1.403va + 2.831 \times 10^{-2}v^2\,cos(\alpha)$$
$$-7.980 \times 10^{-2}v^2a - 1.490v\alpha\,sin(\alpha) + 3.535 \times 10^{-3}v^3a - 0.243va^2 -$$
$$1.279v\alpha\,cos(\alpha) + 6.484 \times 10^{-4}v^3\alpha + 0.998va\alpha \quad (11)$$

Where a, v, α represent the instant acceleration (m/s2), speed (m/s) and road grade (rad) separately. In this study, the road grade is set as zero. Besides, in the original model, the breaking will charge the battery, which may cause a negative influence on RL learning. Thus, we redefined the original model which is shown as follow.

$$E_{rl} = \begin{cases} E, a >= 0 \\ 0, a < 0 \end{cases} \quad (12)$$

where E_{rl}represent the energy consumption model applied in our framework.

For the output action, in this study, the main purpose is to learn an optimal strategy in both longitudinal and latitudinal driving maneuvers. Thus, the output actions are defined both on these two dimensions. For longitudinal maneuver, the action space is $[1.0a_m, 0.8a_m, 0.6a_m, 0.4a_m, 0.2a_m, 0.0, 0.2d_m, 0.4d_m, 0.6d_m, 0.8d_m, 1.0d_m]$ where a_m, d_m represent the maximum acceleration and deceleration separately. For the latitudinal maneuver, the target lane action space is [-1, 0, 1] where -1, 0, 1 represent the target lane is the left lane, the current lane, and the right lane separately.

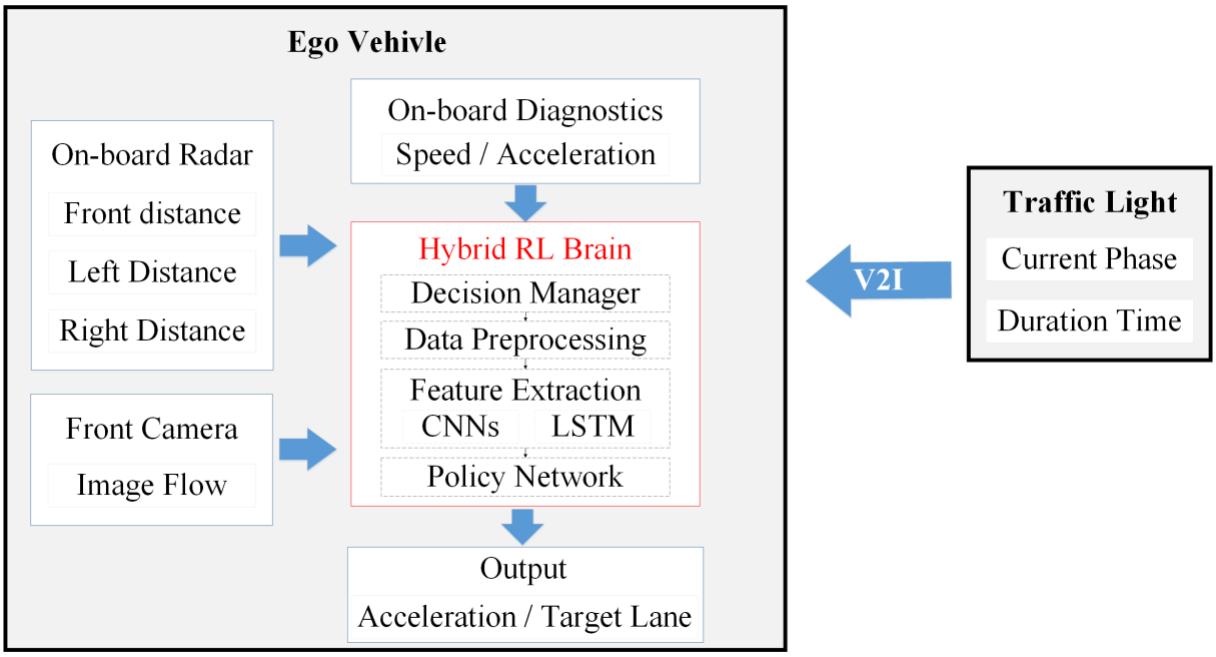
Furthermore, in order to enhance the learning performance, we find a way to decrease the dimension of action space. Generally, the action space will be a 33-dimensional vector. However, we define that the lane-changing maneuvers are only available when the longitudinal acceleration is zero, which means when the vehicle is accelerating or decelerating it should not

change lane (this also fit with the real-world traffic safety rules). Thus, in our paper, the action space is a 13-dimensional vector.

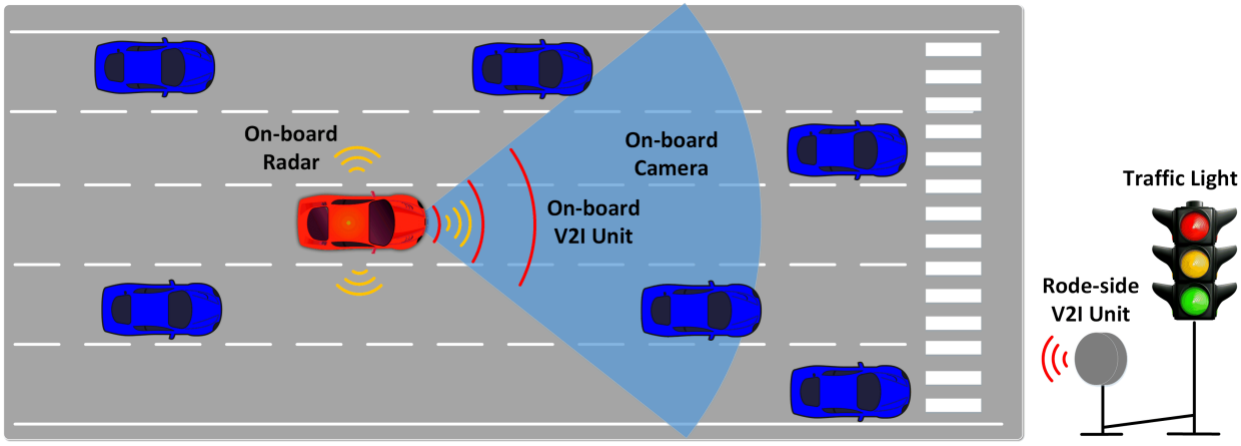
5.2. Hybrid RL-based eco-driving framework

In this research, we proposed the hybrid RL-based CAV eco-driving framework and algorithms for electric passenger vehicles. Figure 8 illustrates the key components of the hybrid-RL based CAV eco-driving system which consists of several components as described briefly below.

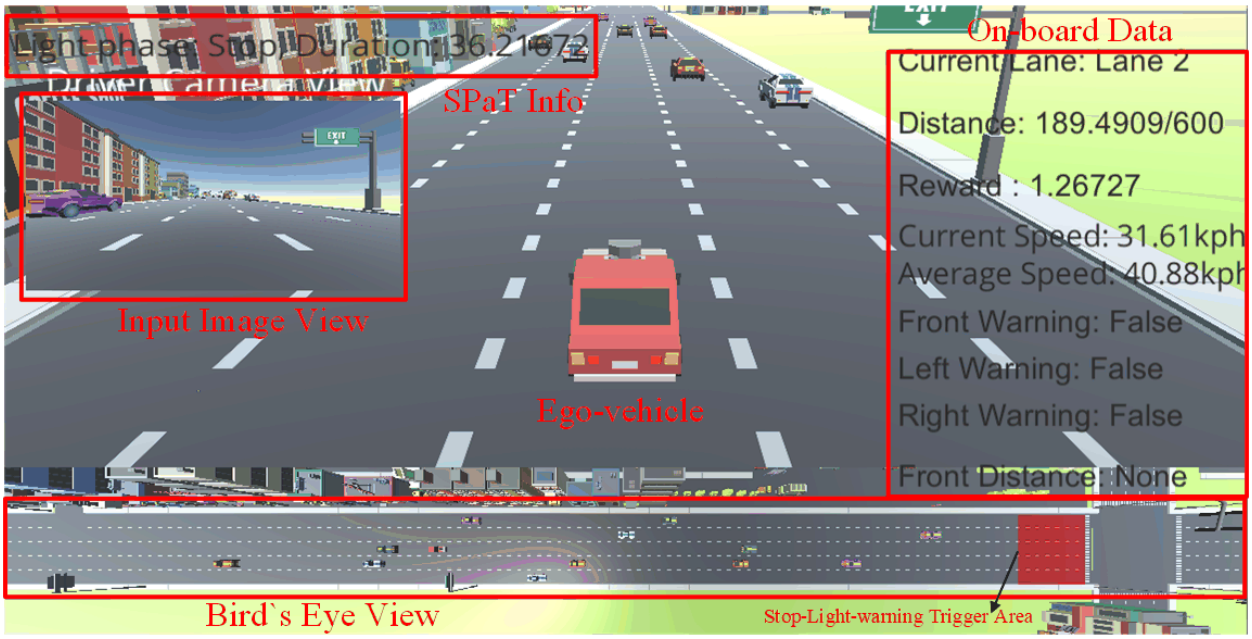
5.2.1. System architecture



(a) The systematic architecture of the HRL method.



(b) Traffic environment structure.



(c) The view of traffic simulator.

Figure 8. The HRL-based eco-driving system architecture

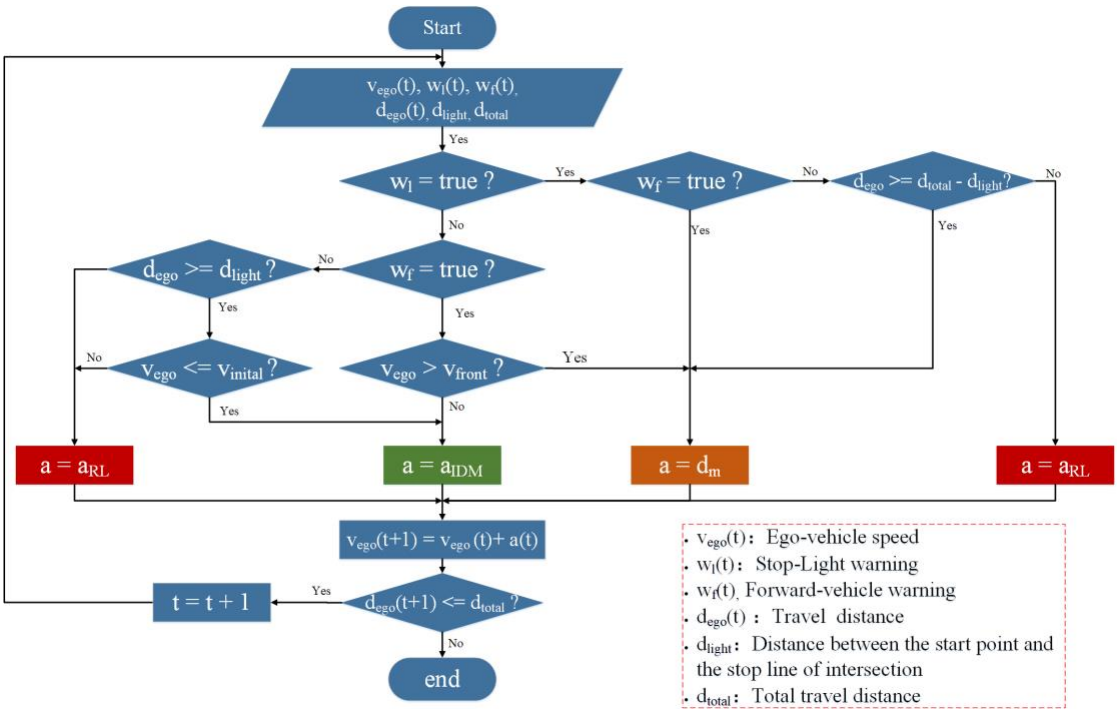
1. On-board computer: it houses the hybrid-RL brain which is the decision-making center of the whole system. The brain will receive perception data and process the data. Then it will return the longitudinal acceleration and target lane information to the vehicle control center.
2. On-board radar: there are three radars applied in this research. One is installed in front of the vehicle, which is used to detect the front distance and front-vehicle velocity. The other two radars are installed at the left side and right side of the vehicle, which are used to detect the left distance and right distance. The range of radars is 100 meters in the simulator. The radar information will be sent to the on-board computer as part of the agent observation.
3. On-board camera: it is installed in the front of the vehicle. The direction of the camera is the same as the vehicle's driving direction. The camera information will be sent to the on-board computer, which is the key part of the traffic perception.
4. On-board diagnostics (OBD): this component can get the instant speed and acceleration information and then the message will be sent to the on-board computer as part of the observation.
5. Traffic light and road-side unit: the traffic light will send the real-time traffic light information to the road-side unit. Then the road-side unit will send the information to any CAV within its communication coverage.

As the most crucial component in the system, the hybrid-RL brain aims to drive through an intersection with less time and energy consumption by generating appropriate instant

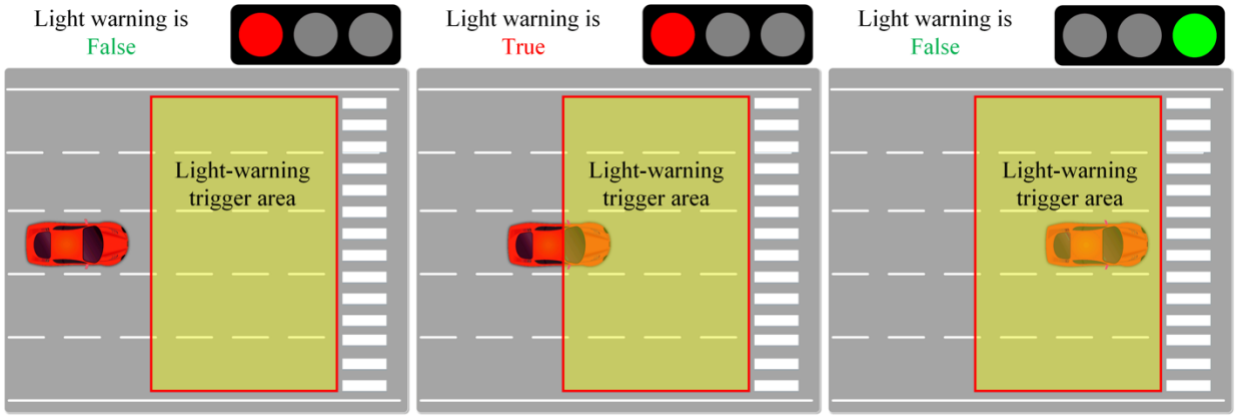
longitudinal acceleration and target lane. As we have discussed above, driving through intersection traffic is a logically complex task. Because there are at least four sub-tasks: (1) cruise control that avoid collision with other vehicle; (2) lane-changing decision that in the traffic; (3) stop-light reaction that stop the vehicle before the stop line when the light is red or yellow; and (4) go-light reaction that speed up when the light is now becoming green. So far, it is nearly impossible for one RL algorithm to handle such a multi-phase task. In this research, we proposed a hybrid RL brain which combines the manual-designed rules and RL algorithm. The hybrid-RL brain consists of several components which are illustrated as follow.

5.2.2. Decision manager

The key component of the hybrid-RL framework is the decision manager because this part combines the manual-designed rules and the RL algorithm to endow the framework with the ability to handle the complex task. In the decision manager, the driving process is divided into different running situations according to the immediate situations of ego-vehicle and traffic light. The architecture of the decision manager is shown in Figure 9(a).



(a) The rule-based workflow of Decision Manager



(b) the detail of the definition of one key variable in Decision Manager: The Stop-Light warning

Figure 9. The architecture and detail of Decision Manager

Figure 9(b) illustrates the definition of Stop-Light warning $w_{light}(t)$. When the vehicle enters the Stop-Light-warning trigger area and the current light is red or yellow, the $w_{light}(t)$ is True. On the other hand, if the vehicle does not enter the trigger area or the light is green, the $w_{light}(t)$ is False. The main purpose of Stop-Light warning is to build a safe and stable vehicle-signal interaction, as it is hard for the RL to learn all the logically different tasks.

In addition, the IDM and emergency breaking model are also integrated into the decision manager. The IDM model is activated when the ego-vehicle need to start at an intersection when the light becomes green. The emergency braking model is defined as $v(t+1) = v(t)d_m$,

where d_m is set as $5m/s^2$and the emergency braking will be activated when the w_f (t) is True.

5.2.3. Data preprocessing

In this research, observation input of ego-vehicle consists of (1) a 50 fps raw image flow and (2) a 13-dimensional vector. Generally, the raw image data cannot be fed into the training network directly because the size of the raw image is not compatible with the convolutional neural networks. Besides, the purpose of this study is to explore the ability of the hybrid RL to generate the eco-driving strategy for CAV. Hence, considering the driving strategy relies on both spatial and temporal information, instead of resizing the images, we also need to transform multiple single-frame images into a multi-frame spatiotemporal data format. To be more specific, in this study, in order to include more information without making the input data too heavy (having too many frames in one format). We proposed a select-stack preprocess method. The whole preprocess includes four steps as below:

1. Recording the raw image flow based on the time sequence;
2. Resizing every frame of the raw image flow into an 80*80*3 format;
3. Selecting one frame out of N_{select} frames;
4. Stacking N_{stack} frame of images into a higher dimensional data format: 80*80*3*N_{stack}.

Specifically, in this study, the N_{select} and N_{stack} are both set as 4. Through the above preprocessing method, the training network could get spatiotemporal observation data, which is of considerable significance on the driving strategy learning procedure.

5.2.4. Deep RL for eco-driving

According to the above discussion, the eco-driving approach of going through an intersection can be formulated as an MDP in which the agent interacts with the environment. Furthermore, due to the discretion of action space in this research, we applied Dueling Deep Q Network (Dueling DQN) as our basic RL framework. The Dueling DQN is developed from Deep Q Network (DQN), which is briefly introduced below.

Deep Q Network (DQN) is a typical deep RL algorithm that uses a deep neural network to predict the value function of each discrete action. DQN performs in discrete action spaces and aims to choose the action with maximum value output. Specifically, the input of DQN is observation state o_t, and the output is the evaluation value $Q(s_t, a_t)$ corresponding to each action at state A. Then, according to the e-greedy algorithm, an action is selected from the action space. After the execution of action at, a reward r_t and an observation state o_t+1 can get from the environment.

In addition, prioritized experience replay algorithm is used to solve the problem of correlation and non-static distribution. Experience state $e_t(s_t, a_t, r_t, s_t+1)$ will be stored in the experience pool $E_t = (e_1, e_2, ..., e_t)$. During the training process, a mini-batch of data will be selected randomly from the experience pool so that the training process can avoid the correlation problem.

$$L(\theta) = (R_t + \gamma max_{A_{t+1}} Q(O_t, A_t; \theta^-))^2 \quad (13)$$

where γ is a discount factor, θ is parameters of neural network and θ^- is the parameters of target network.

In many visual perception-based DRL tasks, the value functions of different state actions are disparate, but in some states, the size of the value function is independent of the action. Thus, Wang et al. proposed a dueling network-based DQN model named Dueling DQN [23]. Dueling DQN is constructed with two streams which separately estimate (scalar) state-value and the advantages of each action and shows significant performance improvement than DQN. In Equation (14) we show the function to calculate Q-value of Dueling DQN is designed to aggregate the states-value and action advantages.

$$Q(S_t, A_t; \theta, \alpha, \beta) = V(S_t, ; \theta, \beta) + A(S_t, A_t; \theta, \alpha) - \frac{1}{|A|}\sum_{A_t} A(S_t, A_t; \theta, \alpha) \quad (14)$$

As shown in the equation, α represents the parameters of *A* (the advantage function). Besides, β represents the parameters of V (the state-value function) and θ is parameters of the neural network. Thus, due to the performance of long-term reward learning and vision-based RL task, Dueling DQN is applied in this research as the basic RL algorithm of the hybrid RL framework.

However, in this research, the biggest challenge is that we want to find a method that can decrease energy consumption without spending more travel time (even save time) under mixed traffic condition. As is well-known, when driving on a freeway, the long-term travel time depends, to most extent, on the short-term reward such as the instant speed or lane changing. Nevertheless, if driving through an intersection, the long-term travel time and energy consumption depend more on the interaction between current vehicle status and traffic signal status. Thus, in order to figure out the optimal eco-driving solution in a realistic signalized intersection traffic situation, the most significant work is to build an RL model that can learn more from the long-term driving reward.

Although Dueling DQN framework is powerful in learning vision-based long-term policy, it is difficult to design an appropriate reward function for the eco-driving RL model. The main reason is the reward function is an instant value. On the contrary, the travel time and the total energy consumption can only be received when the journey is finished (i.e., ego-vehicle cannot know how many the total consumption is until it reaches the destination). Thus, the RL model will not work well or even don't work at all if the reward function is designed straight forward as usual (more information at the Experiments section).

In this study, we proposed a long-short term reward (LSTR) function, which not only considers the instance variables such as speed, lane change, and instant energy consumption but also includes some long-term based indicators. Algorithm 1 illustrates the LSTR function. The two conflicting factors in this issue are the short-term reward (instant speed, energy consumption) and the long-term reward (total travel time and total energy consumption). We designed some

instant reward principles which include indications for long-term benefit, which are shown below.

- When the current phase is red or yellow and ego-vehicle cannot pass the intersection with its current speed, then it shouldn't accelerate.
- When the current phase is red or yellow and ego-vehicle may pass the intersection with current speed or driving faster, then try to accelerate.
- When the current phase is green and the ego-vehicle cannot pass the intersection with its current speed, then it shouldn't accelerate.
- When the current phase is green and the ego-vehicle may pass the intersection with current speed or driving faster, then try to accelerate.

Basing on these principles, the LSTR function is designed as Algorithm 1 in Figure 10(a) in which the definition of RGreen-Pass is further explained by Figure 10(b). In a nutshell, the RGreen-Pass reflects the future benefit for reaching the intersection when the light is green.

Algorithm 1 Reward Function Descriptions.

Input:

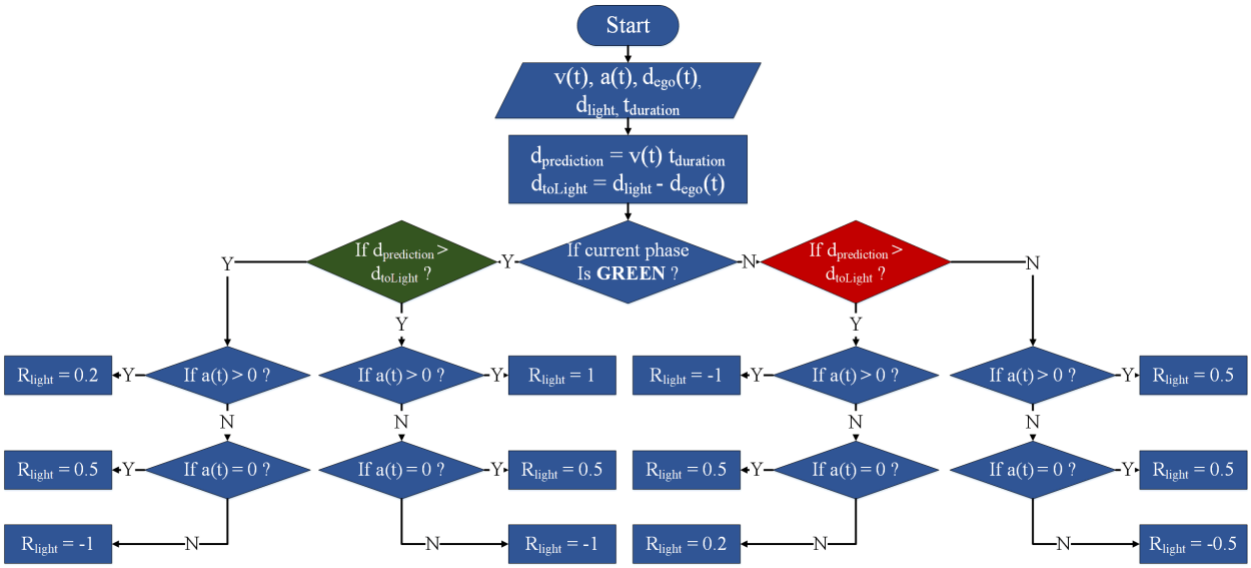
The current speed $v(t)$, acceleration $a(t)$ and current position d(t) of ego-vehicle and the curent phase P_c and the duration time t_d of the traffic light;

Output:

The reward value $R(t)$;

Initialization: $R = R_{velocity} = R_{lanechange} = R_{danger} = R_{energy} = R_{time} = R_{light} = 0$;
$d_{pridction} = v(t)t(d)$; $d_{toLight} = d_{total} - d(t)$
$R_{velocity}(t) = (v(t) - v_{min})/(v_{max} - v_{min})$;
if $a(t) > 0$; **then**
  $R_{energy}(t) = (-3.037 - 0.591v(t) - 0.001047v(t)^3 - 1.403v(t)a(t) = 0.02831v(t)^2 - 0.0798v(t)^2a(t) + 0.003535v(t)^3a(t) - 0.243v(t)a(t)^2)/130$;
else
  $R_{energy}(t) = 0$;
end if
$R_{time}(t) = R_{time}(t-1) + \delta t$
if Lane-change action happens; **then**
  $R_{lanechange}(t) = -0.1$;
end if
if Dangerous action happens; **then**
  $R_{danger}(t) = -0.5$;
end if
$R(t) = R_{velocity}(t) + R_{energy}(t) + R_{time}(t) + R_{lanechange}(t) + R_{danger}(t) + R_{light}(t)$
return $R(t)$;

(a) The algorithm for long-short term reward (LSTR) function



(b) The definition of RGreen-Pass is based on the four principles

Figure 10. Long-short term reward (LSTR) function

5.2.5. Network architecture

The main purpose of the deep neural network is mapping the current observation state *O* to the best action value A in action space. Thus, the main network can be divided into two components: (1) the hidden feature extraction network and (2) the policy network which applies Dueling DQN. Figure 11 illustrates the architecture of the proposed deep RL network.

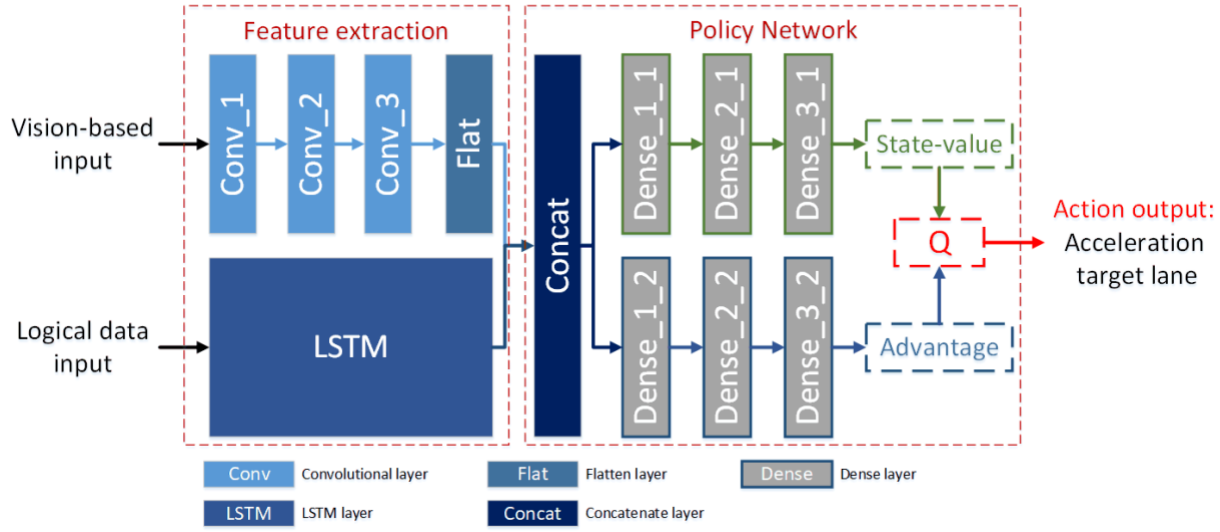


Figure 11. The architecture of the neural network

For the hidden feature extraction network, its main purpose is to extract hidden features from the preprocessed high-dimensional spatiotemporal data. As discussed above, the network input consists of (1) image flow and (2) high dimensional vector. In order to extract features from these different types of data, we designed a multi-channel feature extraction network, which consists of different kinds of parallel neural networks.

For the image flow input, the most popular method to extract image data is convolutional neural network (CNN), because the image is a spatial data format and the CNN has a tremendous capability of solving high-dimensional spatial data. Thus, in this study, a CNN stream is designed to extract the vision-based input data.

On the other hand, the radar, OBD, and V2I information (named as logical data) are integrated into a 13-dimension vector. After the selected-stack preprocess, the logical data are transformed into a time-series based temporal data. Long-short term memory (LSTM) network is famous for its powerful ability in dealing with temporal data series. Thus, for the logical data feature extraction, LSTM is applied in our deep RL net.

After the CNN and LSTM, the two-stream features are combined through a concatenate layer and then a dense layer-based policy network is designed by applying Dueling DQN. Figure 11 illustrates that there are two streams of the dense layer. The upper stream is used to extract state value while the lower stream is used to extract advantage value. Finally, the action with

maximum Q-value will be selected as the action at this moment. To be specific, the details of the configuration of the neural network is illustrated in Table 9.

Table 9. The description of the network configuration

Layer	Actuation	Patch size	Stride	Filter	Unit
Conv 1	ReLU	8 × 8	4	32	-
Conv 2	ReLU	4 × 4	2	64	-
Conv 3	ReLU	3 × 3	1	64	-
LSTM	-	-	-	-	1024
Dense 1 1/2	ReLU	-	-	-	1024
Dense 2 1/2	ReLU	-	-	-	256
Dense 3 1/2	ReLU	-	-	-	128

5.2.6. Network update and hyperparameters

There are four steps of the network updating process, which are:

1. Considering the current state as s_t and predicting the Q_t value of different actions through the evaluation network.
2. Choosing the action $a(t)$ with the largest Q value by utilizing e-greedy policy.
3. Generating the Q values at time $t + 1$: $Q(t+1)$ through the target network.
4. Calculating the loss function and then updating the evaluation network.

In addition, at each learning step, the weight coefficients of the proposed network were updated using the adaptive learning rate method Adam [24] in order to minimize the loss function. For the adopted hyperparameters, the learning rate \alpha, discount factor \gamma, batch size, steps used for observation, replay memory size, steps for target network update, training steps, and test steps are set as 0.00025, 0.99, 64, 10000, 50000, 10000, 2million, 10000 separately.

6. Simulation Study of the RL-based Connected Eco-driving

In order to train the proposed model and evaluate its performance, a simulator is constructed in this paper by utilizing Unity and Unity Machine-learning Agents (Unity ML-Agents). The intersection simulator is developed basing on previous work in [25, 26]. The main reason to use Unity to build the simulator is that Unity can provide a virtual reality environment in which the simulated camera can be applied. In addition, Unity ML-Agents provides a machine learning development platform in which the machine learning algorithms can be constructed readily. For the external RL brain, the deep RL algorithm and deep neural network are developed basing on Tensorflow [27] by Python.

As is discussed in the previous section, the testbed in this study is built as a one-direction intersection with 5 lanes. There are five different kinds of human-driven vehicles. The length of the research area is 550 meters: from 500 meters upstream of the intersection to 50 meters after the stop line. The vehicle speed limit is set to 50 kilometers per hour (kph). For the traffic light, the time for the green phase is 20 seconds, the time for the yellow phase is 2 seconds and the time for the red phase is 41s.

6.1. Experiment setup

For the training procedure, the $\varepsilon - greedy$ policy is implemented as the exploration policy. The ε is decreased linearly from 1 to 0.00001 over 2 million steps. When the training start, the initial phase and time of the traffic light will be randomly selected, which not only avoids the overfitting of the algorithm but also makes the training more realistic. In addition, the simulator is equipped with Intel® Core™ i7-7700k CPU @ 4.20GHz, 64 GB RAM, and an NVIDIA GTX 1080 GPU. The total training time is around 36 hours.

For the test procedure, there are three baselines implemented to compare with the proposed HRL framework, which are briefly introduced below:

1. **IDM method**: the IDM-based control model, which means the ego-vehicle is totally controlled by IDM (i.e., like the other vehicles);
2. **Short-Sighted (SS) method**: only considering short-term benefit in the reward function, i.e., without R_{Time} and R_{Light};
3. **Speed-First (SF) method**: only considering speed efficiency in the reward function, i.e., without R_{Time}, R_{Light} and R_{Energy}.

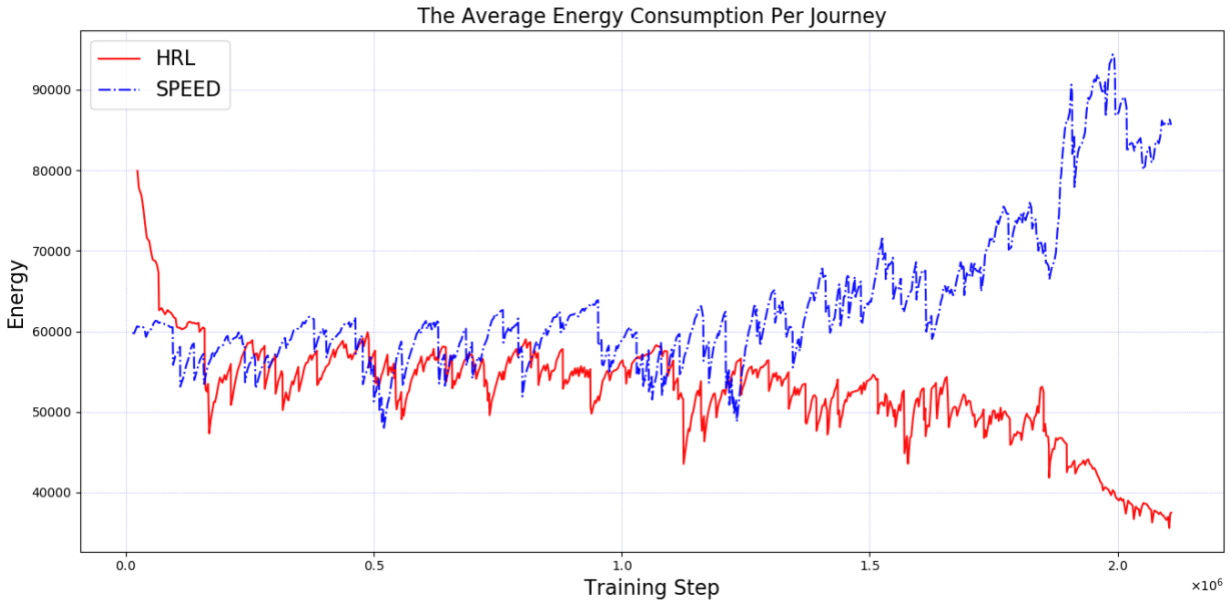
The proposed framework and each of the aforementioned baselines are tested through numerical experiments. As is illustrated in Figure 13 (shown in the next section), six different entry time in a cycle are tested: the 0th second of the cycle (C0), the 10th second of the cycle (C10), the 20th second of the cycle (C20), the 30th second of the cycle (C30), the 40th second of the cycle (C40) and the 50th second of the cycle (C50). Furthermore, different initial speeds from 10 kph to 50 kph with 10 kph as the increment (S10, S20, S30, S40, S50) are also tested. The training and numerical test results are discussed in the next section.

6.2. Training results

Figure 12(a)(b)(c)(d) illustrate the training results of HRL method and SF method, including the average energy consumption, traffic-light interaction reward, average speed and average lane-changing number in a single journey. Figure 12(a) shows that the energy consumption of HRL method is decreasing via the iteration of training while the energy consumption of SF is increasing, as the SF method is trying to get higher instant speed without considering the energy consumption. In addition, the training results of the SS method is not shown below, because during the iteration of training the ego-vehicle will drive more and slower and finally stop in the road, which makes the training have to stop. We realize that this may be because the confliction of two short-term rewards pushes the learning to fall into one side of the conflict factors (i.e., in this case, the short-term energy factor is stronger than speed, the ego-vehicle will not accelerate any more).

Basing on the above discussion, we realize that the Figure 12(b) can help to further explain why the HRL method can successfully learn an optimal policy in such a dilemma. Figure 12(b) shows that after the approximately half way of training steps, the traffic-light interaction reward is increasing obviously, which means that the ego-vehicle is learning more about how to interact with the traffic light intelligently (i.e., balancing the speed and energy consumption). On the other hand, it is also obvious that pursuing a speed-first driving strategy will actually cause a negative effect on the cooperation between vehicles and intersection.

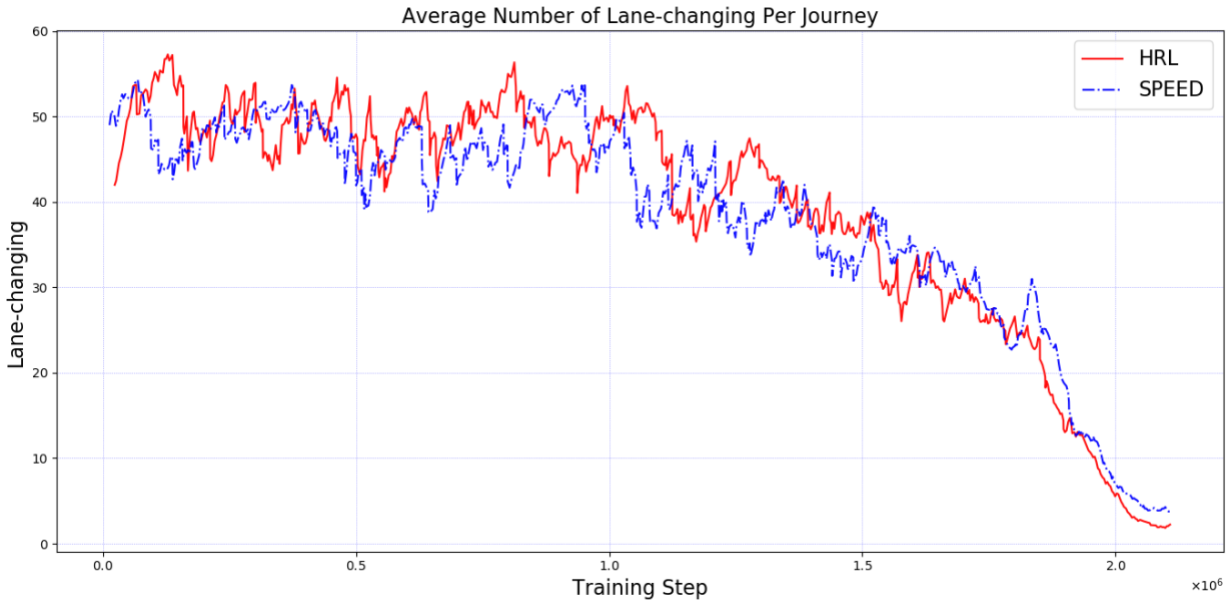
Figure 12(c) and 12(d) shows that the HRL method can also learn how to drive faster with less unnecessary lane-changing behaviors, which represents a smoothly, time-efficient driving strategy.



(a) The average energy consumption per journey during the training for HRL and SF model.

(b) The traffic-light interaction reward per journey during the training for HRL and SF model.

(c) The average speed per journey during the training for HRL and SF model.



(d) The average number of lane changing per journey during the training for HRL and SF model.

Figure 12. The training results

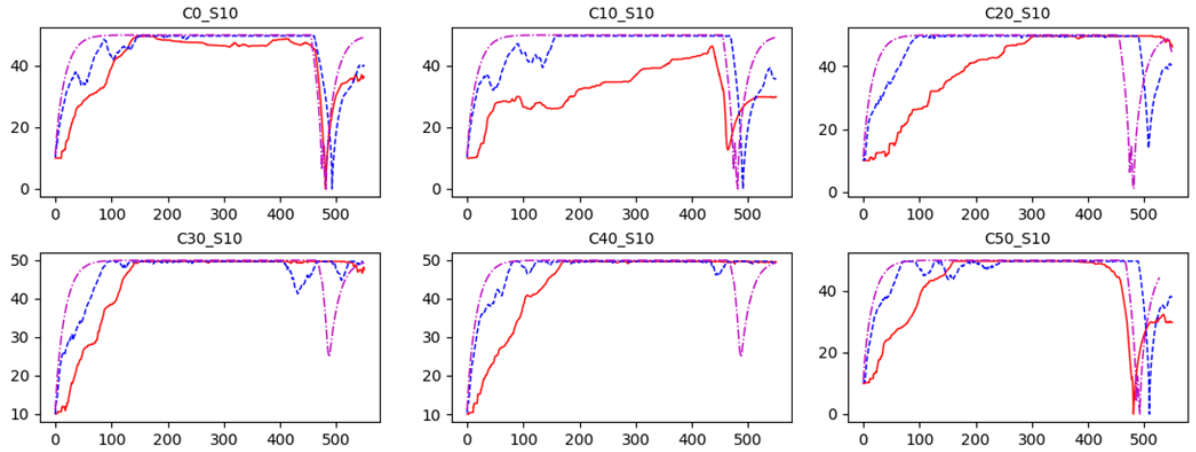
6.3. Testing results

Figure 13(a)(b)(c)(d)(e) illustrates the testing results of the comparison for speed trajectories of HRL, IDM and SF methods with different entry speed and initial phase time. In all subfigures, the red line, purple line and the blue line represent the results of our HRL method, IDM method and SF method separately. The Y-axis and X-axis represent the instant speed (kph) and current distance (meter). The title of each subfigure represents the initial phase time and entry speed of ego-vehicle. For example, the C0_S10 represents the initial phase time is 0s in the signal cycle and the entry speed is 10 kph. According to the testing results, there are three points we want to discuss:

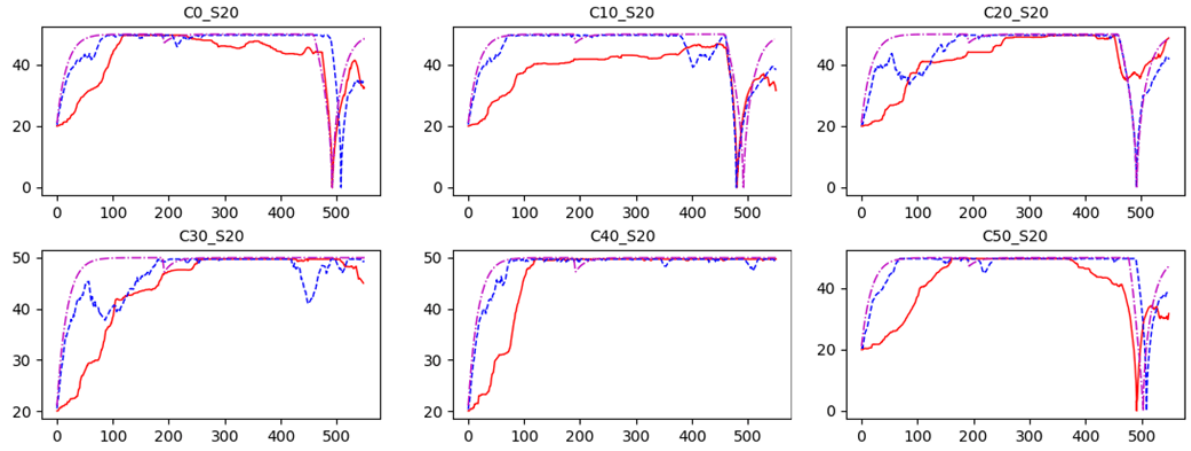
(1) The acceleration: it is evident that the acceleration value of HRL is obviously lower than either IDM or SF in almost all scenarios. Lower acceleration value is better for eco-friendly driving strategy, because, according to the energy consumption model, the energy consumption will go much higher if the acceleration increases. This demonstrates that the HRL model can drive in a more energy-efficient strategy.

(2) The target speed: for IDM and SF model, the target speed is always the maximum speed of the vehicle. However, for HRL model, it will not achieve the highest speed in some situations, such as C10_S10, C20_S10 and C20_S20, because the HRL model learns that in such a situation, a lower speed can gain more benefit in the long run (i.e., waiting for the end of red light so that ego-vehicle can pass without stops, such as C20_S10 and C20_S30). This evidently shows that the HRL can drive in a higher travel-efficient way than IDM and SF method (i.e., better interact with the intersection).

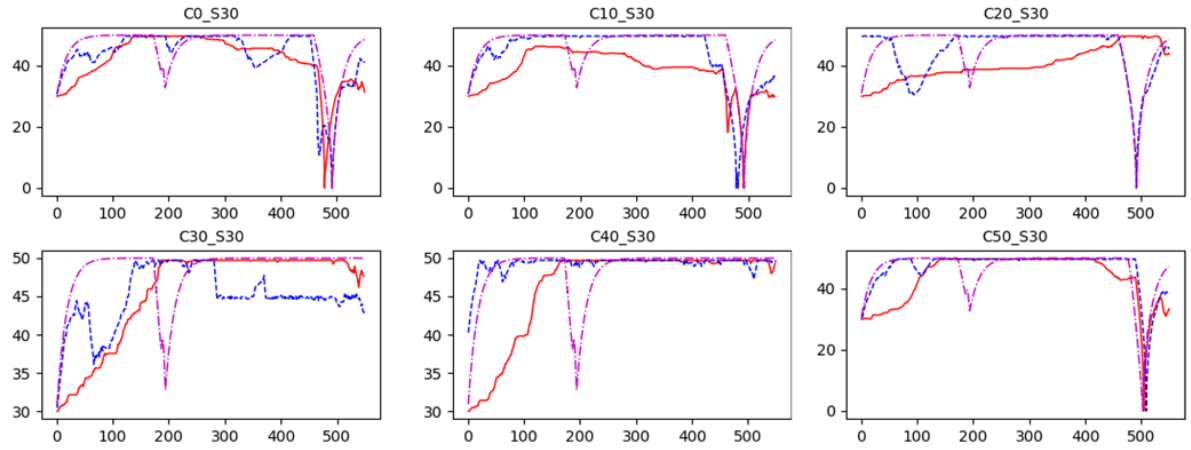
(3) The interaction with the mixed traffic: for the IDM model, it can only longitudinally interact with its front vehicle. For SF and HRL model, they can interact traffic from both longitudinal level and latitudinal level. Thus, in some situation, such as C40_S10, C0_S50, and C40_S40, the HRL and SF performed much better than IDM. In fact, due to the lower acceleration preference, HRL performed even better than SF, such as C30_S30, C20_S30, and C30_S50. This point illustrates that the HRL model can better interact with mixed traffic.



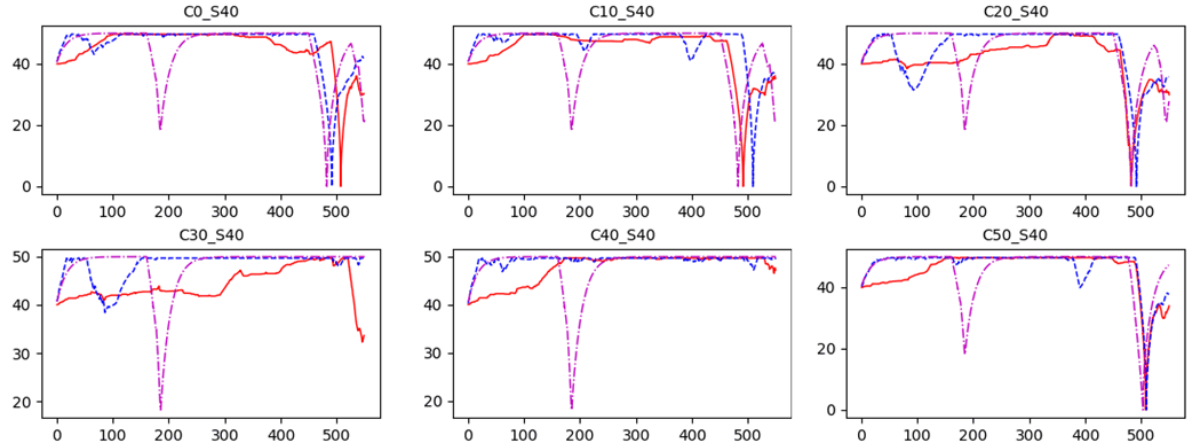
(a) The comparison of speed trajectories with 10 kph entry speed (S10).



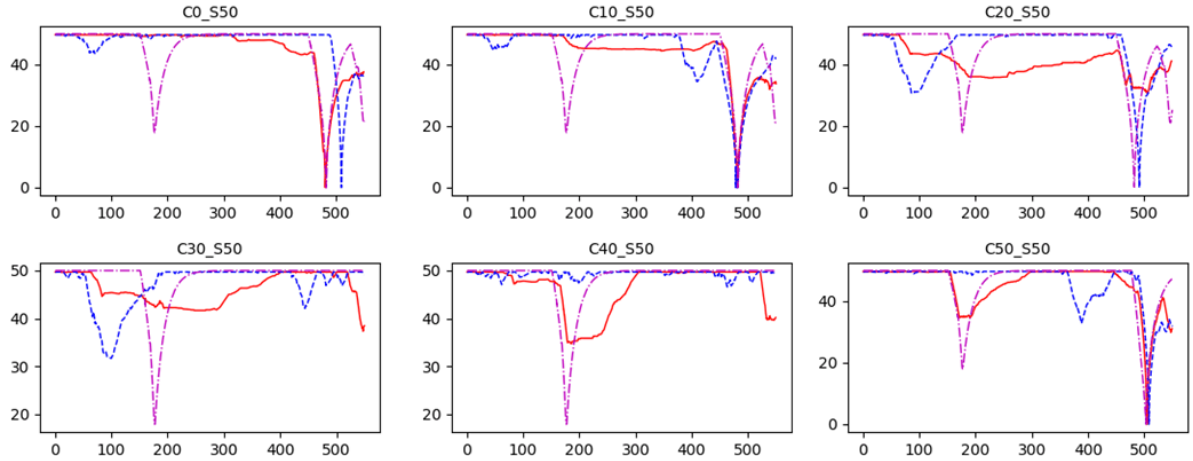
(b) The comparison of speed trajectories with 20 kph entry speed (S20).



(c) The comparison of speed trajectories with 30 kph entry speed (S30).



(d) The comparison of speed trajectories with 40 kph entry speed (S40).



(e) The comparison of speed trajectories with 50 kph entry speed (S50).

Figure 13. The speed trajectories of simulation experiments

Table 10. The average time per travel for three methods shows the comparison for average travel time of HRL, IDM and SF method. According to the table, the HRL can achieve nearly the

same performance to the SF method and better performance (1.12% in average) than IDM method. It is noticed that when the initial phase time is C20, the HRL has the relatively best time performance, which is 7.58% better than IDM method.

Table 10. The average time per travel for three methods

Methods	**S10/C0**	**S10/C10**	**S10/C20**	**S10/C30**	**S10/C40**	**S10/C50**	**Average**
HRL	71.4	62.9	41.9	45.9	47.3	79.7	58.2
SF	69.7	62.4	44.8	44.2	43.2	80.2	57.4
IDM	70.7	62.2	51.2	45.2	44.9	81.5	59.3
Methods	**S20/C0**	**S20/C10**	**S20/C20**	**S20/C30**	**S20/C40**	**S20/C50**	**Average**
HRL	70.2	62.4	48.9	45.9	42.7	80.1	58.4
SF	67.9	63.1	52.6	43.1	41.4	79.6	58.0
IDM	69.6	61.5	49.7	42.5	40.72	80.3	57.4
Methods	**S30/C0**	**S30/C10**	**S30/C20**	**S30/C30**	**S30/C40**	**S30/C50**	**Average**
HRL	71.5	60.8	44.1	42.1	41.1	79.5	56.5
SF	70.2	61.6	51.1	44.2	40.3	79.3	57.8
IDM	71.1	60.2	50.1	44.7	41.9	79.4	57.9
Methods	**S40/C0**	**S40/C10**	**S40/C20**	**S40/C30**	**S40/C40**	**S40/C50**	**Average**
HRL	67.9	61.3	53.5	43.7	40.1	79.5	57.7
SF	72.2	58.18	53.1	40.5	40.2	79.3	57.2
IDM	72.3	61.5	52.1	42.4	41.9	79.1	58.2
Methods	**S50/C0**	**S50/C10**	**S50/C20**	**S50/C30**	**S50/C40**	**S50/C50**	**Average**
HRL	70.6	61.7	48	41.5	41.9	79.9	57.3
SF	67.5	61.8	52.4	42.5	39.6	79.8	57.3
IDM	72.1	61.5	52.7	42.2	42.7	79.9	58.5
HRL_Avg	70.3	61.8	47.3	43.8	42.6	79.7	57.6
SF_Avg	69.5	61.4	50.8	42.9	40.9	79.6	57.5
IDM_Avg	71.2	61.4	51.2	43.4	42.4	80.0	58.3
HRLtoIDM	-1.18%	0.72%	-7.58%	0.97%	0.46%	-0.37%	-1.13%
SFtoIDM	-2.33%	0.06%	-0.70%	-1.15%	-3.50%	-0.50%	-1.25%

Table 11 shows the comparison of the average energy consumption of a single journey. It is obvious that the proposed HRL method can save energy in all the different situations. Due to the better performance in acceleration control, target speed control and interaction with mixed traffic, the HRL method can save 12.2% (up to 33.2% under S30_C20) energy comparing with IDM method and can save 47.1% energy comparing with SF method.

In addition, according to Table 11, when the initial phase time is C20 (i.e., the start of the yellow light), the HRL method has the relatively best average performance. In this situation, the average improvement is 24%, even comparing to the IDM method, which is a tremendous enhancement. On the other hand, when the initial phase time is near-zero or the entry speed is too fast, the improvement is only near 3% comparing to IDM method. According to this analysis, we realize that the initial phase time and entry speed will influence the performance of HRL method, which reminds us that there is an adjustment space of the HRL method. Different traffic situations have different adjustment space for ego-vehicle and if we can control the vehicle to enter the intersection with proper adjustment space, the HRL-based eco-driving approach will get its best performance.

Table 11. The average energy consumption per travel for three methods

methods	S10/C0	S10/C10	S10/C20	S10/C30	S10/C40	S10/C50	Average
HRL	43488	37870	38460	33576	33871	41295	38093.33
SF	77105	78726	88713	79917	76013	84277	80791.83
IDM	44790	41728	40106	36272	35949	43570	40402.5
Saving	-2.91%	-9.25%	-4.10%	-7.43%	-5.78%	-5.22%	-5.72%
methods	S20/C0	S20/C10	S20/C20	S20/C30	S20/C40	S20/C50	Average
HRL	43307	36302	28202	27054	27095	42636	34099.33
SF	76061	82930	78222	69645	76947	80955	77460
IDM	45847	43190	41212	29100	29400	45297	39007.67
Decrease	-5.54%	-15.95%	-31.57%	-7.03%	-7.84%	-5.87%	-12.58%
Methods	S30/C0	S30/C10	S30/C20	S30/C30	S30/C40	S30/C50	Average
HRL	43769	35008	28618	27511	29733	43926	34760.83
SF	77599	83342	82494	59857	78600	85573	77910.83
IDM	46331	43714	42855	31841	30844	46812	40399.5
Decrease	-5.53%	-19.92%	-33.22%	-13.60%	-3.60%	-6.17%	-13.96%
methods	S40/C0	S40/C10	S40/C20	S40/C30	S40/C40	S40/C50	Average
HRL	39454	33745	30168	27570	33597	40522	34176
SF	79545	87708	85055	78179	77700	88732	82819.83
IDM	46565	45047	43671	30632	30548	46879	40557
Decrease	-15.27%	-25.09%	-30.92%	-10.00%	9.98%	-13.56%	-15.73%
methods	S50/C0	S50/C10	S50/C20	S50/C30	S50/C40	S50/C50	Average
HRL	39172	34786	31727	29344	32330	42539	34983
SF	85188	88754	84358	72231	66488	91066	81347.5
IDM	43008	41489	39621	36878	37163	43828	40331.17
Decrease	-8.92%	-16.16%	-19.92%	-20.43%	-13.00%	-2.94%	-13.26%
Average	-7.63%	-17.27%	-23.95%	-11.70%	-4.05%	-6.75%	-12.25%

7. Conclusion

This research proposes an adaptive strategy for connected eco-driving towards a signalized intersection under real world conditions including uncertain traffic and actuated signal condition. A graph-based model is created with nodes representing dynamic states of the host vehicle (distance to intersection and current speed) and indicator of queue status or signal status and directed edges with weight representing expected energy consumption between two connected states. Then a dynamic programing approach is applied to identify the optimal speed for each vehicle-queue-signal state iteratively from downstream to the upstream. The uncertainty can be addressed by formulating stochastic models when describing the transition of queue-signal state. For uncertain traffic conditions, numerical simulation results show an average energy saving of 9%. It also indicates that energy consumption of a vehicle equipped with adaptive EAD strategy and a 100m-range sensor is equivalent to a vehicle with conventional EAD strategy and a 190m-range sensor. To some extent, the proposed strategy could double the effective detection range in eco-driving. For the actuated signals, the numerical simulations with real world SPaT data show that the proposed method is robust and adaptive to varying signal conditions, and achieves 40% energy savings when the vehicle arrives in the red time, and 8.5% energy savings when the vehicle arrives in the green time compared to other baseline methods.

Besides the adaptive eco-driving strategy based on historical traffic and signal data, we also consider the sensor-only based approach when the historical data are not available. An multi-sensor based eco-driving strategy for CAVs under uncertain traffic is proposed under a hybrid reinforcement learning (HRL) framework. According to the microsimulation experiments, the proposed HRL-based ego-vehicle can traverse trough a signalized intersection with eco-driving strategy under mixed traffic conditions. The HRL method can reduce 12.25%–47.1% energy consumption comparing with IDM and SF method and can save 1.2%–6.9% time comparing with IDM. The proposed framework can also be readily implemented to other types of vehicles by replacing the energy-reward function and vehicle dynamic model.

Regarding future work, the performance of the different types of vehicles (e.g., heavy-duty trucks) can be tested and analyzed. In addition, cooperative eco-driving strategy can be conducted by applying multi-vehicle agents in complex traffic network, such as a combination of uncertain traffic condition and actuated signals along a signalized corridor. Furthermore, more experiments including micro-simulation and field experiments can be conducted to analyze the performance in more realistic situations.

References

[1] U.S. Environmental Protection Agency (EPA), “Fast Facts: U.S. Transportation Sector GHG Emissions 1990-2016”, https://nepis.epa.gov/Exe/ZyPDF.cgi?Dockey=P100USI5.pdf. Jan. 2019

[2] U.S. Department of Energy, “Transportation energy data book”, https://cta.ornl.gov/data/tedbfiles/Edition36_Full_Doc.pdf. Jan. 2019

[3] European Commission (EC), “eCoMove – Cooperative Mobility Systems and Services for Energy Efficiency”, http://www.ecomove-project.eu/. Jan. 2019.

[4] U.S. Department of Transportation, “Applications for the Environment: Real-Time Information Synthesis (AERIS)”, https://www.its.dot.gov/research_archives/aeris/index.htm. Jan 2019

[5] M. Li, K. Boriboonsomsin, G. Wu, W. Zhang, and M. J. Barth, “Traffic energy and emission reductions at signalized intersections: A study of the benefits of advanced driver information,” *Int. J. ITS Res.*, vol. 7, no. 1, pp. 49–58, Jun. 2009.

[6] M. J. Barth., S. Mandava, K. Boriboonsomsin, and H. Xia, “Dynamic ECO-driving for arterial corridors”. *Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on*, pp.182-188, June 29 2011-July 1 2011.

[7] M. Li, K. Boriboonsomsin, G. Wu, W. Zhang, and M. J. Barth, “Traffic energy and emission reductions at signalized intersections: A study of the benefits of advanced driver information,” *Int. J. ITS Res*., vol. 7, no. 1, pp. 49–58, Jun. 2009.

[8] P. Hao, G. Wu, K. Boriboonsomsin and M. J. Barth, "Eco-Approach and Departure (EAD) Application for Actuated Signals in Real-World Traffic," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 30-40, Jan. 2019.

[9] X. He, H. X. Liu, and X. Liu, “Optimal vehicle speed trajectory on a signalized arterial with consideration of queue,” *Transp. Res. C, Emerg. Technol*., vol. 61, pp. 106–120, Dec. 2015.

[10] Y. Fei, P. Hao, X. Qi, G. Wu, K. Boriboonsomsin, and M. Barth, “Prediction-based eco-approach and departure at signalized intersections with speed forecasting on preceding vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, 20 (4), 1378-1389.

[11] P. Hao, K. Boriboonsomsin, C. Wang, G. Wu, and M. Barth, “Connected eco-approach and departure (EAD) system for diesel trucks,” *Proceedings of the 97th Annual Meeting of Transportation Research Board*, Washington, DC, 2018

[12] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*. 1: 269–271, 1959.

[13] P. Hao, K. Boriboonsomsin, G. Wu, Z. Gao, T. LaClair and M. Barth “Deeply Integrated Vehicle Dynamic and Powertrain Operation for Efficient Plug-in Hybrid Electric Bus,” *Proceedings of the 98th Annual Meeting of Transportation Research Board*, Washington, DC, 2019.

[14] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, p. 436, 2015.

[15] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. *MIT press*, 2018.

[16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, and Lanctot, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, p. 529, 2015.

[18] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, “Deep reinforcement learning framework for autonomous driving,” *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[19] C. Desjardins and B. Chaib-Draa, “Cooperative adaptive cruise control: A reinforcement learning approach,” *IEEE Transactions on intelligent transportation systems*, vol. 12, no. 4, pp. 1248–1260, 2011.

[20] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “Safe, multi-agent, reinforcement learning for autonomous driving,” arXiv preprint arXiv:1610.03295, 2016.

[21] J. Chen, Z. Wang, and M. Tomizuka, “Deep hierarchical reinforcement learning for autonomous driving with distinct behaviors,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1239–1244.

[22] F. Ye, G. Wu, K. Boriboonsomsin, and M. J. Barth, “A hybrid approach to estimating electric vehicle energy consumption for ecodriving ap- plications,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 719–724.

[23] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and De Freitas, “Dueling network architectures for deep reinforcement learning,” pp. 1995–2003, 2015.

[24] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” Computer Science, 2014.

[25] Z. Bai, B. Cai, W. Shangguan, and L. Chai, “Deep reinforcement learning based high-level driving behavior decision-making model in heterogeneous traffic,” arXiv preprint arXiv:1902.05772, 2019.

[26] K. Min and H. Kim, “Deep q learning based high level driving policy determination,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 226–231.

[27] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, “Tensorflow: a system for large- scale machine learning,” 2016.

Data Management Plan

Products of Research

In this project, we collected vehicle speed trajectories and energy consumptions data for all the host vehicles from all the numerical and micro-simulation experiment. Those data are used to validate the proposed algorithms and estimate the performance on energy savings.

Data Format and Content

The data were saved in CSV files in the format of second-by-second trajectories. For each time stamp, the vehicle's dynamic state, e.g., location, speed and acceleration rate, the signal timing information and the traffic information are archived along with the estimate energy consumption calculated by the specific models for gasoline vehicles or electric vehicles.

Data Access and Sharing

The data are publicly available via Dryad: https://datadryad.org/stash/, which is in compliance with the USDOT Public Access Plan. This dataset can be cited as:

> Hao, Peng; Wei, Zhensong; Barth, Matthew (2019), Speed trajectory data from adaptive eco-driving applications, UC Riverside, Dataset, https://doi.org/10.6086/D11H3P

Reuse and Redistribution

The data are restricted to research use only. If the data are used, our work should be properly cited: Hao, Peng; Wei, Zhensong; Barth, Matthew (2019), Speed trajectory data from adaptive eco-driving applications, UC Riverside, Dataset, https://doi.org/10.6086/D11H3P.