Automotive Innovation (2020) 3:374–385
https://doi.org/10.1007/s42154-020-00113-1

Deep Reinforcement Learning Enabled Decision-Making for Autonomous Driving at Intersections

Guofa Li[1,2] · Shenglong Li[1] · Shen Li[3] · Yechen Qin[4] · Dongpu Cao[2] · Xingda Qu[1] · Bo Cheng[5]

Received: 14 January 2020 / Accepted: 21 July 2020 / Published online: 13 November 2020

Abstract

Road intersection is one of the most complex and accident-prone traffic scenarios, so it's challenging for autonomous vehicles (AVs) to make safe and efficient decisions at the intersections. Most of the related studies focus on the solution to a single scenario or only guarantee safety without considering driving efficiency. To address these problems, this study proposed a deep reinforcement learning enabled decision-making framework for AVs to drive through intersections automatically, safely and efficiently. The mapping relationship between traffic images and vehicle operations was obtained by an end-to-end decision-making framework established by convolutional neural networks. Traffic images collected at two timesteps were used to calculate the relative velocity between vehicles. Markov decision process was employed to model the interaction between AVs and other vehicles, and the deep Q-network algorithm was utilized to obtain the optimal driving policy regarding safety and efficiency. To verify the effectiveness of the proposed decision-making framework, the top three accident-prone crossing path crash scenarios at intersections were simulated, when different initial vehicle states were adopted for better generalization capability. The results showed that the developed method could make AVs drive safely and efficiently through intersections in all of the tested scenarios.

Keywords Autonomous vehicles · Driving safety and efficiency · Intersection · Decision-making · Deep reinforcement learning

Abbreviations

AV	Autonomous vehicle
DQN	Deep Q-network
DRL	Deep reinforcement learning
LTAP/LD	Left turn across path-lateral direction
LTAP/OD	Left turn across path-opposite direction
MDP	Markov decision process
OV	Other vehicle
SCP	Straight crossing path
V2I	Vehicle-to-infrastructure
V2V	Vehicle-to-vehicle

✉ Shen Li
sli299@wisc.edu

Guofa Li
hanshan198@gmail.com; guofali@szu.edu.cn

1 Institute of Human Factors and Ergonomics, College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China

2 Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada

3 Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

4 School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

5 State Key Laboratory of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China

1 Introduction

Autonomous driving is a topic widely concerned by scientific research institutions and enterprises because of its great potential in changing existing mobility and, most importantly, reducing the occurrence of traffic accidents. Among the various scenarios in autonomous driving, intersection has been regarded as the most challenging scenario because of the complex traffic environment [1, 2]. Traffic crash statistics show that intersections are related with 60% of severe traffic injuries in Europe [3, 4]. As per the officially published statistics in 2019 in the USA, automotive fatal crashes related with intersections account for 29% of all the car crashes, leading to 18% of pedestrian fatalities [5].

The situation is even worse for older drivers [5]. Therefore, developing driving strategies for autonomous vehicles (AVs) at intersections is important for safety enhancement [6].

Previous studies on AVs driving strategies mainly focus on making decisions based on motion prediction, collision risk assessment, and collaborative decision-making to avoid collisions by applying vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) technologies [7]. However, these developed strategies are usually tested in a single scenario without considering the application to different intersection scenarios. Moreover, unlike camera systems [8, 9], V2V and V2I technologies require high costs of infrastructure, equipment installation and maintenance, which limits the application of these proposed methods [10, 11].

Besides driving safety, travel efficiency should also be improved by avoiding over-conservative decisions. Under the premise of not speeding and crashing, it will be more intelligent and efficient for AVs to pass intersections as fast as possible. However, most of the current studies mainly work to ensure driving safety with over-conservative decisions [12]. The resulting long waiting time in yielding maneuvers may be unacceptable in real driving.

To solve these problems, a deep reinforcement learning (DRL) method was developed, which could avoid collisions in various scenarios at intersections while maintaining high travel efficiency. The mapping relationship between local observations (traffic images) and AVs actions (vehicle operations) was obtained by using an end-to-end decision-making framework. The top three accident-prone scenarios at intersections [13] were developed in Carla (a simulator) to see the effectiveness of the proposed method. The contributions of this study are summarized as follows:

(1) A novel DRL-enabled decision-making framework is proposed to ensure driving safety and travel efficiency for autonomous driving at intersections.
(2) The observation state of AVs from cost-effective images acquired at different moments is proposed to make more reasonable driving decisions.
(3) The generalization capability of the proposed method is verified in three different crash scenarios.

This paper is structured as follows. Section 2 reviews the related literature about AVs decision-making. Section 3 describes the key elements in the reinforcement learning (RL) model for autonomous driving at intersections. The proposed DRL-based decision-making framework is presented in Sect. 4. The test details are introduced in Sect. 5. The test results and discussion are shown in Sect. 6. The final Sect. 7 gives the conclusions.

2 Literature Review

The existing literature about AVs decision-making solutions mostly focuses on motion prediction, collision risk assessment, and collaborative decision-making. These three categories are reviewed in detail as follows.

2.1 Decision-Making Based on Motion Prediction

The motion-prediction-based decision-making methods generally make decisions according to the prediction of a vehicle's future movement, which can be interpreted from the vehicle's kinematics and dynamics [14–17]. For example, Ref. [14] proposed a model-based motion prediction method to estimate how the vehicle would avoid collision by parameterizing the potential evasive maneuvers. It was concluded that the developed algorithm could be used to help drivers avoid or mitigate collisions with a lower computational cost. A probabilistic motion prediction algorithm was proposed in Ref. [17] based on an unscented Kalman filter to assess possible conflicts and make decisions to ensure safety. Simulation and test results showed that the algorithm could effectively detect and avoid future frontal collisions at intersections.

2.2 Decision-Making Based on Collision Risk Assessment

The collision-risk-assessment-based decision-making methods can be categorized into physical model-based and data-driven-based methods. In physical model-based methods, collision risk assessment and decision-making are developed through the mathematical analysis of the probability of collision drawn from physical insights and models. Time-to-collision (TTC) is a representative physical model-based index and is frequently used as an indicator to enable a warning or an intervention during the decision-making process [18]. In Ref. [19], the authors used a combination of TTC and inter-vehicle-time to get a more comprehensive and accurate understanding of driving risk. The results showed that this combination enabled AVs to avoid collisions. Probabilistic models (another physical-model-based method) are also frequently used to evaluate driving risk and make decisions based on a system's uncertainties and incomplete information [20–23]. These models generally computed how likely the collision with other vehicles would occur in the near future under the assumptions of uncertainties (e.g., dynamic modeling errors, sensor noise, and driver intention misunderstanding). For example, Ref. [21] proposed an online decision-making framework by utilizing motion planning technologies that embedded uncertainty predictors of other road users based on partially observable Markov decision

process (MDP). The proposed framework was able to comprehensively observe uncertainties and provide optimized strategies for traveling through intersections. In Ref. [23], the Bayesian framework for decision-making was developed to deal with stochastic risk assessment. The results showed that the collision risk could be calculated online by using a real-time system for collision mitigation.

Data-driven-based decision-making methods use machine learning algorithms to assess collision risk and make decisions accordingly [24–28]. The main idea is to map the system's state to a risk level by utilizing neural networks. References [24, 25] developed a Gaussian-process-based method to learn driver's behavior for decision-making at intersections. The results showed that driving decisions could be optimized by predicting the driver's intention to keep safe when driving through intersections. A random forest algorithm was used in Ref. [26] to learn how autonomous parking decisions were made. The test results showed that autonomous parking was effectively realized with high robustness.

2.3 Collaborative Decision-Making

The collaborative decision-making methods employ cooperative communications to share detailed information of local surroundings among vehicles and to make collaborative decisions for safety enhancement [29–33]. For example, Ref. [30] leveraged V2V technology to deploy decentralized algorithms for cooperative decision-making to avoid collisions at intersections, which implemented a control theoretic approach. The results showed that the proposed method effectively avoided collisions. Similarly, a cooperative vehicle intersection control system was developed in Ref. [32]; it utilized V2V and V2I technologies for decision inference under the assumption that all vehicles were fully automated. The results showed that the developed method achieved a shorter stop delay and travel time than the conventional systems while ensuring safety.

Differently, DRL obtains the optimal decision-making policy by maximizing the long-term rewards, which has been verified to be an effective emerging method in recent years [34–39]. Reference [34] proposed a decision-making method for obstacle avoidance based on deep deterministic policy gradient (DDPG). The results showed that the developed method could learn steering and acceleration operations directly to avoid collisions. A DDPG-based decision-making framework for lane following was proposed in Ref. [35]. This study carried out a real-world application to examine the effectiveness of the proposed framework. The results showed that AVs were able to learn lane-following behavior after less than 30 min of training. Ref. [39] proposed a DRL-enabled decision-making method for energy-efficient driving by implementing a deep Q-network with dueling structure. The evaluation results based on real-world driving data showed the proposed method achieved 16.3% fuel saving than the compared binary control strategy. All these studies have demonstrated that DRL can learn driving decisions effectively for the development of autonomous driving.

3 Elements in the RL Model for Autonomous Driving at Intersections

In this section, an RL model for autonomous driving at intersections is established. The RL process is shown in Fig. 1. An RL model comprises an agent that interacts with the surrounding environment according to the observations. The agent learns to choose optimal actions through interacting with traffic environment to maximize the reward function. The formulations of the key elements in the RL model are shown in detail in the following sections.

3.1 Markov Decision Process for Autonomous Driving at Intersections

Autonomous driving at intersections is regarded as an MDP [40]. MDP is a mathematical decision-making framework, which is described by (S, A, T, R, γ).

S: the state space. s_t is the state at time t ($s_t \in S$);
A: the action space. a_t is the action at time t ($a_t \in A$);
T: the transition model. It describes the transition probability from one state to another.
R: the reward function. R_t is the reward for deploying action a_t at state s_t.
γ: the discount factor, $\gamma \in [0,1]$. It is used to calculate the cumulative expected reward.

In RL model, the environment returns a numerical reward from a given reward function R based on the current state and the action that the agent takes. The goal of RL model is to learn an optimal strategy by maximizing the sum of discounted future rewards.

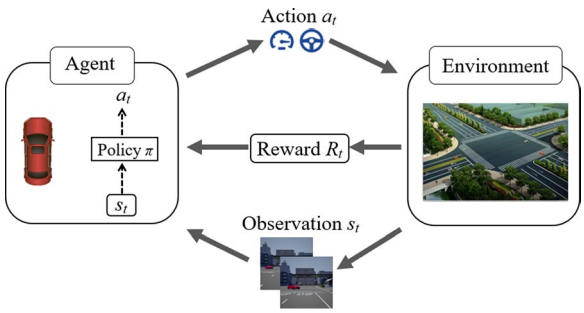

Fig. 1 RL model for autonomous driving at intersections

Intersection driving policy learning can also be considered as an MDP, in which AVs is an intelligent agent with self-learning ability to improve its behaviors through the interaction with other vehicles. The details of the above-mentioned elements are defined as follows for driving policy learning at intersections.

3.2 State Space

The observation definition is key establishing the state space without ambiguity. A monocular camera was used to define the observations. Front road environment can be obtained from images collected by the front camera mounted on the AVs. However, a single image only contains information about the distance between the AVs and other vehicles (OVs), which is not enough to clearly define the current state of AVs. To solve this, the relative velocity between AVs and OVs were used. Specifically, two images obtained at time t and t-2 (2 timesteps ago) were used to collect the current and previous observations, based on which the difference between these two images can be used to estimate the relative velocity between these two vehicles. Thus, the current state of AVs was obtained.

3.3 Action Space

When driving straight across an intersection, a vehicle can adjust throttle and brake operations to accelerate, decelerate, or keep a constant velocity to ensure safety according to the surrounding traffic environment. Therefore, the action space for intersection driving policy learning was defined as four velocity-related operations: throttle = 0.8, throttle = 0.65, brake = 0.2, and brake = 1.0, representing acceleration, driving at a constant velocity, gentle brake, and hard brake, respectively. The initial throttle value of AVs was set as 0.65.

3.4 Reward Function

Once an action is selected, an AV will get a resulting reward. The aim of the RL model about intersection driving is to learn an optimal driving policy by maximizing the expectation of the discounted future reward, which indicates that different reward functions will lead to different driving policies. Therefore, it is of great importance to appropriately design the reward function to guide AVs for better learning performance.

When an AV is straightly crossing an intersection, there will be a potential cross area which is identified as the collision area for AVs and OVs. See Fig. 2, D_t^{AVs} denotes the driving distance of AVs from the current location to the collision area; D_t^{OVs} denotes the driving distance of OVs from the current location to the immediate pass of the collision area at time t. The positions of vehicles and collision area are expressed by coordinates in Carla, and these coordinates can be used to approximately calculate D_t^{AVs} and D_t^{OVs}. In practical applications, the information of the variables adopted in this study can be obtained by employing precise digital map, V2V and V2I communication technologies, based on which most of the current decision-making approaches for AVs were developed [7].

The decision-making process should consider two requirements: (1) avoid collision with OVs, (2) do not be too conservative and travel as fast as possible. Based on these requirements, the reward function R_t is designed as follows:

$$R_t^{\mathrm{vel}} = \begin{cases} -1, & \text{if} \quad v_t^{\mathrm{AVs}} > v_{\max}^{\mathrm{AVs}} \text{ or } v_t^{\mathrm{AVs}} < v_{\min}^{\mathrm{AVs}} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$R_t^{\mathrm{sae}} = \begin{cases} \dfrac{v_t^{\mathrm{AVs}} - v_{\min}^{\mathrm{AVs}}}{v_{\max}^{\mathrm{AVs}} - v_{\min}^{\mathrm{AVs}}}, & \text{if} \quad t_{\mathrm{con}}^{\mathrm{AVs}} \le t^{\mathrm{AVs}} \\ d \times \exp\left(-\dfrac{\left(t^{\mathrm{AVs}} - t^{\mathrm{OVs}}\right)^2}{2\sigma^2} \right), & \text{if} \quad t_{\mathrm{con}}^{\mathrm{AVs}} > t^{\mathrm{AVs}} \end{cases} \tag{2}$$

$$R_t = R_t^{\mathrm{vel}} + R_t^{\mathrm{sae}} \tag{3}$$

where R_t^{vel} is the reward for not speeding; R_t^{sae} is the reward for driving safely and efficiently; v_t^{AVs} is the current velocity of AVs; $v_{\max}^{\mathrm{AVs}}$ is the maximal velocity that does not violate traffic rules; $v_{\min}^{\mathrm{AVs}}$ is the minimum acceptable velocity; t^{AVs} is the time AVs take to travel from the current position to the potential collision area; $t_{\mathrm{con}}^{\mathrm{AVs}}$ is the needed time of AVs when traveling at constant velocity v_t^{AVs}; t^{OVs} is the time OVs take to drive from the current position to the immediate pass of the potential collision area. If $t_{\mathrm{con}}^{\mathrm{AVs}} \le t^{\mathrm{AVs}}$, driving safety can be guaranteed; otherwise, the reward will be lower. If

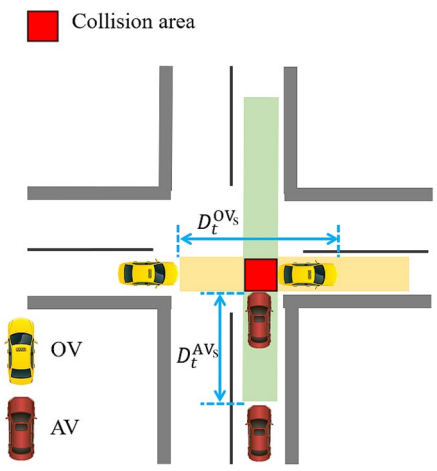

Fig. 2 Illustration of the collision area and variables in the reward function design

t_{con}^{AVs} is closer to t^{OVs}, the collision probability will be higher. Therefore, $d=-10$ is a manually set coefficient, used to highlight safety as the first priority. The uncertainty of the reward value σ is set as 0.5 [12]. The t^{AVs}, t_{con}^{AVs} and t^{OVs} are determined as follows:

$$t^{AVs}=\begin{cases}\dfrac{\sqrt{2ac_t^{AVs}D_t^{AVs}+v_t^{AVs^2}}-v_t^{AVs}}{ac_t^{AVs}}, & \text{if} \quad ac_t^{AVs}\neq 0\\ \dfrac{D_t^{AVs}}{v_t^{AVs}}, & \text{if} \quad ac_t^{AVs}=0\end{cases} \tag{4}$$

$$t_{con}^{AVs}=\frac{D_t^{AVs}}{v_t^{AVs}} \tag{5}$$

$$t^{OVs}=\frac{D_t^{OVs}}{v^{OVs}} \tag{6}$$

where ac_t^{AVs} is the current acceleration of AVs, and v^{OVs} is the velocity of OVs.

4 Deep Reinforcement Learning Algorithm

Deep Q-network (DQN) was developed based on Q-learning and has been frequently used in previous studies due to its good performance in solving decision-making problems with discrete actions [41]. This model-free DRL algorithm integrates the advantages of deep learning and RL to learn an optimal policy from high-dimensional data. In this study, DQN was used to train AVs to get the ability of driving safely and efficiently at intersections.

4.1 Deep Q-Network

Q-learning-based methods can be adopted to derive an optimal policy to maximize the sum of the discounted long-term rewards. The Q-value for a given state-action pair $Q(s, a)$ of policy π can be defined as follows:

$$Q_\pi(s,a)=E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k}\middle|s=s_t,a=a_t\right] \tag{7}$$

where γ is the discount factor; R_{t+k} is the reward at time $t+k$; k represents the timesteps after t; E is the mathematical expectation. $Q_\pi(s_t, a_t)$ is the accumulated discounted reward when taking action a_t at state s_t. It can be rewritten as

$$Q_\pi(s_t,a_t)=R_t+\gamma\sum_{s_{t+1}\in S}P_t\sum_{a_{t+1}\in A}\pi(a_{t+1}|s_{t+1})Q_\pi(s_{t+1},a_{t+1}) \tag{8}$$

where P_t is the probability of transition from the current state s_t to the next state s_{t+1} by taking action a_t, a_{t+1} is the next action for state s_{t+1}.

The optimal policy is determined by obtaining the optimal Q-value $Q^*(s_t, a_t)$ according to the Bellman optimal equation [42].

$$Q^*(s_t,a_t)=R_t+\gamma\sum_{s_{t+1}\in S}P_t\max_{a_{t+1}}Q^*(s_{t+1},a_{t+1}) \tag{9}$$

Based on Eq. (10), the state-action Q-value can be iteratively updated:

$$Q_{updated}(s_t,a_t)=Q_{current}(s_t,a_t)+\alpha(R_t+\gamma\max_{a_{t+1}}Q_{current}(s_{t+1},a_{t+1})-Q_{current}(s_t,a_t)) \tag{10}$$

where α is the learning rate. The polynomial multiplied by α is called temporal difference error which determines how much the Q-value changes when it is updated. The error will be close to zero for the optimal Q-value.

In MDP, if an agent takes all actions at all states for endless times, it will learn the optimal policy with an appropriately decaying learning rate α [43], indicating that Q-value will finally converge to the optimal Q*. The optimal policy π* can be obtained when the Q-value converges to the optimum.

$$\pi^*(a|s)=\begin{cases}1, & \text{if} \quad a=\arg\max_{a\in A}Q^*(s,a)\\ 0, & \text{otherwise}\end{cases} \tag{11}$$

The classical Q-learning algorithm can only obtain the optimal policy for small state space cases because the look-up table to update the Q-value of each element is limited in the state-action space. When the state space is large, this method is not applicable. The look-up table in this study is too large to be built because a large number of pixels in traffic images lead to a large amount of states, which makes the classical Q-learning method time-consuming and not applicable for real-time decision-making. To solve this problem, DQN was adopted. It uses convolutional neural network (CNN) to approximate the nonlinear relationship between states and Q-values of all the actions. See Fig. 3, $Q(s, a; \theta)$ can be estimated by Q-network using CNN with θ (the weights of the neural network); a_n is the action of AVs and the number of actions is n. The weights θ are updated in each iteration to minimize.

$$L(\theta)=E\left[(TargetQ-Q(s_t,a_t;\theta))^2\right] \tag{12}$$

where

$$TargetQ=R_t+\gamma\max_{a_{t+1}}Q(s_{t+1},a_{t+1};\theta^-) \tag{13}$$

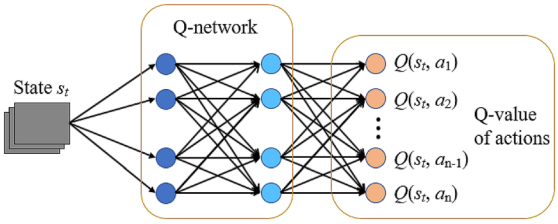

Fig. 3 The approximation of $Q(s, a)$ by CNN

TargetQ is obtained by the output of the target Q-network $Q(s_{t+1}, a_{t+1}; \theta^-)$. It should be noted that the structure of $Q(s_{t+1}, a_{t+1}; \theta^-)$ is the same as the current Q-network $Q(s_t, a_t; \theta)$. θ^- denotes the weights of target Q-network and is assigned by the value of θ every 200 iterations, while θ is updated in each iteration. By utilizing two networks to design loss function and update θ and θ^- in two different ways, the correlation between the target Q-network and the current Q-network will be weakened, which improves the algorithm stability during the learning process and makes $Q(s_t, a_t; \theta)$ converge to the optimum [44]. Finally, the optimal $Q^*(s, a)$ will be obtained and used to make an optimal decision by selecting the action strategy with the highest Q-value.

4.2 Network Architecture for Autonomous Decision-Making

The overall algorithm architecture of the DQN-based decision-making framework at intersections is illustrated in Fig. 4. The current state s_t is obtained from observations and then fed into online Q-network to get Q-values of all the action is made. The action a_t is decided by the ε-greedy strategy [45] to balance the ratio between exploration and exploitation. In this study, $\varepsilon = 0.9$, and $P(Q)$ is a random number between 0 and 1. Before the decision on action, $P(Q)$ will be randomly generated to decide how to choose an action. When $P(Q) < \varepsilon$, a_t is the most rewarding action. Otherwise, it is randomly chosen within the action space. The current image is obtained first, and once the action a_t is determined, agent will execute a_t and transit it to the next state. Then the next image can be obtained and the instant reward R_t for executing a_t at s_t will be determined. The current image m_t, next image m_{t+1}, action a_t, and reward R_t are saved into the replay memory as a tuple (m_t, a_t, R_t, m_{t+1}). The information in the replay unit will be used to generate mini-batches, e.g., $(m_{t-2}, m_t, a_t, R_t, m_{t-1}, m_{t+1})$, and then train the online Q-network and update parameters θ. The target network with θ^- is an independent neural network established to ensure the stability and convergence of the DQN algorithm.

4.3 Experience Replay and Training

As shown in Fig. 5, all the data generated by the interaction between AVs and OVs are accumulated in the replay memory. There are N_j $(0 \le j < M$, where M is the number of intersection driving events and j is the number of specific intersection driving event) driving episodes in the whole process of an intersection driving and N_j data tuples (m_t, a_t, R_t, m_{t+1}) are accumulated in a data list. When selecting samples from the replay memory for training, $Z_t^X (0 \le t < N_j, 0 \le X < M)$ is used to denote the selected sample every time, where X is the number of selected list and t means that the tuple (m_t, a_t, R_t, m_{t+1}) will be selected. X will be randomly decided first, and then t will be randomly chosen among X. As mentioned above, the observations

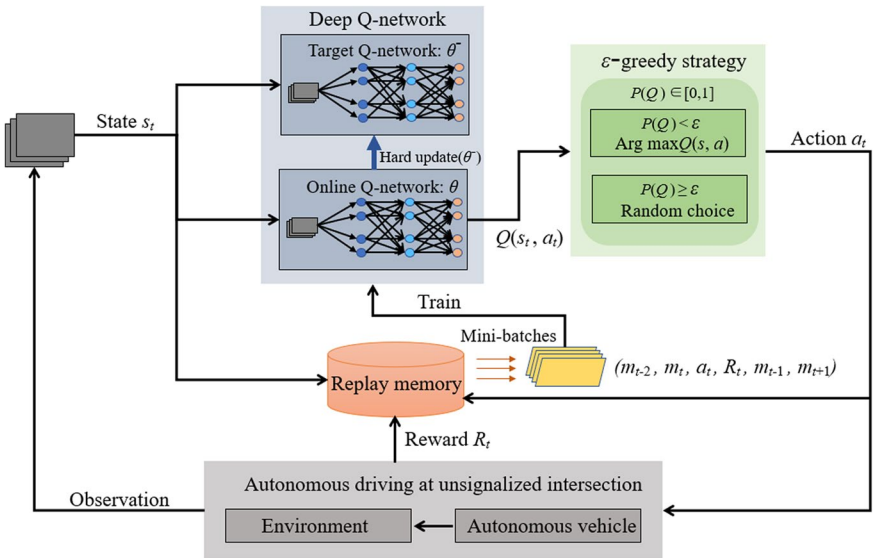

Fig. 4 The overall algorithm architecture of the DQN-based decision-making framework at intersections

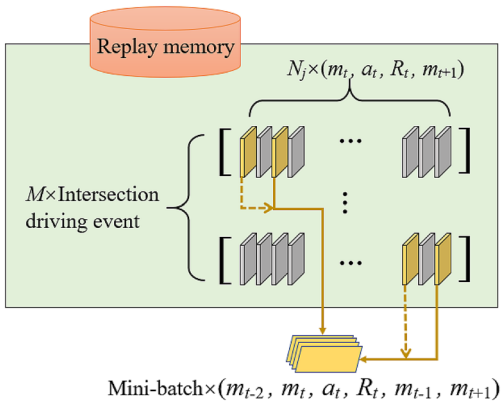

Fig. 5 Experience replay to obtain data for training

at the current timestep and two timesteps ago are used to construct the current state s_t of AVs. Therefore, a sample of mini-batch $(m_{t-2}, m_t, a_t, R_t, m_{t-1}, m_{t+1})$ is constructed by (m_t, a_t, R_t, m_{t+1}) in Z_t^X and (m_{t-2}, m_{t-1}) in Z_{t-2}^X, where Z_t^X and Z_{t-2}^X are in the same list. By developing such a sample extraction method, the temporal correlation between the extracted samples which are used to construct mini-batch is ensured. Thus, the authenticity and validity of mini-batch used for training are guaranteed. The size of data used for training is different from that of traditional DQN because m_{t-2} and m_t are combined to define the current state.

The proposed DQN structure is shown in Fig. 6. $\boldsymbol{FV}_{t-2}$ and $\boldsymbol{FV}_t$ are the convolution feature vectors and also the outputs of the same neural network with m_t and m_{t-2} as inputs. Then, $\boldsymbol{FV}_{t-2}$ and $\boldsymbol{FV}_t$ are concatenated together to obtain the concatenating feature vector which is further inputted into the fully connected layers to finally get the Q-values of all actions.

5 Intersection Scenarios Simulation Tests

According to the Intersection 2020 project in Europe and the traffic safety facts in the USA, straight crossing path (SCP), left turn across path-opposite direction (LTAP/OD), and left turn across path-lateral direction (LTAP/LD) are the top typical types of accidents at intersections [46–50]. These three types accounted for 35.7%, 27.3%, and 20.0% of all intersection crashes in Germany, respectively [46]. The numbers were 39%, 36%, and 23% in the USA, according to a crash causation analysis database of NMVCCS (national motor vehicle crash causation survey) [47]. Data from the national automotive sampling system/general estimates system (NASS/GES), the CDS (NASS crashworthiness data system), the fatality analysis reporting system (FARS), the German in-depth accident study (GIDAS), and the PCM (GIDAS-based pre-crash matrix) also supported that LTAP/LD, LTAP/OD, and SCP accounted for more than 70% of all intersection crashes [47, 50]. Hence, these three typical scenarios were selected for analysis.

As illustrated in Fig. 7, these three types of crossing path scenarios at intersections were created in Carla [51] to train the proposed DRL model. The velocity of AVs was initialized within 7–9 m/s, and the velocity of OVs was initialized within 5–7 m/s. The maximal and minimal velocities of AVs were set as 10 m/s and 2 m/s, respectively. The initialized position of AVs was located at a distance of 45–65 m away from the intersection.

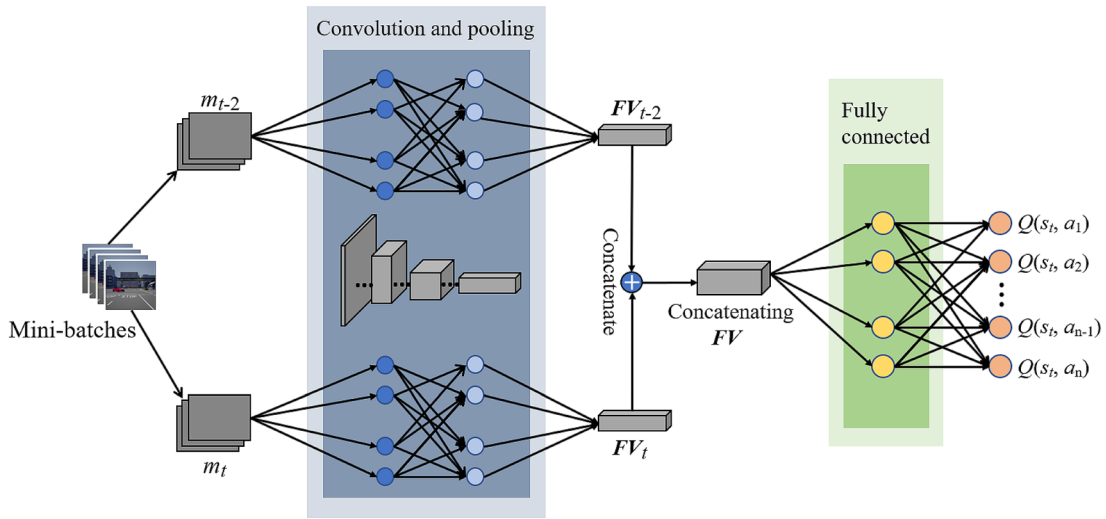

Fig. 6 The proposed DQN structure

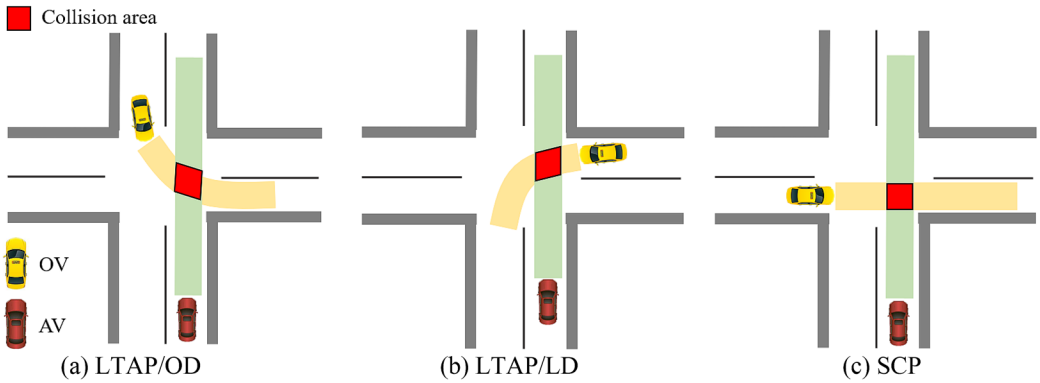

Fig. 7 Three tested scenarios at intersections

A trial would be ended if the AVs straightly crossed the intersection without a crash or collision with OVs during training. Every time the simulation started, the initialized positions and velocities of AVs and OVs would be randomly selected from the pre-determined ranges to improve the generalization capability of the proposed method. For each tested scenario, 20 trials were conducted to examine the effectiveness of the proposed method.

To further demonstrate the advantages of the proposed method, a Monte-Carlo-sampling-based method (MCS method) [52] and a Bayesian-network-based method (BN method) [12] were adopted for comparison. The MCS method used Monte Carlo sampling to predict the future motion of vehicles and estimate the risk for decision inference. The BN method used Bayesian theories to predict the probability of collision and make driving decisions accordingly. Both methods made effective decisions in collision avoidance at intersections.

6 Results and Discussion

The generated trajectories of AVs and OVs when using different decision-making methods are illustrated in Fig. 8. The results show that all three methods could effectively generate collision-free trajectories in LTAP/OD, LTAP/LD, and SCP scenarios. However, the gradients of the trajectories generated by MCS increased when the AVs were still far away from the collision area and the OVs, which indicates that the AVs took an over-conservative strategy advance decelerations, to avoid potential collisions. Similar performances were observed when using the BN method. As for the performance of the proposed DRL-based method, the gradients of the trajectories were mostly constant at the beginning, indicating that AVs and OVs were driving at stable velocities. When the trajectories were close to the collision area, which means that AVs and OVs were getting closer with potential collisions, the gradients of the trajectories increased to avoid entering the collision area, which means that AVs were decelerating to ensure driving safety. No trajectories ever entered the collision area, indicating that all the potential collisions were successfully avoided. Meanwhile, the short distance between trajectories and the collision area was negligible, showing that the proposed method avoided over-conservative decisions to achieve high driving efficiency.

To quantitatively demonstrate the higher travel efficiency of the decisions from the proposed method, the mean velocities and standard deviations of the AVs while driving through intersections in the three tested scenarios using different methods are shown in Table 1. The mean velocities of the proposed DRL-based method were higher than those of the MCS and BN methods in all three scenarios, indicating that the proposed DRL-based method could achieve higher travel efficiency while driving through intersections with potential collisions. Hence, the illustrated results showed that the proposed DRL-based method made AVs travel more safely and efficiently through intersections than the MCS and BN methods did. The standard deviations of the proposed method in LTAP/LD and SCP were higher than the numbers of the MCS and BN methods, indicating that the velocity of AVs changed more dramatically when using the method proposed in this study. The larger standard deviations meant the driving comfort decreased, which is one of the limitations of this study.

To better quantify the performance of the three methods, the kinematic information of AVs during the whole driving process through intersections is shown in Fig. 9. The blue, orange and green lines represent the velocity, acceleration of AVs, and the relative distance between AVs and OVs, respectively. In the first 3 s, the velocity of AVs was relatively stable when using the proposed DQN-based method because the relative distance between AVs and OVs had not obviously affected driving safety yet. When the relative distance continuously decreased, the velocity and acceleration of AVs increased (but not speeding) in order to pass the intersection quickly before OVs to improve travel efficiency.

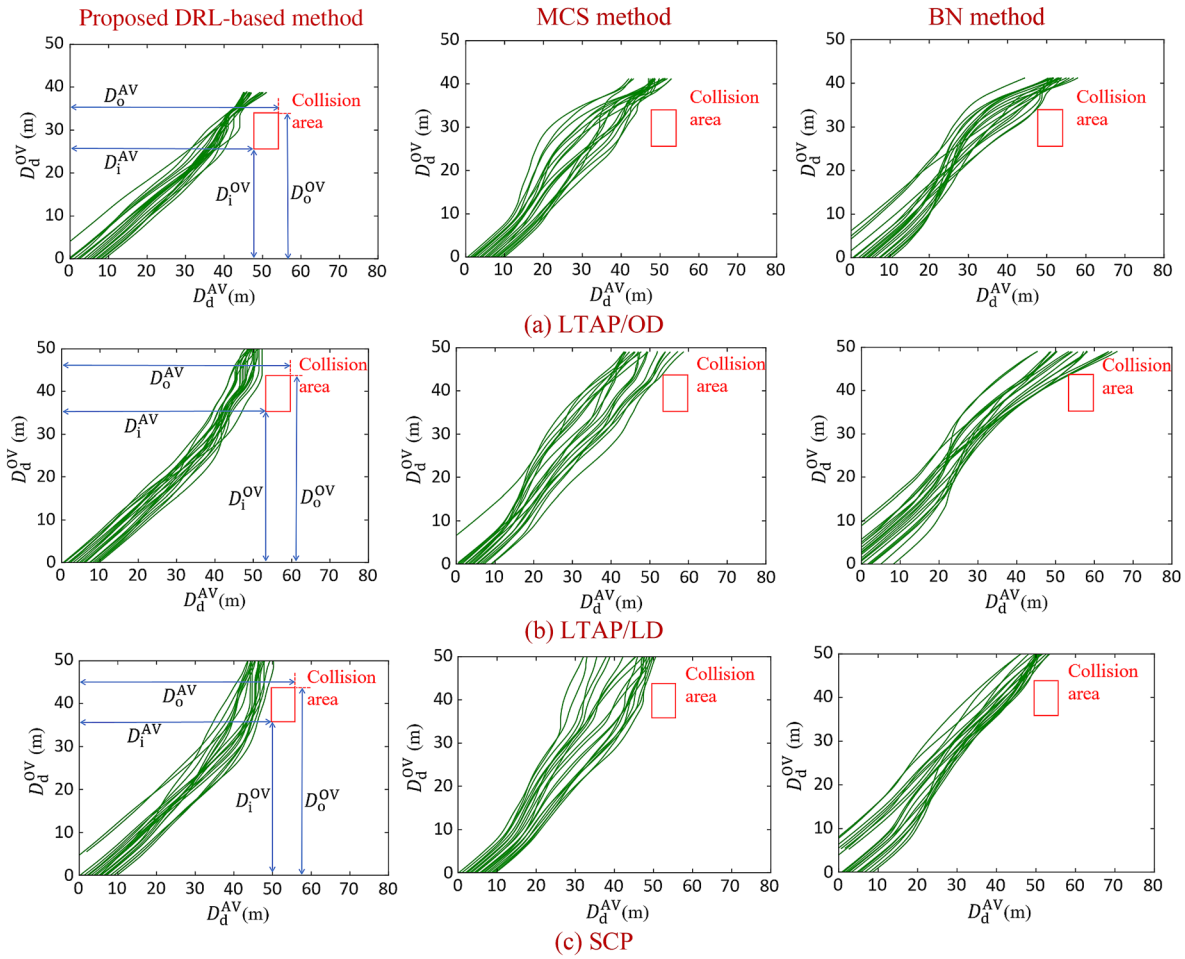

Fig. 8 Trajectories in all the tested scenarios with different methods. $D_{\mathrm{d}}^{\mathrm{AV}}$ is the driving distance of AVs from the initialized position, and $D_{\mathrm{d}}^{\mathrm{OV}}$ is the driving distance of OVs from the initialized position. The driving distances of AVs from the start point to entering and traveling out of the collision area are denoted as $D_{\mathrm{i}}^{\mathrm{AV}}$ and $D_{\mathrm{o}}^{\mathrm{AV}}$, respectively. The driving distances of OVs from the start point to entering and traveling out of the collision area are denoted as $D_{\mathrm{i}}^{\mathrm{OV}}$ and $D_{\mathrm{o}}^{\mathrm{OV}}$, respectively

Table 1 Velocity of AVs when passing through intersection using different methods

Method	LTAP/OD		LTAP/LD		SCP	
	Mean (m/s)	Standard deviation (m/s)	Mean (m/s)	Standard deviation (m/s)	Mean (m/s)	Standard deviation (m/s)
This study	6.54	1.43	7.03	1.62	6.36	1.77
MCS method	5.52	1.57	6.24	1.55	4.67	1.72
BN method	5.71	1.68	5.48	0.93	5.23	1.54

However, when the RD decreased to a more dangerous level, the AVs gave up the acceleration decision and decelerated to avoid collisions. When the OVs travelled out of the collision area, the AVs accelerated to cross the intersection. Similar performance can be found in all the three tested scenarios when using the proposed method. However, when using the MCS or BN method, the AVs started to decelerate for collision avoidance when the relative distance between AVs and OVs was still too long to challenge driving safety. This led to over-conservative action strategies which may strongly weaken drivers' acceptances to autonomous vehicles. In addition, the time it took for the relative distance between

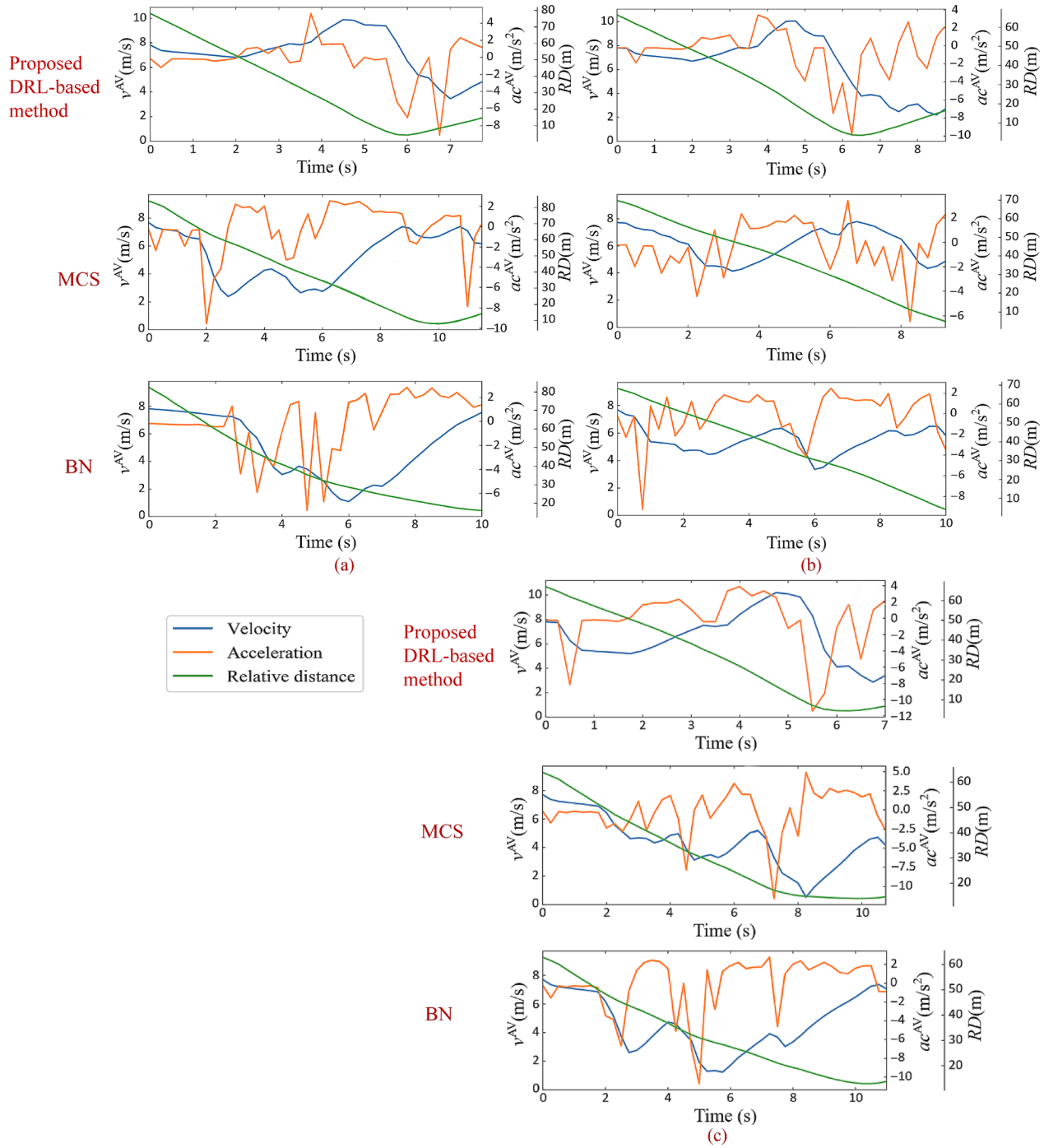

Fig. 9 The kinematic information of AVs during the process of driving through intersections when using **a** LTAP/OD, **b** LTAP/LD and **c** SCP. v^{AV} is the velocity of AVs. ac^{AV} is the acceleration of AVs. *RD* is the relative distance between AVs and OVs

AVs and OVs to reach the minimum, where AVs were about to travel into the center of intersection, is obviously shorter when adopting the proposed method compared with the MCS and BN methods. Hence, the kinematic results further proved that the proposed method performed better than the compared methods in both driving safety and efficiency.

7 Conclusions

This paper proposed a novel DRL-based decision-making framework for autonomous driving at intersections. Based on the end-to-end decision-making model trained by DQN, the proposed framework used both relative distance and

velocity between AVs and OVs to make safe and efficient driving decisions. The improved performance and reliability of the proposed method were verified in the three most accident-prone scenarios at intersections (LTAP/OD, LTAP/LD, and SCP). The results show that the trained AVs drive safely and efficiently when traveling through intersections. The proposed method could help design the decision-making module of AVs to enhance traffic safety and travel efficiency. As the proposed DQN-based decision-making framework is with discrete actions, continuous action strategies will be further developed for driving comfort improvement in future studies. Besides, lateral maneuvers need to be considered in future work. Moreover, multi-agent RL solutions with individual decision-making models can be extended for analysis based on this paper.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant No. 51805332), the Young Elite Scientists Sponsorship Program funded by the China Society of Automotive Engineers, the Natural Science Foundation of Guangdong Province (Grant No. 2018A030310532), and the Shenzhen Fundamental Research Fund (Grant No. JCYJ20190808142613246).

Compliance with Ethical Standards

Conflict of interest On behalf of all authors, the corresponding authors state that there is no conflict of interest.

References

1. Tay, R.: A random parameters probit model of urban and rural intersection crashes. Accid. Anal. Prev. **84**, 38–40 (2015)
2. Li, G.F., Wang, Y., Zhu, F.P., et al.: Drivers' visual scanning behavior at signalized and unsignalized intersections: a naturalistic driving study in China. J. Saf. Res. **71**, 219–229 (2019)
3. Werneke, J., Vollrath, M.: How do environmental characteristics at intersections change in their relevance for drivers before entering an intersection: analysis of drivers' gaze and driving behavior in a driving simulator study. Cogn. Technol. Work **16**(2), 157–169 (2014)
4. Lemonnier, S., Brémond, R., Baccino, T.: Gaze behavior when approaching an intersection: dwell time distribution and comparison with a quantitative prediction. Transp. Res. Part F Traffic Psychol. Behav. **35**, 60–74 (2015)
5. NHTSA: Traffic Safety Facts 2017 Data (DOT HS 812 806, updated September 2019). National Highway Traffic Safety Administration, Washington, DC (2019). https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812806. Accessed 14 Feb 2020
6. Zhang, G.G., Yau, K.K.W., Chen, G.H.: Risk factors associated with traffic violations and accident severity in China. Accid. Anal. Prev. **59**, 18–25 (2013)
7. Ronald, J.: V2V/V2I Communications for Improved Road Safety and Efficiency. SAE, San Diego (2012)
8. Li, G.F., Yang, Y.F., Qu, X.D.: Deep learning approaches on pedestrian detection in hazy weather. IEEE Trans. Ind. Electron. (2019). https://doi.org/10.1109/TIE.2019.2945295
9. Dooley, D., McGinley, B., Hughes, C., et al.: A blind-zone detection method using a rear-mounted fisheye camera with combination of vehicle detection methods. IEEE Trans. Intell. Transp. Syst. **17**(1), 264–278 (2015)
10. Li, S.E.B., Li, G.F., Yu, J.Y., et al.: Kalman filter-based tracking of moving objects using linear ultrasonic sensor array for road vehicles. Mech. Syst. Signal Proc. **98**, 173–189 (2018)
11. Li, G.F., Li, S.E.B., Zou, R.B., et al.: Detection of road traffic participants using cost-effective arrayed ultrasonic sensors in low-speed traffic situations. Mech. Syst. Signal Proc. **132**, 535–545 (2019)
12. Noh, S.: Decision-making framework for autonomous driving at road intersections: safeguarding against collision, overly conservative behavior, and violation vehicles. IEEE Trans. Ind. Electron. **66**(4), 3275–3286 (2018)
13. Najm, W., Smith, J.D., Smith, D.L.: Analysis of crossing path crashes. John A. Volpe National Transportation Systems Center (US). No. DOT-VNTSC-NHTSA-01-03 (2001)
14. Brannstrom, M., Coelingh, E., Sjoberg, J.: Model-based threat assessment for avoiding arbitrary vehicle collisions. IEEE Trans. Intell. Transp. Syst. **11**(3), 658–669 (2010)
15. Li, G.F., Li, S.E., Cheng, B., et al.: Estimation of driving style in naturalistic highway traffic using maneuver transition probabilities. Transp. Res. Part C Emerg. Technol. **74**, 113–125 (2017)
16. Polychronopoulos, A., Tsogas, M., Amditis, A., et al.: Sensor fusion for predicting vehicles' path for collision avoidance systems. IEEE Trans. Intell. Transp. Syst. **8**(3), 549–562 (2007)
17. Campos, G.R., Runarsson, A.H., Granum, F., et al.: Collision avoidance at intersections: a probabilistic threat-assessment and decision-making system for safety interventions. Paper presented at the 17th International IEEE Conference on Intelligent Transportation Systems, Qingdao, China, 8–11 October 2014
18. Jansson, J.: Collision avoidance theory with application to automotive collision mitigation. Dissertation, Linköping University (2005)
19. Noh, S., Han, W.Y.: Collision avoidance in on-road environment for autonomous driving. Paper presented at the 14th International Conference on Control, Automation and Systems, Seoul, South Korea, 22–25 October 2014
20. Naranjo, J.E., Gonzalez, C., Garcia, R., et al.: Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver. IEEE Trans. Intell. Transp. Syst. **9**(3), 438–450 (2008)
21. Hubmann, C., Schulz, J., Becker, M., et al.: Automated driving in uncertain environments: planning with interaction and uncertain maneuver prediction. IEEE Trans. Intell. Veh. **3**(1), 5–17 (2018)
22. Schubert, R.: Evaluating the utility of driving: toward automated decision making under uncertainty. IEEE Trans. Intell. Transp. Syst. **9**(3), 354–364 (2011)
23. Jansson, J., Gustafsson, F.: A framework and automotive application of collision avoidance decision making. Automatica **44**(9), 2347–2351 (2008)
24. Armand, A., Filliat, D., Ibanez-Guzman, J.: Modelling stop intersection approaches using Gaussian processes. Paper presented at the 16th International IEEE Conference on Intelligent Transportation Systems, The Hague, Netherlands, 6–9 October 2013
25. Huang, R., Liang, H., Zhao, P., et al.: Intent-estimation-and motion-model-based collision avoidance method for autonomous vehicles in urban environments. Appl. Sci. **7**(5), 457 (2017)
26. Notomista, G., Botsch, M.: A machine learning approach for the segmentation of driving maneuvers and its application in autonomous parking. J. Artif. Intell. Soft Comput. Res. **7**(4), 243–255 (2017)
27. Hussein, A., Gaber, M.M., Elyan, E., et al.: Imitation learning: a survey of learning methods. ACM Comput. Surv. **50**(2), 1–35 (2017)

28. Bojarski, M., Yeres, P., Choromanska, A., et al.: Explaining how a deep neural network trained with end-to-end learning steers a car (2017). arXiv preprint arXiv:1704.07911
29. Mo, C.M., Li, Y.N., Zheng, L.: Simulation and analysis on overtaking safety assistance system based on vehicle-to-vehicle communication. Automot. Innov. **1**(2), 158–166 (2018)
30. Hafner, M.R., Cunningham, D., Caminiti, L., et al.: Cooperative collision avoidance at intersections: algorithms and experiments. IEEE Trans. Intell. Transp. Syst. **14**(3), 1162–1175 (2013)
31. Rios-Torres, J., Malikopoulos, A.A.: A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. IEEE Intell. Transp. Syst. Mag. **18**(5), 1066–1077 (2017)
32. Lee, J., Park, B.: Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Trans. Intell. Transp. Syst. **13**(1), 81–90 (2012)
33. Luo, Y.G., Yang, G., Xu, M.C., et al.: Cooperative lane-change maneuver for multiple automated vehicles on a highway. Automot. Innov. **2**(3), 157–168 (2019)
34. Zong, X.P., Xu, G.Y., Yu, G.Z., et al.: Obstacle avoidance for self-driving vehicle with reinforcement learning. SAE Int. J. Passeng. Cars Electron. Electr. Syst. **11**(1), 30–39 (2017)
35. Kendall, A., Hawke, J., Janz, D., et al.: Learning to drive in a day. Paper presented at the 2019 International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May (2019)
36. Zhu, M.X., Wang, X.S., Wang, Y.H.: Human-like autonomous car-following model with deep reinforcement learning. Transp. Res. Part C Emerg. Technol. **97**, 348–368 (2018)
37. Ye, Y.J., Zhang, X.H., Sun, J.: Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment. Transp. Res. Part C Emerg. Technol. **107**, 155–170 (2019)
38. Zhou, M.F., Yu, Y., Qu, X.B.: Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: a reinforcement learning approach. IEEE Trans. Intell. Transp. Syst. **21**(1), 433–443 (2020)
39. Qi, X.W., Luo, Y.D., Wu, G.Y., et al.: Deep reinforcement learning enabled self-learning control for energy efficient driving. Transp. Res. Part C Emerg. Technol. **99**, 67–81 (2019)
40. Bellman, R.: A Markovian decision process. J. Math. Mech. **6**(5), 679–684 (1957)
41. Arulkumaran, K., Deisenroth, M.P., Brundage, M., et al.: Deep reinforcement learning: a brief survey. IEEE Signal Process. Mag. **34**(6), 26–38 (2017)
42. Dolcetta, I.C., Ishii, H.: Approximate solutions of the Bellman equation of deterministic control theory. Appl. Math. Optim. **11**(1), 161–181 (1984)
43. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
44. Mnih, V., Kavukcuoglu, K., Silver, D., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
45. Tokic, M., Palm, G.: Value-difference based exploration: adaptive control between epsilon-greedy and softmax. Paper presented at the 34th Annual German conference on Advances in artificial intelligence, Heidelberg, Berlin, 4–7 October (2011)
46. Wisch, M., Hellmann, A., Lerner, M., et al.: Car-to-car accidents at intersections in Europe and identification of use cases for the test and assessment of respective active vehicle safety systems. Paper presented at 26th International Technical Conference on the Enhanced Safety of Vehicles, Eindhoven, Netherlands, 10–13 June (2019)
47. Kusano, K.D., Gabler, H.C.: Target population for intersection advanced driver assistance systems in the U.S. SAE Int. J. Transp. Saf. **3**(1), 1–16 (2015)
48. Arikere, A., Yang, D.R., Klomp, M.: Optimal motion control for collision avoidance at left turn across path/opposite direction intersection scenarios using electric propulsion. Veh. Syst. Dyn. **57**(5), 637–664 (2019)
49. Sander, U., Lubbe, N., Pietzsch, S.: Intersection AEB implementation strategies for left turn across path crashes. Traffic Inj. Prev. **20**(sup1), S119–S125 (2019)
50. Ulrich, S., Nils, L.: The potential of clustering methods to define intersection test scenarios: assessing real-life performance of AEB. Accid. Anal. Prev. **113**, 1–11 (2018)
51. Dosovitskiy, A., Ros, G., Codevilla, F., et al.: CARLA: an open urban driving simulator (2017). arXiv preprint arXiv:1711.03938
52. Eidehall, A., Petersson, L.: Statistical threat assessment for general road scenes using Monte Carlo sampling. IEEE Trans. Intell. Transp. Syst. **9**(1), 137–147 (2008)