Towards Robust Decision-Making for Autonomous Driving on Highway

Kai Yang, Xiaolin Tang, *Senior Member, IEEE*, Sen Qiu, *Member, IEEE*, Shufeng Jin, Zichun Wei, and Hong Wang, *Senior Member, IEEE*

***Abstract*—Reinforcement learning (RL) methods are commonly regarded as effective solutions for designing intelligent driving policies. Nonetheless, even if the RL policy is converged after training, it is notoriously difficult to ensure safety. In particular, RL policy is susceptible to insecurity in the presence of long-tail or unseen traffic scenarios, *i.e.*, out-of-distribution test data. Therefore, the design of the RL-based decision-making method must account for this shift in distribution. This paper proposes a robust decision-making framework for autonomous driving on the highway to improve driving safety. First, a Deep Deterministic Policy Gradient (DDPG)-based RL policy that directly maps observations to actions is constructed. Subsequently, the model uncertainty of the DDPG policy is evaluated at runtime to quantify the policy's reliability and identify unseen scenarios. In addition, a complementary principle-based policy is developed using the intelligent driver model (IDM) and the model for minimizing overall braking induced by lane changes (MOBIL). It will take over the DDPG policy when encountering unseen scenarios to guarantee a lower-bound performance of the decision-making system. Finally, the proposed method is implemented on an embedded system, *i.e.*, NVIDIA Jetson AGX Xavier, and out-of-training distribution challenging cases are considered in the experiment, *i.e.*, observation with sensor noise, traffic density increasing significantly, objects falling from the front vehicle, and road construction causing temporal changes in road structure. Results indicate that the proposed framework outperforms state-of-the-art benchmarks. Additionally, the code is provided.**

***Index Terms*—Autonomous vehicles, decision-making, reinforcement learning policy, rule-based policy.**

Manuscript received 30 August 2022; revised 29 January 2023; accepted 14 April 2023. Date of publication 24 April 2023; date of current version 19 September 2023. This work was supported in part by the National Natural Science Foundation of China Project under Grants 52222215, 52072051, U1964203, and 52072215, and in part by the National Key R&D Program of China under Grant 2020YFB1600303. The review of this article was coordinated by Dr. Dinh-Thuan Do. *(Corresponding authors: Xiaolin Tang; Hong Wang.)*

Kai Yang, Xiaolin Tang, and Shufeng Jin are with the College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing 400044, China (e-mail: kaiyang0401@gmail.com; tangxl0923@cqu.edu.cn; jinshufeng1997@163.com).

Sen Qiu is with the Dalian University of Technology, Dalian 116024, China (e-mail: qiu@dlut.edu.cn).

Zichun Wei is with the Viterbi School of Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: zichunwe@usc.edu).

Hong Wang is with the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (e-mail: hong_wang@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TVT.2023.3268500

I. Introduction

A. Motivation

AUTONOMOUS driving is regarded as one of the revolutionary technologies enhancing driving safety, mobility and energy efficiency in various traffic scenarios [1], [2], [3], [4]. Waymo, Baidu-Apollo, and others have demonstrated in recent years that their autonomous vehicle (AV) products perform well in normal traffic situations. However, according to their safety report, hundreds of takeover incidents were still documented when operating in unknown or long-tail scenarios [5]. Meanwhile, a recent survey reveals that the leading concern regarding the acceptability of AVs is safety, not economic consequences or privacy concerns [6], [7]. Therefore, the gap between this forthcoming autonomous future and existing state-of-the-art technologies must be bridged by investigating additional safety guarantees [8], [9], [10].

Currently, decomposing the task into multiple subtasks in a hierarchical manner, which reduces computational complexity and provides good decision-making transparency, is a prevalent solution for autonomous driving companies. Nonetheless, it requires cumbersome hand-crafted rules and may fail when processing difficult and highly interactive cases. Data-driven methods such as reinforcement learning (RL) methods that can learn policy from driving logs, can be used to design more intelligent policies. Unfortunately, RL-based decision-making is an endeavour wrought with a high degree of uncertainty, a large portion of which, arguably, is epistemic uncertainty (model uncertainty), arising from unavoidable learning errors when training models from finite driving logs. Specifically, they are likely to be unsafe in the face of long-tail or unseen traffic scenarios, *i.e.*, far-from-distribution test scenarios. Consequently, the reliability of RL policy must be monitored in real-time, and the far-from-distribution scenarios must be identified, allowing AVs to avoid making rash, potentially risky decisions outside of the training distribution. In addition, when confronted with unanticipated scenarios, safety guarantees should be developed to enhance driving safety and take over RL policy when it is unreliable.

B. Related Work

1) Decision-Making Methods: Existing decision-making algorithms can be roughly categorized as rule-based [11], [12], [13] and and learning-based [14] techniques. Rule-based methods predefine multiple behavior rules based on expert knowledge, thereby providing interpretability and transparency in behavior-risk reasoning. To account for extremely complex driving conditions, however, the rules would be quite complex, which fundamentally limits the potential for further improvement. In some instances, the possible contradiction between rules may result in unresolved decision problems, which may also pose security risks. Recently, learning-based decision-making methods, such as deep RL and imitation learning, have been employed to solve the decision-making problem. These techniques are promising because they rely on interactions with the environment instead of hand-coded features, which could be generalized to complex and highly interactive conditions. For instance, an intelligent overtaking method for highway autonomous driving was developed based on Q-learning, which instructs the vehicle to drive in a proper lane and generate acceleration commands. Moreover, a decision-making framework based on the Deep Deterministic Policy Gradient (DDPG) was proposed to control the vehicle so that it can respond to emergencies effectively and safely [15]. In addition, a deep Q-network (DQN)-based method for determining the AV's lane selection and acceleration was proposed [16]. Furthermore, Tang et al. [17] proposed a decision-making controller with continuous action space for highway driving scenarios based on soft actor-critic (SAC). Results indicate that the proposed method can efficiently solve the decision-making problem. Nonetheless, although these RL-based methods are effective in highway scenarios, they were evaluated in training scenarios without taking out-of-distribution scenarios into account.

2) Safety Guarantees of RL-based Decision-Making: Currently, multiple works have been completed to ensure the performance of the RL policy, which can be roughly categorized into three categories: expert policy heuristic, dangerous action correction, and hybrid decision-making. The expert policy heuristic technique involves learning to improve performance by imitating an expert policy, such as a human-engineering policy. Behavior cloning [18], inverse reinforcement learning (IRL) [19], [20], and adding expert policy heuristic reward are common ways to utilize expert policy. The safety of RL policy can also be improved by correcting dangerous actions. For instance, merging mixed-traffic on-ramps was formulated as a decentralized multi-agent RL problem in [21] where an action masking scheme was used to improve learning efficiency by filtering out invalid or unsafe actions. In addition, in [22], unsafe actions were corrected by developing a safety-oriented reward function and imposing a penalty when the policy results in a harmful result for autonomous highway driving. However, these techniques still require substantial training data or advanced safeguards. The third way to guarantee the RL policy performance is to integrate RL policy with other conventional methods, *e.g.*, model prediction control (MPC), and principle-based methods like the intelligent driver model (IDM). These techniques can guarantee a minimum performance threshold for policies using conventional methods. In [23], RL and MPC were combined to enhance learning efficiency, achieving a good balance between passenger comfort, fuel economy, and accident rate. Similarly, Cao et al. [24] combined RL policy and principle-based methods, where RL only intervenes when the rule-based method appears to have difficulty handling and when RL policy confidence is high. However, the switching mechanism relies on the accurate motion prediction of road users, and their hybrid methods do not account for the model uncertainty of RL policy.

3) Measures of Model Uncertainty and Applications in RL: As previously stated, learning-based decision-making algorithms provide black box solutions, *i.e.*, they provide a result whenever input is provided, regardless of whether the result is reliable or not. In other words, the inherent uncertainty of learning-based agents poses a threat to safety-critical tasks, such as autonomous driving. Therefore, it would be desirable for the learning-based agent to provide an estimate of its confidence level or, equivalently, the model uncertainty associated with its decisions. Specifically, model uncertainty reflects how well a model fits all possible environmental observations and what the learning model does not know. Several research fields have adapted neural networks to express model uncertainty [25], [26], [27]. For example, a fully Bayesian recurrent neural network architecture based on the Probabilistic Backpropagation (PBP) method was built to estimate the model uncertainty of RL policy [28]. Besides, the bootstrapping method was investigated to generate approximate uncertainty measures to guide exploration [26]. In addition, dropout has been shown to approximate Bayesian inference for Gaussian processes [29]. Specifically, authors in [30] incorporated Monte-Carlo Dropout and bootstrapping techniques to estimate model uncertainty, which was embedded within an RL framework to produce uncertainty-aware navigation around pedestrians. Furthermore, the confidence in the RL policy was determined by applying the Lindeberg-Levy Theorem to the training data distribution [24]. Moreover, by training an ensemble of networks on partially overlapping dataset samples, models agree in common data areas and disagree in rare data areas, with a substantial variation in sample size. A comprehensive review of uncertainty for deep reinforcement learning can be found in [31], [32]. Recent research has employed an ensemble of neural networks with randomized prior functions (RPF) to extend ensemble methods to deep RL [33]. Utilizing Bayesian RL with RPF to estimate the model uncertainty of discrete action space RL enhanced decision-making safety [34], [35], [36]. The results indicate that the proposed ensemble RPF method could be aware of high uncertainty when encountering scenarios outside the training distribution. To the author's best knowledge, this is the study that is closest to our work. However, it concentrates on discrete action space-based RL, ignoring continuous action space. Moreover, this research does not consider how to improve safety when RL algorithms are unreliable; instead, the maximum deceleration maneuver is utilized. In contrast to the related work, this paper investigates a continuous action space RL, *i.e.*, DDPG, that can estimate model uncertainty and is integrated with a principle-based (PB) method to further improve safety.

C. Contribution

In brief, the main contributions and the technical advancements of this paper are summarized as follows: 1) To improve decision-making safety on the highway, a robust decision-making framework is proposed, which combines a continuous action space RL policy based on DDPG with a PB policy. 2) The model uncertainty of DDPG policy is estimated that can be used to quantify the reliability of DDPG policy and identify unseen scenarios. 3) The mechanism for switching between DDPG and PB policies is investigated and designed. The effectiveness of proposed methods is validated using challenging cases and the real-time performance is also demonstrated via embedded equipment.

D. Paper Organization

The rest of this paper is organized as follows. Section II provides the preliminaries of this work. The details of the robust decision-making framework are shown in Section III. In Section IV, the specific implementation of the experiment is given. The results and discussion are provided in Section V, followed by the conclusion in Section VI.

II. Background

A. Fundamentals of Reinforcement Learning

The decision-making problem of AVs is commonly formulated as a Markov decision process (MDP) or a partially observable Markov decision process (POMDP). A finite horizon MDP is denoted by a tuple (S, A, P, r, γ), where S is the state space, $s \in S$, A denotes the action space, $a \in A$, $P = P(s'|s, a)$ is the system dynamics probability of reaching a state s' from s when taking action a, $P : S \times A \times S \to \mathbb{R}$, r is the reward function $r : S \times A \to \mathbb{R}$, $\gamma \in (0, 1]$ is the discount factor. A general policy π maps each state to a distribution over action, *i.e.*, $\pi(s|a)$ which means the probability of acting a at state s using policy π. The agent aims at acquiring an optimal policy $\pi^* \in \prod$ by maximizing the following long-term discounted cumulative reward:

$$\pi^* = \arg\max_{\pi\in\prod} \mathbb{E}\pi\left[\sum_{t=0}^{H}\gamma^t r(s_t, a_t)\right] \tag{1}$$

where $\prod$ denotes the set containing all candidate policies π, t represents the time step, H is the finite planning horizon, $\mathbb{E}[\cdot]$ is the expectation.

Estimating the state or state-action value function and recovering a policy is a common strategy for optimizing the target. Following are definitions for the state value function and state-action value function.

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s, a)] \tag{2}$$

$$Q^\pi(s, a) = r(s, a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V^\pi(s')] \tag{3}$$

The optimal policy π^* can be obtained by maximizing Q^π, *i.e.*,

$$Q^{\pi_*}(s) = \arg\max_{\pi\in\prod} Q^\pi(s, a) \tag{4}$$

In this paper, the DDPG algorithm with continuous action space is adopted as the RL policy generator, *i.e.*, π_{rl}, which controls both longitudinal and lateral motions of the AV directly. Note that other RL methods with continuous action space may also work in our proposed framework but are not explored. In detail, DDPG is an off-policy actor-critic algorithm, which concurrently learns four networks, two critic-networks (online: $Q(s, a; \theta^Q)$ and target: $Q'(s, a; \theta^{Q'})$) and two actor-networks (online: $\mu(s; \theta^\mu)$ and target: $\mu'(s; \theta^{\mu'})$). The update process of the policy network is defined as follows:

$$\nabla\mathcal{J}(\theta^\mu) = \mathbb{E}_{s\sim\mathcal{D}}[\nabla_a Q(s, a; \theta^Q)|_{a=\mu(s)}\nabla_{\theta^\mu}\mu(s; \theta^\mu)] \tag{5}$$

$$\begin{aligned}\nabla\mathcal{L}(\theta^Q) = \mathbb{E}_{s,a,r,s'\sim\mathcal{D}}[(r + \gamma Q'(s', \mu'(s'; \theta^{\mu'}); \theta^{Q'})\\ - Q(s, a; \theta^Q))\nabla_{\theta^Q} Q(s, a; \theta^Q)]\end{aligned} \tag{6}$$

where θ^μ, $\theta^{u'}$ are the network parameters of online and target actor-networks, respectively, θ^Q, $\theta^{Q'}$ are the network parameters of online and target critic-networks, respectively, $\mathcal{D}$ means the replay buffer.

B. Principle-Based Driving Policy

As mentioned previously, it is difficult to ensure the safety of RL-based decision-making methods and they may pose a risk when confronted with far-from-distribution scenarios. Thus, a typical principle-based decision-making approach, *i.e.*, π_{pb} is developed, which works as a complementary driving policy to guarantee a lower-bound performance of the decision-making system. Specifically, the IDM is used to make decisions regarding longitudinal motion. The IDM model is illustrated below [12].

$$\dot{u} = \alpha\left[1 - \left(\frac{u}{u_r}\right)^\eta - \left(\frac{d_0 + T_0 u + \frac{u\Delta u}{2\sqrt{\alpha\beta}}}{\Delta d}\right)\right] \tag{7}$$

where u and $\dot{u}$ are the velocity and acceleration of the ego vehicle, respectively. u_r represents the desired free flow velocity, η denotes the exponent for velocity, d_0 is the standstill distance, and T_0 represents the safe time gap. Δu, Δd represent the velocity difference and actual gap between the ego vehicle and its front vehicle. α, β are predefined parameters of the IDM model.

In addition, the minimizing overall braking induced by lane change (MOBIL) model is applied to the decision-making for lateral maneuvers [37]. When the subsequent condition is met, the lane change maneuver is generated.

$$\tilde{\dot{u}}_e + p\left(\tilde{\dot{u}}_n - \dot{u}_n + \tilde{\dot{u}}_o - \dot{u}_o\right) > a_{th} \tag{8}$$

where the subscript e, n, and o represent ego vehicle, new follower, and old follower, respectively. $\dot{u}$ denotes the current acceleration, $\dot{u}^{\sim}$ represents the acceleration if a lane change command is made. p and a_{th} are the courtesy factor of the MOBIL model and acceleration threshold, respectively.

With the target lane provided by MOBIL, a proportional-derivative controller is used to calculate the steering angle δ:

$$v_{ex}^{lat} = -K_p^{lat} d_{tar}$$

$$\theta_{ex} = \arcsin\left(\frac{v_{ex}^{lat}}{v} + \theta_{tar}\right)$$

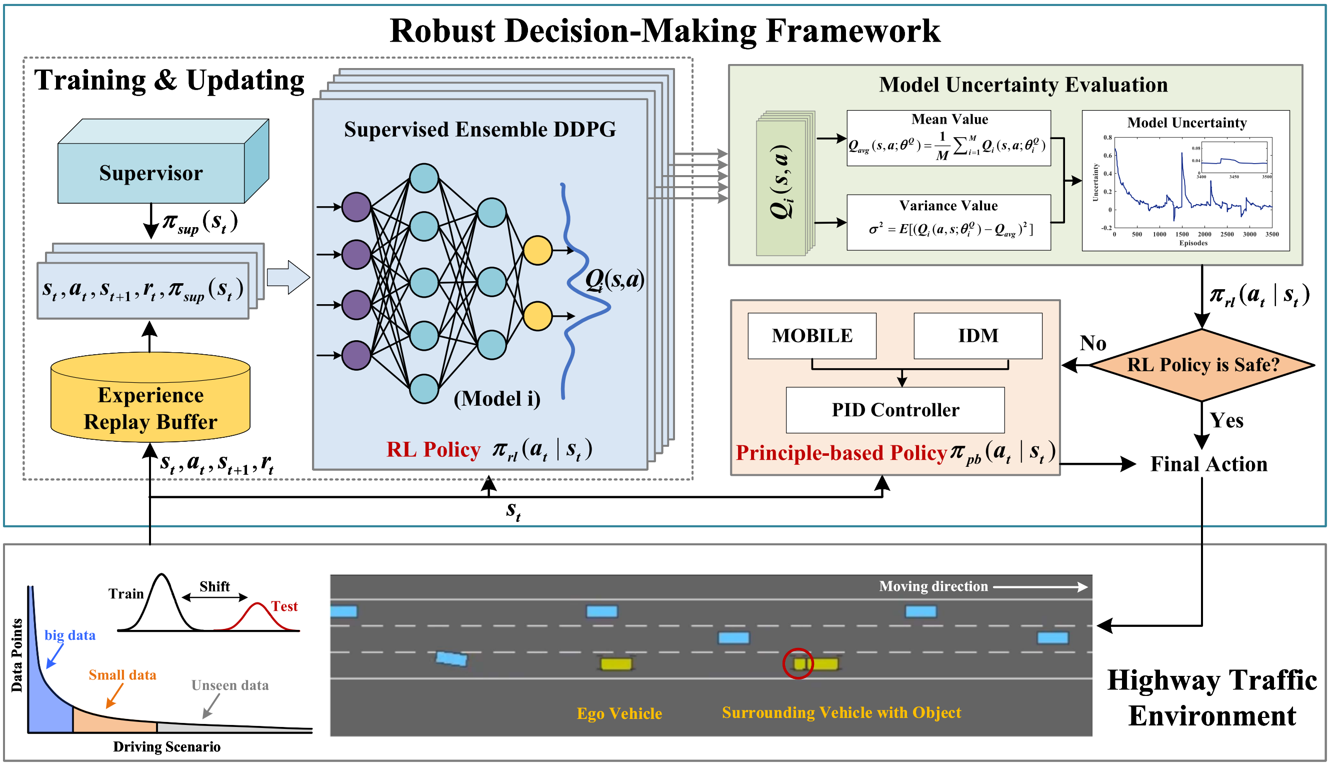


Fig. 1. Conceptual framework of the proposed method.

$$\dot{\theta} = K_p^{\theta}(\theta_{ex} - \theta)$$
$$\delta = \arcsin\left(\frac{l_r \dot{\theta}}{2v}\right) \tag{9}$$

where v_{ex}^{lat} represents the desired lateral speed, K_p^{lat} and K_p^{θ} are the position and heading control gains, d_{tar} denotes the lateral distance between the vehicle and the center-line of the target lane, θ_{ex} represents the desired heading, θ_{tar} is the heading of the center line of the target lane, θ denotes the heading of the vehicle, and l_r is the distance between the mass center of the vehicle and the rear axle.

III. Robust Decision-Making Framework

A. Framework

Fig. 1 depicts the conceptual framework of the proposed method, which is comprised of four components: the supervised ensemble DDPG policy (π_{rl}), the IDM+MOBIL policy (π_{pb}), the model uncertainty evaluation, and the switching mechanism. First, a decision-making controller with continuous action space is constructed based on the DDPG algorithm. This controller directly maps observations of the traffic environment to low-level actions. The part that evaluates model uncertainty is in charge of supervised ensemble DDPG policy model uncertainty quantification. When the model uncertainty is higher than the safety threshold, we consider the current scenario to be out-of-training distribution, *i.e.*, the RL policy is unsafe. Consequently, the PB driving controller will be implemented as a supplementary driving policy to enhance safety. It intervenes and assumes control of the AV when the DDPG policy confronts unseen scenarios. Notice that one key is to determine the safety threshold of model uncertainty that will be investigated in Section V. Once the threshold is determined, a reasonable and feasible mechanism for switching between the two strategies can be designed to take advantage of both strategies.

Definition 1: Model uncertainty results from the distribution mismatch between the data the RL model sees during testing and that used to train the RL model. In other words, it arises from unavoidable learning errors when training RL models from finite driving logs [31].

B. Supervised Ensemble Deep Deterministic Policy Gradient

The vanilla actor-critic RL algorithms, such as DDPG, do not provide any risk information regarding decisions, which may potentially threaten the AV's safety. Therefore, it is expected that the DDPG driving policy will output its action with risk information. And the ensemble technique is used to estimate the risk associated with the DDPG policy based on [38]. The overall network structure diagram is shown in Fig. 1. Specifically, assume that the number of critic-networks used in DDPG policy is M, and estimate the Q-value function using the average value $Q_{\text{avg}}(s,a;\theta^Q)$ of M online critic-networks. Analogously, $Q'_{\text{avg}}(s,a;\theta^{Q'})$ represents the average value of the M target critic-networks, which is calculated as follows:

$$Q_{\text{avg}}(s,a;\theta^Q) = \frac{1}{M}\sum_{i=1}^{M} Q_i(s,a;\theta_i^Q)$$

$$Q'_{\text{avg}}(s', \mu'(s'; \theta^{\mu'}); \theta^{Q'}) = \frac{1}{M} \sum_{i=1}^{M} Q'_i(s', \mu'(s'; \theta^{\mu'}); \theta_i^{Q'}) \tag{10}$$

where $Q_i(s, a; \theta_i^Q)$ and $Q'_i(s', \mu'(s'; \theta^{\mu'}); \theta_i^{Q'})$ are the Q-values of $i-th$ online critic and target critic-networks, s' represents the state at the next moment of s, θ_i^Q, and $\theta_i^{Q'}$ are corresponding network parameters, respectively, $\mu'(s'; \theta^{\mu'})$ represents the action of the target actor-network. Then, the average TD deviation and cost function are:

$$\delta = r(s, a) + \gamma Q'_{\text{avg}}(s', \mu'(s'; \theta^{\mu'}); \theta^{Q'}) - Q_{\text{avg}}(s, a; \theta^Q) \tag{11}$$

$$\mathcal{L}_{\text{avg}}(\theta^Q) = \delta^2 \tag{12}$$

where $r(s, a) + \gamma Q'_{\text{avg}}(s', \mu'(s'; \theta^{\mu'}); \theta^{Q'})$ represents the average target Q-values. When updating the parameters of M evaluation networks, the cost function can be calculated by balancing the average TD deviation of all evaluation networks and their corresponding TD deviations. In this way, fluctuations in network training time can be reduced. If the $i-th$ online critic-network is selected to be updated, the gradient backpropagation will only activate for $i-th$ online critic-network. In addition, the selected critic-network must consider the following loss function:

$$\mathcal{L}_{td_i}(\theta_i^Q) = (r(s, a) + \gamma Q'_i(s', \mu'(s'; \theta^{\mu'}); \theta^{Q'_i}) - Q_i(s, a; \theta_i^Q))^2 \tag{13}$$

When the disparities between the networks become too great, additional unstable factors will be introduced into the training of the actor-network and other critic-networks. Consequently, the cost function must incorporate an extra penalty term. The value is the mean square error between the output value of each evaluation network and the average output value of all evaluation networks, ensuring that the output results of the networks are comparable. The loss function of the chosen critic-network is defined as follows:

$$\mathcal{L}(\theta_i^Q) = \zeta \mathcal{L}_{\text{avg}}(\theta^Q) + \eta \mathcal{L}_{td_i}(\theta_i^Q) + \omega (Q_i - Q_{\text{avg}})^2 \tag{14}$$

where ζ, η are the weight coefficients, which are subject to $\zeta + \eta = 1$. ω is the penalty factor of the penalty term, which is commonly smaller than ζ and η.

The parameters of the chosen critic-network are then updated using the loss function and stochastic gradient descent (SGD) method described previously. The online actor-network loss function is represented as:

$$\mathcal{J}_{\text{act}}(\theta^\mu) = Q_{\text{avg}}(s, a; \theta^Q) \tag{15}$$

Due to the ensemble DDPG model having M online and target critic-networks, and in each step, only one critic-network will be trained, resulting in relatively low training efficiency. Therefore, to further improve the training efficiency, a Supervised Ensemble DDPG (SE-DDPG) is proposed, in which a supervisor is added for the ensemble DDPG model. The supervisor signal is provided by (7)–(9). The supervised objective function is defined as:

$$\mathcal{J}_{\text{sup}}(\theta^\mu) = (\pi_{\text{sup}}(s) - \mu(s; \theta^\mu))^2 \tag{16}$$

TABLE I
PSEUDO-CODE OF SUPERVISED ENSEMBLE DDPG

```
Algorithm 1 Pseudocode of the training procedures
Initialization:
  Replay buffer $\mathcal{D}$, batch size $\mathcal{B}$
  Set max episodes $\mathcal{E}$, max environment steps $N$
  Set online actor-network $\theta^\mu$, target actor-network $\theta^{\mu'} \leftarrow \theta^\mu$
  Set the number of online and target critic-networks $M$ and
    parameters $\{\theta_i^Q, \theta_i^{Q'}\}_{i=1,2,\ldots\mathcal{M}}$;
for $episode = 1 : \mathcal{E}$ do:
    $step = 0$
    for $step = 1 : \mathcal{N}$ do:
        Observe the state $s_t$, calculate action $a_t = \mu(s_t; \theta^\mu)$,
          obtain supervisor signal $\pi_{sup}(s_t)$;
        Calculate standard deviation $\sigma$ and mean value $E$ of $\mathcal{M}$
          online critic-network Q-values;
        Execute action $a_t$ and obtain the next state $s_{t+1}$ and reward $r_t$,
          $step = step + 1$;
        Store$\{s_t, a_t, s_{t+1}, r_t, \pi_{sup}(s_t)\}$ to the replay buffer $\mathcal{D}$
        Randomly sample a batch of memory $\mathcal{B}$ from replay buffer $\mathcal{D}$;
        Update online actor-network via SGA and loss function
          $\mathcal{J}(\theta^\mu) = \rho \mathcal{J}_{sup}(\theta^\mu) - (1-\rho)\mathcal{J}_{act}(\theta^| \mu)$;
        Randomly select an online critic-network $Q_i(s, a; \theta_i^Q)$ and
          update it via SGD and loss function $\mathcal{L}(\theta_i^Q)$
          $= \zeta \mathcal{L}_{avg}(\theta^Q) + \eta \mathcal{L}_{td_i}(\theta_i^Q) + \omega(Q_i - Q_{avg})^2$
    end
end
```

where $\pi_{\text{sup}}(s)$ is the action value of the supervisor.

The loss function of the SE-DDPG online actor-network is represented as:

$$\mathcal{J}(\theta^\mu) = \rho \mathcal{J}_{\text{sup}}(\theta^\mu) - (1 - \rho) \mathcal{J}_{\text{act}}(\theta^\mu) \tag{17}$$

where ρ is the factor that balances supervised learning and RL, and it needs to be decayed at each step to improve the exploration performance.

$$\rho_t = \lambda \rho_{t-1} \tag{18}$$

where λ is the decay factor, t means time step.

The pseudo-code of the SE-DDPG algorithm is shown in Table I.

C. Switching Mechanism Design

SE-DDPG utilizes M parallel online critic-networks to implicitly model the distribution of Q-values. Thus, motivated by [35], the model uncertainty of SE-DDPG policy can be represented via standard deviation σ of M online critic-network Q-values, which can be used to estimate the reliability of actions generated by SE-DDPG policy. The uncertainty of the model is quantified as follows:

$$\mathcal{C} = \frac{\sigma}{E} \tag{19}$$

where E is the mean value of M online critic-network Q-values.

Note that if $\mathcal{C}$ is less than 0, the SE-DDPG policy is unreliable as the mean Q-value as the mean Q-values E is negative. Besides, the value of $\mathcal{C}$ will converge to a relatively lower range when the reward of SE-DDPG policy converges after training. When the model uncertainty $\mathcal{C}$ is obtained, the next step is to define the safe threshold bound $\mathcal{C}_b$. Subsequently, the switching mechanism can be designed based on the safe threshold to integrate both RL

TABLE II
PSEUDO-CODE OF ROBUST DECISION-MAKING FRAMEWORK

```
Algorithm 2 Robust decision-making framework
Initialization:
  Initialize the traffic environment
  Initialize the SE-DDPG policy: $\pi_{rl}$, PB policy: $\pi_{pb}$
   and safe threshold bound $\mathcal{C}_b$
Operation:
    Observe traffic environment state $S_t$
    $\pi_{rl}$ calculates the action $\pi_{rl}(S_t)$ and model uncertainty $\mathcal{C}_t$
    if: $\mathcal{C} \in \mathcal{C}_b$
        Final action $A_{st} = \pi_{rl}(S_t)$
    else:
        Final action $A_{st} = \pi_{pb}(S_t)$
    end
  Execute final action $A_{st}$
  Transit into next operation loop until terminal condition triggered.
```

TABLE III
PARAMETERS OF SURROUNDING VEHICLE CONTROLLER (LEFT) AND RL CONTROLLER (RIGHT)

Parameter	Value	Parameters	Value
a_{max}	$4m/s^2$	l_r	$2.5m$
α	4	L_1	$10m$
v_{ex}	$33m/s$	L_2	$180m$
d_0	$5m$	k_1	1
T	$1s$	k_2	1
b	$-4m/s^2$	k_3	0.5
b_{safe}	$4m/s^2$	k_4	0.3
p	0.001	k_5	0.2
a_{th}	$0.2m/s^2$	k_6	0.3
K_p^{lat}	1.6	v_{min}	$17m/s$
K_p^{θ}	5	v_{max}	$33m/s$

policy and PB policy, which is referred to as the robust decision-making framework (RDMF).

$$\pi_{RDMF} = \begin{cases} \pi_{rl}, & \text{if } \mathcal{C} \in \mathcal{C}_b \\ \pi_{pb}, & \text{otherwise} \end{cases} \tag{20}$$

The safe threshold bound $\mathcal{C}_b$ is investigated and determined via the experiment in Section V. Finally, the pseudo-code of the proposed RDMF is shown in Table II.

IV. EXPERIMENT SETUP

A. Driving Scenarios Settings

In this study, to test the effectiveness of the proposed decision-making method, the highway driving scenario is constructed utilizing the highway-env simulator [39]. The training scenario consists of a three-lane highway with a 4-meter-wide lane. The vehicle's length and width are set to 5 m and 2 m, respectively. While the kinematics are updated, the vehicles that exceed the roadway will be removed from view. To ensure that the ego vehicle can pass other vehicles, the initial speed of the ego vehicle in each episode is set to 25 m/s, while the initial speed of surrounding vehicles (SVs) is randomly chosen from [23,25 m/s]. The initial lane of each vehicle is also chosen at random. The defined vehicle density in the highway-env simulator is 1. IDM and MOBIL models allow for the speed and driving lane of SVs to change at any time during the driving process.

A kinematic bicycle model is used to describe the vehicle's motion, which considers the left and right wheels as a single wheel and assumes that the front wheel controls the steering motion and there is no sliding. The following are the formulas of the kinematic model:

$$\begin{aligned} \dot{x} &= v\cos(\theta+\beta) \\ \dot{y} &= v\sin(\theta+\beta) \\ \dot{v} &= a \\ \dot{\theta} &= \frac{v\sin\beta}{l_r} \\ \beta &= \arctan\frac{l_r\tan\delta}{l_f+l_r} \end{aligned} \tag{21}$$

where x and y are the longitudinal positions and lateral positions, respectively. θ represents the heading angle of the vehicle, β is the sideslip angle at the mass center, v denotes vehicle speed, a is acceleration, l_f and l_r represent the distance between the mass center of the vehicle and the front axle and the distance between the mass center of the vehicle and the front axle, respectively, and δ is the steering angle of the front wheel. The corresponding parameters are given in Table III.

B. Markov Decision Process Design

This part introduces the implementations of RL elements: state, action, and reward. In this work, the state S includes the position, speed, and heading of the vehicles, which is defined as:

$$\begin{aligned} S &= (S_e, S_i) \\ S_e &= (p_e, x_e, y_e, v_{xe}, v_{ye}, sin\theta_e) \\ S_i &= (p_i, \Delta x_i, \Delta y_i, \Delta v_{xi}, \Delta v_{yi}, \sin\theta_i) \\ s.t. &- L_1 < \Delta x_i < L_2, i \leq N \end{aligned} \tag{22}$$

where S_e is the state of ego vehicle, S_i is the state of SVs, p_i denotes the indicator, which is set as 1 if the vehicle i exists in the simulator, otherwise 0, x and y are the longitudinal positions and lateral positions respectively. v_x and v_y are the longitudinal position and lateral speed respectively. Δ refers to the relative value between the ego vehicle and the SV, θ is the heading angle of the vehicle. L_1 and L_2 are backward and forward detection distance respectively, which means that we only consider vehicles within this range. The actions of SE-DDPG are the acceleration a and the steering angle δ. The action space is defined as:

$$A = [a, \delta], s.t. a \in [-4, 4]m/s^2, \delta \in [-0.1, 0.1]rad \tag{23}$$

The reward function in this work consists of safety, efficiency, comfort, and rules. 1) Safety: avoid veering off the roadway and colliding with other vehicles. 2) Efficiency: driving as quickly as possible within the speed limit. 3) Comfort: low lateral acceleration while driving. 4) Rules: driving in the rightmost lane and on the lane's center line is prohibited. The reward function is defined as follows:

$$\begin{aligned} R &= R_{\text{safe}} + R_{\text{efficiency}} + R_{\text{comfort}} + R_{\text{rule}} \\ R_{\text{safe}} &= k_1 r_{\text{collsion}} + k_2 r_{\text{out}} \end{aligned}$$

$$\begin{aligned}
r_{\text{collision}} &= \begin{cases} -15, \text{ if } Collision = True \\ 0, \text{ else} \end{cases} \\
r_{\text{out}} &= \begin{cases} -1, \text{ if } out = True \\ 0, \text{else} \end{cases} \\
R_{\text{efficiency}} &= \begin{cases} k_3 \frac{v - v_{\min}}{v_{\max} - v_{\min}}, \text{if } v \in [v_{\min}, v_{\max}] \\ -0.5, \text{ else} \end{cases} \\
R_{\text{comfort}} &= k_4 \left(1 - \left|\frac{v_\delta}{4}\right|\right) \\
R_{\text{rule}} &= k_5 \left(\frac{l}{l_{\text{right}}}\right) + k_6 \left|\frac{y_{\text{center}}}{w_{\text{center}}}\right|
\end{aligned} \tag{24}$$

where k_1, k_2, k_3, k_4, k_5, k_6 are the corresponding weighting factors in the reward function, $v_{\min}$ and $v_{\max}$ represent minimum speed and maximum speed, respectively. l represents the lane index of ego vehicle, l_{right} is the lane index of the rightmost lane, y_{center} is the lateral distance between ego vehicle and the center line of the current lane, and w_{center} represents the distance between the center line of the lane and the boundary of the lane. Parameters of this part are also presented in Table III.

C. Implementation Details

1) Network Architecture: The critic-network comprises 4 online Q networks and 4 target Q networks with identical architecture. In addition, each Q network consists of four layers: layer 1 has 26 input units for state and action information, layers 2 and 3 are the hidden layers with 128 units and ReLu, and layer 4 outputs the Q value.

The actor-network includes one online policy network and one target policy network with the same architecture. Each policy network has four layers: layer 1 has 24 units for state information input, layers 2 and 3 are the hidden layers with 128 units and ReLu, and layer 4 outputs the action value using the Tanh function. The actor's and critic's learning rates are set to 0.0001 and 0.001, respectively. The coefficient for the soft update is 0.01 and the discount factor is 0.9. Finally, the batch size and buffer size are set as 1024 and 10^5.

2) Training Process: In this study, the RL agent is trained for 3500 episodes with a maximum step of 800 per episode (policy frequency is 20 Hz, *i.e.*, each episode runs 40 seconds). Each episode will be terminated when the ego vehicle reaches the goal, or a collision occurs between the ego vehicle and other traffic participants, or the number of steps reaches the max steps. The positions and velocities of SVs are initialized randomly at the beginning of each episode of the training process, and the driving environment is reset if an episode is terminated.

3) Policy Performance Metrics:

- *Success Rate:* the percentage of AVs that reach the maximum number of steps without colliding; a higher success rate indicates the policy's superior safety performance.
- *Average Reward:* the ego vehicle is expected to explore a driving strategy in a dynamic highway scenario, which must be safe (no collision) and efficient (high speed), and comfortable (low lateral acceleration). Thus, this metric is used to quantify the overall performance of the policy.
- *Average Speed:* the average speed of AV is used to represent the time efficiency for a successful episode, and a higher average speed indicates higher time efficiency.
- *Computation Time:* this metric reflects the real-time performance of the agent, which is calculated based on the NVIDIA Jetson AGX Xavier platform.

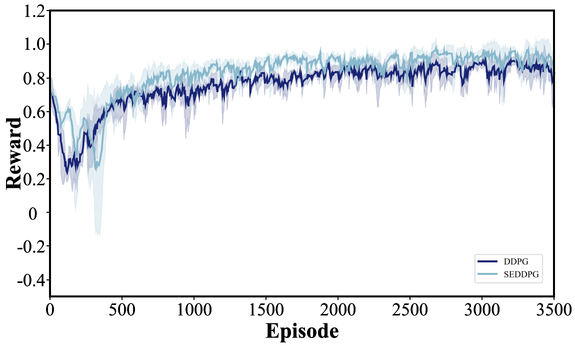


Fig. 2. Policy convergence. One episode terminates if the number of timesteps reaches 800, the ego vehicle crashes with other vehicles, or the ego vehicle runs out of the road. The accumulated reward sums the normalized reward achieved at each timestep during one episode.

4) Comparison Baselines: Several methods are implemented as benchmarks to compare the performance and efficacy of the proposed method to those of the baselines. The baseline methods are listed as follows.

- *DQN:* DQN is a well-known RL algorithm with discrete action space. In the comparison experiment, the action space is defined as turn left, turn right, speed up, slow down, and idle. These high-level actions commanded by DQN will be tracked by a lower controller, *i.e.*, a PID controller.
- *DDPG:* the vanilla DDPG algorithm is implemented, and the network structure, input observation, and output control are the same as that in the proposed method.
- *SAC:* the soft actor-critic (SAC) is a state-of-the-art off-policy RL algorithm, which optimizes a trade-off between the expected return and entropy.
- *SE-DDPG:* the SE-DDPG built-in Section III-B without PB policy integration is also used for comparison.
- *IDM+MOBIL:* the IDM and MOBIL models mentioned in Section III-B are utilized for comparison. The parameters of IDM and MOBIL are set the same as SVs, as shown in Table III.

V. Results and Discussion

A. Policy Convergence

This subsection describes the SE-DDPG training procedure. It is expected that the SE-DDPG will enable the AV to achieve the maximum accumulative reward in a single episode and output risk information regarding its actions. In the training process, an increase in the accumulative reward indicates a policy improvement. The convergence of the accumulative reward indicates that the policy has reached its local maximum reward. Fig. 2 depicts the training process of the proposed SE-DDPG and the

TABLE IV
Test Result of Different Threshold Upper Bound

Upper bound	Success rate	π_{pb} activation rate	Average speed(m/s)	Average driving distance (m)
0.04	**100.00%**	**14.32%**	**26.09**	**1034.78**
0.05	99.17%	7.42%	26.10	1039.14
0.06	97.50%	5.75%	26.27	1040.93

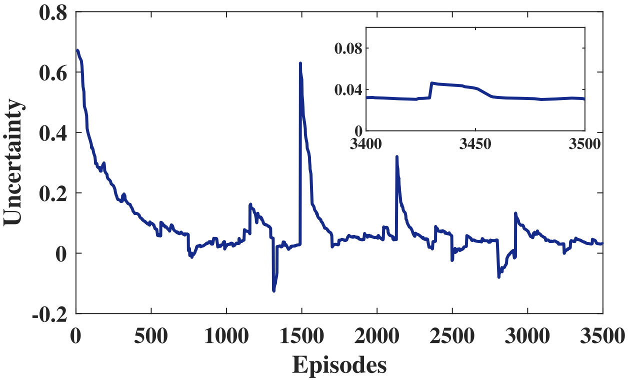


Fig. 3. Average uncertainty value curve of SE-DDPG during the training process.

Vanilla DDPG. As experience grows, the cumulative reward attained by SE-DDPG increases and eventually converges. The fluctuation of the cumulative reward during the converged phase is primarily attributable to the uncertainty of the driving scenario. Even when the RL policy has been converged after training, it is notoriously difficult to guarantee that RL-based AVs are 100% collision-free with other traffic participants. Compared with DDPG, the convergence rate of SE-DDPG is close to its, which means the low training efficiency issue of ensemble DDPG is mitigated. Moreover, the final converged reward of SE-DDPG is significantly greater than that of DDPG.

B. Investigation and Modification of Switching Threshold

As mentioned previously, the model uncertainty of SE-DDPG policy, *i.e.*, $\mathcal{C}$ will converge to a relatively lower range after training. Fig. 3 demonstrates the variation of $\mathcal{C}$ during the SE-DDPG training process. As shown in Fig. 3, $\mathcal{C}$ is converged to a relatively lower range, indicating that the trained SE-DDPG model is well-suitable for the current training scenario. To design the switch mechanism of the proposed framework, a reasonable bound of the $\mathcal{C}$ should be determined. Firstly, if the value $\mathcal{C}$ is less than 0, the DDPG policy is unreliable as the mean Q-values are negative. Therefore, the lower bound of the switching threshold is set as 0. As shown in Fig. 3, the value of $\mathcal{C}$ is converged to around 0.03. To determine the reasonable upper bound of the switching threshold, the following values are selected: 0.04, 0.05, 0.06. Thus, the potential safe bounds $\mathcal{C}_b$ are [0 0.04], [0, 0.05], and [0, 0.06]. Besides, to avoid unnecessary switching and improve the switching stability, the mean value of model uncertainty at the last three steps is utilized as the final model uncertainty at the current step. The test experiment is implemented, with each option bound being tested with 120 episodes and max steps of 800; the results are presented in Table IV. Therefore, the safe bound $\mathcal{C}_b$ is set as [0 0.04] as safety is the priority.

C. Performance Evaluation

The test scenario in this subsection is identical to the training scenario, except for the random seed used in the experiments. Using 120 episodes, the performance of the proposed framework and baseline methods is evaluated. Table V compares the performance of the proposed RDMF to that of other baseline methods. Detail-wise, the IDM +MOBIL strategy has a 100% success rate, indicating that there is no collision with other SVs. Nevertheless, the average speed is lower than that of SVs (23-25 m/s), and the average reward is only 0.84. Due to the rigid and strict rules, it can be concluded that IDM+MOBIL's driving policy is too conservating for passing other SVs in this scenario. In addition, DQN, DDPG, and SAC demonstrate that their policies are more effective than IDM+MOBIL. Their respective average speeds are 27.89 m/s, 25.45 m/s, and 25.92 m/s. However, safety is not guaranteed, and the success rates are only 92.50%, 74.44%, and 95.83%. In addition, the success rate and average speed of SE-DDPG are higher than those of standard DDPG (77.44% and 25.45 m/s, respectively). Moreover, the RDMF that combines SE-DDPG and IDM+MOBIL outperforms all other baselines. Specifically, the proposed RDMF has a success rate of up to 100%, improving 8.33% over SE-DDPG. The reason is that RDMF activates the π_{pb} policy when the model uncertainty C of SE-DDPG is high. However, the average speed of RDMF (25.36 m/s) is relatively smaller than SE-DDPG (26.42 m/s) as the policy π_{pb} is activated several times, *i.e.*, 18.93%. The process of the experiment indicates that the π_{pb} policy commonly chooses to deaccelerate to avoid collision with SVs. On the other hand, it can also be discovered that the safety of RDMF and IDM+MOBIL is as good, but RDMF is smarter and more efficient. A Jetson AGX Xavier embedded system is used to obtain the computation time for these methods. The proposed RDMF has a calculation time (each step) of 10.52 ms and a control frequency of 20 Hz (50 ms). In conclusion, the RDMF can satisfy real-time requirements. Note that the computation time of RDMF is relatively high compared with other methods, *e.g.*, SE-DDPG. The reason is that RDMF adds a module to calculate the model uncertainty level $\mathcal{C}$.

D. Robust Testing

As previously emphasized, RL policy tends to be unsafe in the face of unknown or unseen traffic scenarios. To demonstrate the robustness of the proposed RDMF method, the following two testing scenarios are considered for RDMF testing. First, it is well known that the input observation of RL policy is crucial, and

TABLE V
POLICY EVALUATION RESULTS

Methods	Success Rate(%)	π_{pb} Activation Rate	Average Reward	Average Speed (m/s)	Computation Time (ms)
IDM+MOBIL	100.00 ± 0.00	/	0.84 ± 0.01	21.51 ± 0.05	1.81 ± 0.16
DQN	92.50 ± 1.36	/	0.99 ± 0.02	27.89 ± 0.05	1.98 ± 0.12
DDPG	74.44 ± 1.96	/	0.81 ± 0.01	25.45 ± 0.24	2.10 ± 0.24
SAC	95.83 ± 0.70	/	0.97 ± 0.01	25.92 ± 0.15	2.32 ± 0.26
SE-DDPG	91.67 ± 0.60	/	0.94 ± 0.01	26.42 ± 0.14	1.70 ± 0.32
RDMF (Ours)	**100.00 ± 0.00**	**18.93 ± 1.88%**	**0.92 ± 0.01**	**25.36 ± 0.31**	**10.52 ± 0.23**

TABLE VI
ROBUST TESTING RESULTS OF ADDING SENSOR NOISE AND INCREASING TRAFFIC DENSITY

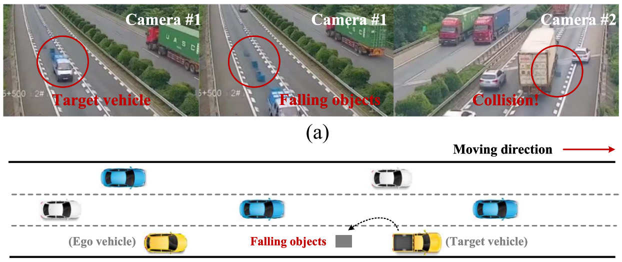
Cases	Methods	Success Rate(%)	π_{pb} Activation Rate	Average Reward	Average Speed (m/s)
Sensor Noise (Gaussian noise 20%)	DQN	58.61 ± 1.04	/	0.78 ± 0.01	27.64 ± 0.09
	DDPG	51.39 ± 3.07	/	0.67 ± 0.02	26.92 ± 0.16
	SAC	91.67 ± 0.68	/	0.95 ± 0.01	26.40 ± 0.11
	SE-DDPG	69.17 ± 5.31	/	0.83 ± 0.02	27.48 ± 0.09
	RDMF (Ours)	**100.00 ± 0.00**	**43.87 ± 1.69%**	**0.87 ± 0.01**	**24.58 ± 0.22**
Traffic Density (Density increases 50%)	DQN	18.89 ± 0.40	/	0.33 ± 0.07	25.33 ± 0.08
	DDPG	39.44 ± 5.19	/	0.45 ± 0.04	23.57 ± 0.21
	SAC	63.61 ± 3.21	/	0.60 ± 0.02	23.00 ± 0.14
	SE-DDPG	55.26 ± 3.36	/	0.52 ± 0.02	23.00 ± 0.08
	RDMF (Ours)	**99.45 ± 0.39**	**42.32 ± 0.76%**	**0.72 ± 0.01**	**21.07 ± 0.10**

in this test, Gaussian noises are added to the input observation S. The observation with noise can be represented as follows.

$$S_{\text{noise}} = S + \kappa * N(0, \Lambda^2) \tag{25}$$

where S is the original states variable presented in (22), κ is the noise proportion, $N(0, \Lambda^2)$ means Gaussian distribution with mean value 0 and variance $\Lambda^2 = diag(\sigma_x^2, \sigma_y^2, \sigma_{vx}^2, \sigma_{vy}^2, \sigma_{\sin\theta}^2)$, $\sigma_x = 10\,\text{m}$, $\sigma_y = 1\,\text{m}$, $\sigma_{vx} = 2\,\text{m/s}$, $\sigma_{vy} = 0.2\,\text{m/s}$, $\sigma_{\sin\theta} = 0.1$ rad, and the p is set as 20%. The test results for sensor noise is shown in Table VI. It can be discovered that the performance of DQN, DDPG, and SE-DDPG degrades by a large margin compared with Table V. In detail, the success rate of them is only 58.61 %, 51.39%, 69.17%. However, SAC policy shows relatively well robustness of sensor noise, and the success rate decreases only by 4.16%. In terms of robustness, the proposed RDMF method outperforms other strategies. The success rate is still up to 100 %, and in this test, the activation rate of π_{pb} reaches 43.87%. It indicates that the model uncertainty of the SE-DDPG policy is high and that the SE-DDPG and PB policies often take over to ensure safety.

The second experiment is implemented considering the change in traffic density. In the training process, the traffic density (a parameter defined by the highway-env simulator) is set to 1 (spare traffic). In this test, traffic density is set to 1.5 (dense traffic), an increase of 50%. This test is assumed to be an out-of-distribution test from light to heavy traffic. The results are displayed in Table VI. Compared to V, the success rates of DQN and DDPG have significantly decreased, falling to 18.89% and 39.44%, respectively. In addition, the performance of SAC and SE-DDPG declines, with respective success rates of 63.61% and 55.26%. Compared to this, the success rate of the proposed RDMF method remains at 99.45%. In this test, the activation rate of policy π_{pb} is up to 42.32%, which means the RL policy inside RDMF does not perform well. Thus, there is a need for policy π_{pb} to take over RL policy π_{rl}. In addition, the average speed of RDMF decreases compared with V, and the reason is that the traffic density increases and the possibility of safe overtaking decreases.



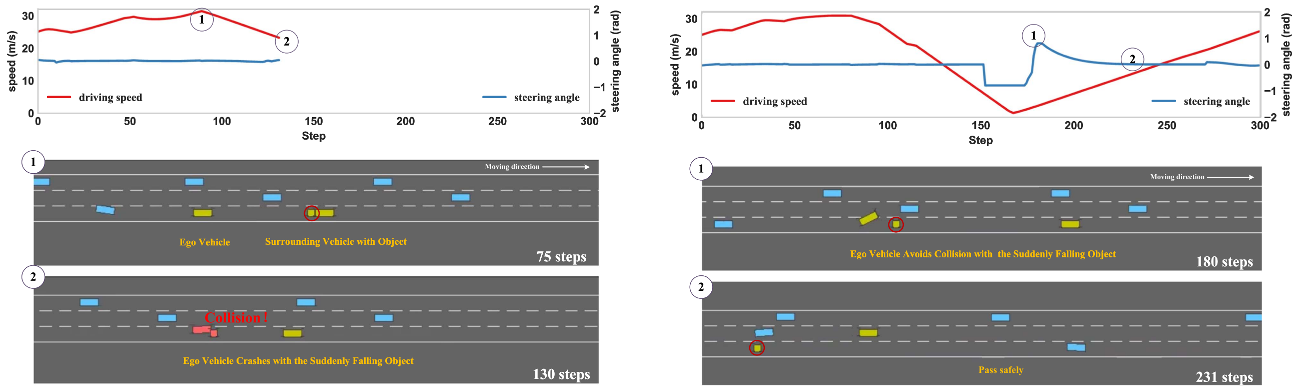
(a)

(b)

Fig. 4. Demonstration of objects falling from front vehicles. (a) a real traffic accident occurred in Zhejiang, China; (b) a similar case is reconstructed in the highway-env simulator.

E. The Analysis of Two Specific Cases

1) Objects Falling From Front Vehicles Case: On the highway, falling objects from vehicles in front pose a hazard to following vehicles [40], resulting in several severe accidents in the real world as shown in Fig. 4(a). From the picture, it can be found that the cargo placed on the target vehicle suddenly fell and the vehicle behind collided with the cargo. This scenario is challenging for the AV from the perspective of decision-making since it is hard to consider all such long-tail scenarios like this during the development phase. Therefore, this subsection reconstructs a similar situation in the highway-env simulator shown in Fig. 4(b). Specifically, an SV traveling at 25m/s carrying a load that abruptly drops off at a specific time. This type of situation is also not covered in the training process, making it an edge case for RL-based decision-making. In addition, the AV will follow this SV to test the safety of the proposed RDMF method, with the



(a) (b)

Fig. 5. Comparison results of SE-DDPG and RDMF policy in objects falling from front vehicles case. (a) SE-DDPG policy; (b) RDMF policy.

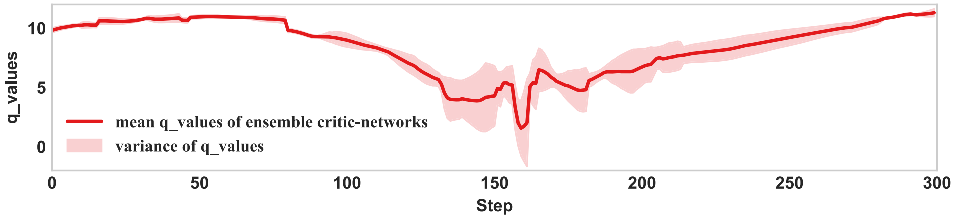


Fig. 6. Mean ensemble critic-networks Q-value and its variance of RL policy within RDMF in objects falling from front vehicles case.

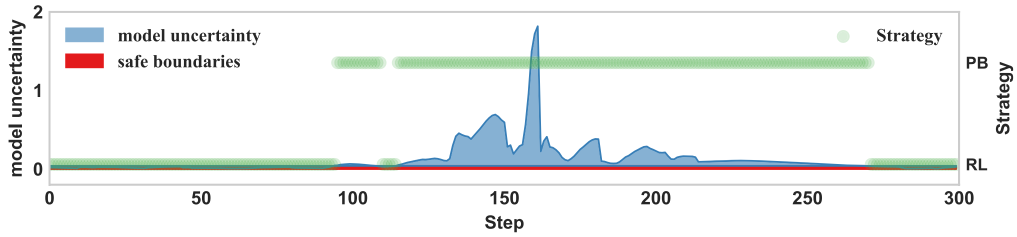
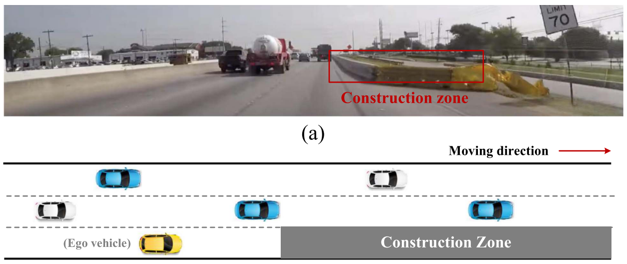


Fig. 7. Decision-making uncertainty and the corresponding process of RDMF policy in objects falling from front vehicles case.

SE-DDPG policy serving as a comparison. Other environments are identical to the training procedure

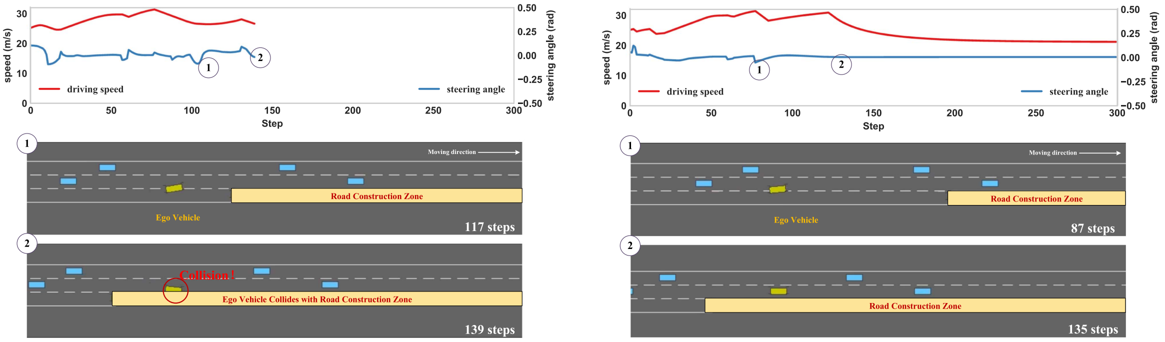
Fig. 5(a) demonstrates that the AV driven by the SE-DDPG policy collides with the falling object. It indicates that the SE-DDPG strategy is incapable of handling this unseen circumstance, resulting in the collision at the 130 steps. Fig. 5(b) depicts the driving performance of RDMF. The corresponding decision-making process is illustrated in Figs. 6 and 7. The mean ensemble critic-networks Q-value and its variance are given in Fig. 6, which is the decision-making basis of the RL policy inside the RDMF. It can be concluded that when the AV approaches the falling object (unseen scenario), the variance of ensemble critic-networks Q-value increases, indicating that the RL policy within the RDMF is not confident. Based on this, the final model uncertainty is determined as shown in Fig. 7. It reveals that RDMF recognizes this situation has not been previously addressed, and the model uncertainty increases significantly (the peak value reaches approximately 1.8), far beyond the safety threshold range [0, 0.04]. In other words, the RL policy within RDMF is unreliable at this time, and the AV is taken over by the PB policy. The RDMF does not employ the RL strategy until the vehicle has left the hazardous area.



(a) (b)

Fig. 8. Demonstration of temporary road construction. (a) a real road construction case causing changes to the road's structure; (b) a similar case is reconstructed in the highway-env simulator.

2) *Road Construction Case:* Real-world traffic situations occasionally involve road construction or other temporary traffic control events, resulting in temporary changes to the road's structure [41], as shown in Fig. 8(a). Such ad hoc events pose



(a)

(b)

Fig. 9. Comparison results of SE-DDPG and RDMF policy in temporary road construction case. (a) SE-DDPG policy; (b) RDMF policy.

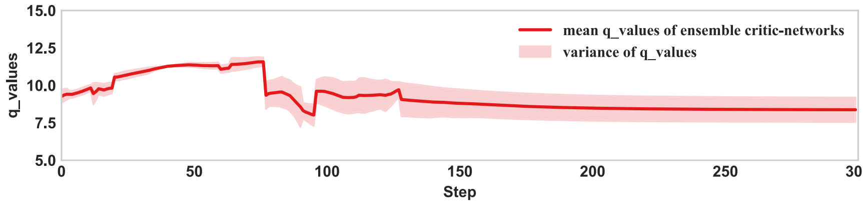


Fig. 10. Mean ensemble critic-networks Q-value and its variance of RL policy within RDMF in temporary road construction case.

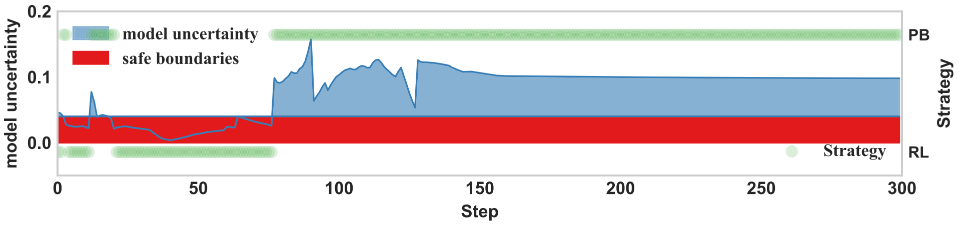


Fig. 11. Decision-making uncertainty and the corresponding process of RDMF policy in temporary road construction case.

a challenge to the RL-based decision-making approaches, as it is difficult to cover all such temporary traffic control events in a training process. Besides, re-training the RL model can be worthless for temporary traffic control.

To further illustrate and validate the effectiveness of the proposed RDMF, a similar case in Fig. 8(b) is reconstructed in the highway-env simulator. A three-lane road temporarily becomes a two-lane road. Likewise, this situation is not covered in the training scenario. The performance of the SE-DDPG policy is shown in Fig. 9(a). The AV driven by the SE-DDPG policy fails to recognize this unseen situation and crashes into the construction zone's perimeter. In contrast, it is evident from Fig. 9(b) the RDMF policy operates safely in this unknown situation without collisions. The mean ensemble critic-networks Q-value and its variance are given in Fig. 10. The model uncertainty and the corresponding decision-making process are shown in Fig. 11. Specifically, when the RDMF-controlled AV encounters this case, the model uncertainty immediately increases to around 0.1, which also exceeds the safe bound. Because the training process does not include this two-lane road scenario. Therefore, the RDMF always adopts the PB policy, *i.e.*, π_{pb} when the three-lane road becomes a two-lane road as the RL policy π_{rl} is not reliable. Finally, supplementary videos[1] of objects falling and road construction cases are provided.

VI. Conclusion

This paper proposes a robust decision-making framework for autonomous highway driving to enhance driving safety. First, a DDPG-based continuous action space RL policy is developed. Considering that RL policy is prone to be unsafe in the presence of unseen traffic scenarios, *i.e.*, out-of-distribution test data, the model uncertainty of the DDPG policy is quantified at runtime via ensemble technique. In addition, a principle-based policy, *i.e.*, IDM+MOBIL is implemented as a complementary policy. When model uncertainty is high, the principle-based policy takes precedence over the RL policy. Finally, several challenging scenarios are considered to validate the performance of the proposed method. The results demonstrate that the proposed algorithm is robust and secure.

References

[1] A. Jain, L. D. Pero, H. Grimmett, and P. Ondruska, "Autonomy 2.0: Why is self-driving always 5 years away?," 2021, *arXiv:2107.08142*.

[2] Z. Ju, H. Zhang, and Y. Tan, "Distributed deception attack detection in platoon-based connected vehicle systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 4609–4620, May 2020.

[3] X. Tang, J. Chen, K. Yang, M. Toyoda, T. Liu, and X. Hu, "Visual detection and deep reinforcement learning-based car following and energy management for hybrid electric vehicles," *IEEE Trans. Transport. Electrific.*, vol. 8, no. 2, pp. 2501–2515, Jun. 2022.

[1] https://github.com/Kayne0401/Robust-Decision-Making-Framework

[4] K. Yang, X. Tang, Y. Qin, Y. Huang, H. Wang, and H. Pu, "Comparative study of trajectory tracking control for automated vehicles via model predictive control and robust h-infinity state feedback control," *Chin. J. Mech. Eng.*, vol. 34, no. 1, pp. 1–14, 2021.

[5] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature Commun.*, vol. 12, no. 1, pp. 1–14, 2021.

[6] S. S. Ho, "Complementary and competitive framing of driverless cars: Framing effects, attitude volatility, or attitude resistance?," *Int. J. Public Opin. Res.*, vol. 33, no. 3, pp. 512–531, 2021.

[7] Z. Ju, H. Zhang, X. Li, X. Chen, J. Han, and M. Yang, "A survey on attack detection and resilience for connected and automated vehicles: From vehicle dynamics and control perspective," *IEEE Trans. Intell. Veh.*, vol. 7, no. 4, pp. 815–837, Dec. 2022.

[8] T. Zhao, E. Yurtsever, J. A. Paulson, and G. Rizzoni, "Formal certification methods for automated vehicle safety assessment," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 232–249, Jan. 2023. [Online]. Available: https://doi.org/10.1109%2Ftiv.2022.3170517

[9] X. Tang, Z. Zhang, and Y. Qin, "On-road object detection and tracking based on radar and vision fusion: A review," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 5, pp. 103–128, Sep./Oct. 2021.

[10] J. Chen, Z. Shuai, H. Zhang, and W. Zhao, "Path following control of autonomous four-wheel-independent-drive electric vehicles via second-order sliding mode and nonlinear disturbance observer techniques," *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2460–2469, Mar. 2021.

[11] X. Tang et al., "Driving environment uncertainty-aware motion planning for autonomous vehicles," *Chin. J. Mech. Eng.*, vol. 35, no. 1, pp. 1–14, 2022.

[12] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, no. 2, 2000, Art. no. 1805.

[13] X. Tang et al., "Prediction-uncertainty-aware decision-making for autonomous vehicles," *IEEE Trans. Intell. Veh.*, vol. 7, no. 4, pp. 849–862, Dec. 2022.

[14] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine, "Uncertainty-aware reinforcement learning for collision avoidance," 2017, *arXiv:1702.01182*.

[15] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5876–5888, Jun. 2020.

[16] C.-J. Hoel, K. Wolff, and L. Laine, "Automated speed and lane change decision making using deep reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2148–2155.

[17] X. Tang, B. Huang, T. Liu, and X. Lin, "Highway decision-making and motion planning for autonomous driving via soft actor-critic," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4706–4717, May 2022.

[18] Z. Huang, J. Wu, and C. Lv, "Efficient deep reinforcement learning with imitative expert priors for autonomous driving," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 26, 2022, doi: 10.1109/TNNLS.2022.3142822.

[19] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2111–2117.

[20] B. D. Ziebart et al., "Maximum entropy inverse reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, Chicago, IL, USA, 2008, vol. 8, pp. 1433–1438.

[21] D. Chen et al., "Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic," 2022, *arXiv:2105.05701*.

[22] S. Nageshrao, H. E. Tseng, and D. Filev, "Autonomous highway driving using deep reinforcement learning," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2019, pp. 2326–2331.

[23] J. Lubars et al., "Combining reinforcement learning with model predictive control for on-ramp merging," in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 942–947.

[24] Z. Cao, S. Xu, H. Peng, D. Yang, and R. Zidek, "Confidence-aware reinforcement learning for self-driving cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7419–7430, Jul. 2022.

[25] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[26] I. Osband, C. Blundell, A. Pritzel, and B. V. Roy, "Deep exploration via bootstrapped DQN," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29, pp. 4026–4034.

[27] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, Univ. Cambridge, U.K., 2016.

[28] M. Benatan and E. O. Pyzer-Knapp, "Fully Bayesian recurrent neural networks for safe reinforcement learning," 2019, *arXiv:1911.03308*.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] B. Lütjens, M. Everett, and J. P. How, "Safe reinforcement learning with model uncertainty estimates," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 8662–8668.

[31] O. Lockwood and M. Si, "A review of uncertainty for deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell. Interactive Digit. Entertainment*, 2022, vol. 18, no. 1, pp. 155–162.

[32] M. Abdar et al., "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Inf. Fusion*, vol. 76, pp. 243–297, 2021.

[33] I. Osband, J. Aslanides, and A. Cassirer, "Randomized prior functions for deep reinforcement learning," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 8617–8629, 2018.

[34] C.-J. Hoel, K. Wolff, and L. Laine, "Tactical decision-making in autonomous driving by reinforcement learning with uncertainty estimation," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1563–1569.

[35] C.-J. Hoel, K. Wolff, and L. Laine, "Ensemble quantile networks: Uncertainty-aware reinforcement learning with applications in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–12, 2023 doi: 10.1109/TITS.2023.3251376.

[36] C.-J. Hoel, T. Tram, and J. Sjöberg, "Reinforcement learning with uncertainty estimation for tactical decision-making in intersections," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–7.

[37] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transp. Res. Rec.*, vol. 1999, no. 1, pp. 86–94, 2007.

[38] W. Liu et al., "Ensemble bootstrapped deep deterministic policy gradient for vision-based robotic grasping," *IEEE Access*, vol. 9, pp. 19916–19925, 2021.

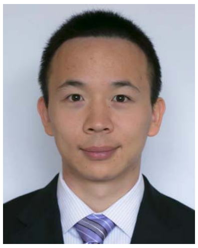
[39] E. Leurent, "An environment for autonomous driving decision-making," *GitHub Repository*, 2018. [Online]. Available: https://github.com/eleurent/highway-env

[40] D. Bogdoll, S. Guneshka, and J. M. Zöllner, "One ontology to rule them all: Corner case scenarios for autonomous driving," in *Proc. Comput. Vis. – ECCV 2022 Workshops*, 2023, pp. 409–425.

[41] J. Lin et al., "Road traffic law adaptive decision-making for self-driving vehicles," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst.*, 2022, pp. 2034–2041.

Kai Yang received the B.E. degree in vehicle engineering from the Wuhan University of Technology, Wuhan, China, in 2018. He is currently working toward the Ph.D. degree with the College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing, China. He researches as a Joint Ph.D. Student with the School of Vehicle and Mobility, Tsinghua University, Beijing, China. His research interests include motion prediction and decision-making of autonomous vehicles.

Xiaolin Tang (Senior Member, IEEE) received the B.S. degree in mechanics engineering and the M.S. degree in vehicle engineering from Chongqing University, Chongqing, China, in 2006 and 2009, respectively, and the Ph.D. degree in mechanical engineering from Shanghai JiaoTong University, Shanghai, China, in 2015. He is currently a Professor with the State Key Laboratory of Mechanical Transmissions and College of Mechanical and Vehicle Engineering, Chongqing University. He has led and has been involved in more than ten research projects. He has authored or coauthroed more than 40 papers. His research interests include hybrid electric vehicles (HEVs), vehicle dynamics, energy management, and autonomous vehicle. He is also an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON TRANSPORTATION ELECTRIFICATION.



Sen Qiu (Member, IEEE) received the B.Sc. and Ph.D. degrees in automatic control from the Dalian University of Technology, Dalian, China, in 2008 and 2016, respectively. He is currently an Associate Professor with the Dalian University of Technology. From 2013 to 2014, he was a Visiting Researcher with the Department of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K. His research interests include information fusion, wireless sensor network, wearable computing and pattern recognition.

Shufeng Jin received the B.S. degree in vehicle engineering from Fuzhou University, Fuzhou, China, and the M.S. degree in vehicle engineering from Chongqing University, Chongqing, China, in 2019 and 2022, respectively. His research interests include decision-making of autonomous vehicle and reinforcement learning.

Zichun Wei received the B.S. degree in mechanical engineering from Miami University, Oxford, OH, USA, in 2018, and the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2020. His research interests include multi objective control, data driven control and motion planning for multi-agents systems.

Hong Wang (Senior Member, IEEE) received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2015. She is currently a Research Associate Professor with Tsinghua University, Beijing, China. From 2015 to 2019, she was a Research Associate of mechanical and mechatronics engineering with the University of Waterloo, Waterloo, ON, Canada. She has authored or coauthored more than 60 papers on top international journals. Her research interests include the safety of the on-board AI algorithm, the safe decision-making for intelligent vehicles, and the test and evaluation of SOTIF. Since 2017, she has been an IEEE Member. She is also an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON INTELLIGENT VEHICLES.