

Autonomous Driving in Highways with Safe and Efficient Lane Changes: Risk-Aware Deep Reinforcement Learning Approaches

Pedram Lamei¹, Mohammad Haeri^{2*}

¹MSc, Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

²Professor, Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

Abstract:

This study presents a safety-prioritized deep reinforcement learning framework for navigation and safe lane-change decision-making in autonomous driving on highways. The framework introduces a discrete action space that combines streamlined lateral lane-change actions with longitudinal acceleration and deceleration maneuvers. A key feature of the framework is the integration of probabilistic risk assessment, grounded in Bayesian inference, to account for uncertainties in the behavior of surrounding vehicles. Unlike many prior studies, this work implements a fully autonomous decision-making system that models both lateral and longitudinal actions within a unified structure. Two deep reinforcement learning algorithms, deep Q-network and proximal policy optimization, are employed to enable the agent to make risk-aware decisions in dynamic environments. A multi-objective reward function is designed that prioritizes safety over efficiency while also considering travel speed, collision avoidance, and survival. Additionally, a dynamic highway environment is configured using seeded randomization of vehicle lanes, initial speeds, longitudinal spacing, and selected behavioral parameters, thereby generating diverse and reproducible traffic conditions. Each method is trained over five independent runs and evaluated deterministically under identical traffic seeds. The results show that incorporating risk assessment improves the success rate of both policies while reducing collisions and unnecessary overtaking. Comparative evaluations demonstrate that proximal policy optimization offers conservative behavior, often resulting in lower-speed navigation, whereas deep Q-network exhibits faster and more aggressive strategies that favor high-speed performance.¹

Keywords:

* Corresponding author's email: haeri@sharif.ir

¹ The authors used ChatGPT (OpenAI) to assist with English language editing and improve the clarity of the text. The authors reviewed and revised the AI-assisted content and take full responsibility for the final manuscript.

Deep Reinforcement Learning, Autonomous Vehicles, Decision Making, Lane Change, Safety

1. Introduction

Intelligent transportation systems (ITSs) enhance safety and efficiency but face challenges in data, integration, and security. Ongoing advancements bring both innovation opportunities and increased complexity [1-3]. Recent ITS advancements, including advanced driver assistance systems (ADASs) and autonomous vehicles (AVs), enhance safety by reducing collisions and traffic issues. AVs offer solutions to congestion, emissions, and energy use, presenting a major shift in urban mobility and driving ongoing research and innovation [4-6]. Currently, one common issue with AVs is collisions with other vehicles while changing lanes, often leading to accidents and reduced safety. Therefore, designing real-time and advanced prediction algorithms to detect the lane-changing intentions of other vehicles can significantly reduce accident rates [7-9]. Additionally, in the future, AVs will operate in complex and multifaceted traffic conditions, making it critical to ensure driving and passenger safety [10].

To date, various strategies have been developed and applied to address the challenges and needs of different driving scenarios [11, 12]. Advances in environmental perception modeling, decision-making, and vehicle control have enabled the application of autonomous driving (AD) technology in diverse scenarios [13]. However, the safety of AVs remains a fundamental challenge requiring special attention [14]. In mixed traffic environments involving human and machine interaction, the lane-changing behavior of AVs is often cautious, which can lead to inaccurate identification of the driving intentions of surrounding vehicles [7, 15]. Decision-making in AD acts as the system's intelligence, enabling adaptation to diverse and dynamic road conditions. While traditional rule-based methodologies often struggle in complex scenarios, data-driven decision-making frameworks have gained prominence for their ability to provide adaptive and intelligent responses to environmental changes [16]. Advances in AV technology have driven significant progress in decision-making research. Data-driven approaches, including machine learning, deep learning (DL), and reinforcement learning (RL), have become essential tools for researchers. Data-driven methods often outperform rule-based approaches in complex scenarios, but their performance relies on diverse and high-quality datasets. Generalizability requires

comprehensive data, as real-world effectiveness hinges on robust training and testing conditions [17-19].

Existing methods for decision-making in AVs can be categorized into three main groups: motion planning-based methods, risk assessment (RA)-based methods, and learning-based methods, which include supervised learning and RL [10, 20, 21]. RA-based methods involve two key stages: evaluating the risk of the current driving situation to identify potential hazards, and developing a sequential operational strategy to ensure safety based on the assessment. This hierarchical approach makes these methods particularly suitable for AVs and ADASs [22]. Risk assessment methods are broadly categorized into deterministic and probabilistic approaches. While computationally efficient and effective for simple scenarios like straight-road driving, deterministic methods struggle in complex situations such as lane changes and are less practical due to their inability to handle uncertainties. Probabilistic methods address the limitations of deterministic approaches by using probabilistic modeling to analyze temporal and spatial vehicle relationships. By incorporating uncertainties, they provide more accurate risk assessments in real-world conditions, making them highly effective for complex, multi-lane scenarios and real-time decision-making [20, 23]. Developing real-time prediction models that manage uncertainties is a key research area for improving the performance of AD systems. In research related to risk assessment and the improvement of AD systems, various studies have been conducted, each proposing new methods to address existing challenges. In [24] standard metrics are integrated into a Bayesian framework for evaluating risk probability, followed by rule-based vehicle control in complex scenarios such as intersections and highways. This combination enhances safety by linking risk assessment with control. In [25] vehicle-to-vehicle communication is used to predict distant vehicle positions under uncertainty. Although it improves environmental analysis, the method relies on expert knowledge and fixed rules, limiting adaptability in unpredictable scenarios [26]. Authors in [27] proposed a cooperative risk assessment approach to prevent rear-end collisions during lane changes, improving flexibility and safety in dense traffic. In [28] an emergency lane-change path planner is developed using the lateral load transfer ratio as a risk metric. A seventh-order polynomial model and phase-plane-based risk classification is applied to improve safety and stability, particularly on low-

friction roads.

Driver-state monitoring provides another important layer of automotive safety. In [29], facial analysis, machine learning, and alcohol sensing are combined to detect drowsiness and impairment. In [30], a real-time FPGA-based ADAS is developed that links drowsiness detection with autonomous vehicle stopping.

DL-based AD technologies rely on extensive, diverse driving datasets for training, but collecting such data, especially for rare dangerous scenarios like accidents, is costly and challenging. Limited critical scenario data in naturalistic datasets may hinder AV performance in real-world, high-risk situations [18, 23]. The applications of RL in AD have garnered significant attention in recent years, with advancements in deep reinforcement learning (DRL) due to its ability to manage high-dimensional state and action spaces, making it ideal for complex vehicle control [5, 31]. Recent progress in DRL has enabled researchers to address decision-making problems in AD, reducing reliance on large volumes of labeled training data. DRL-based methods learn and improve driving skills through trial-and-error processes, making them suitable for application in dangerous or near-miss scenarios to prevent accidents [7, 32]. In [33], authors developed an RL-based method enabling an agent to learn autonomous lane-changing behavior and make collision-avoidance decisions in unpredictable conditions. In [34], a proximal policy optimization (PPO)-based automated lane-change strategy is proposed, achieving efficient learning, stable performance, and safe, smooth maneuvers in dense traffic. Authors in [20] proposed a DRL-based lane-change decision-making framework to find a risk-aware strategy minimizing expected risk. A probabilistic risk assessment method and a risk-aware decision algorithm were developed and evaluated with static and moving obstacles, showing robust, safe strategies and improved performance. Brake control was handled by human drivers or ADASs. In [35], a DRL-based multi-vehicle intelligent lane-changing method is introduced, featuring a hierarchical structure with a low-level lane-changing controller and a high-level DRL decision-making system. The low-level controller includes a lateral controller using a dueling double deep Q-network for lane-change decisions and a longitudinal controller employing an intelligent driver model (IDM) for vehicle following. Results demonstrate that multi-vehicle intelligent lane-changing enables safe and efficient

lane changes for multiple vehicles, enhancing traffic efficiency and reducing congestion. In [36], authors combined high-level maneuver selection with lateral and longitudinal low-level control through hierarchical reinforcement learning. Authors in [37] introduced risk-potential fields and task decomposition for continuous lane-changing and overtaking decisions.

However, many DRL-based decision-making methods do not fully consider driving risks. This shortcoming can lead to driving strategies with low sensitivity to hazards, compromising vehicle safety [7]. To tackle this, authors in [38] proposed a risk-aware framework combining path planning with RL-based control for safe driving. Although risk assessment has been considered in some DRL-based decision-making methods, a more thorough and in-depth investigation, particularly in lane-changing scenarios in AD, remains an area requiring further research and development [7, 31]. Table 1 provides a summary of relevant prior studies.

This research contributes to the field by introducing a comprehensive framework for safe lane-change and acceleration-deceleration decision-making in AD, utilizing DRL to address the limitations of traditional rule-based and deterministic methods. Unlike most existing studies that focus solely on lateral lane-changing decisions and assume constant vehicle speed or delegate braking to ADAS or human control, this work incorporates both lateral and longitudinal actions into a unified, fully autonomous decision-making framework. By streamlining the lateral lane-change decisions and explicitly modeling longitudinal maneuvers such as acceleration and deceleration, the framework offers a more complete action space for autonomous agents. Furthermore, a probabilistic risk assessment module based on Bayesian inference is integrated to support risk-aware longitudinal control, enhancing the agent's ability to handle uncertainties in dynamic driving environments. In contrast to many existing studies that assume surrounding vehicles maintain constant speeds and fixed lanes, our environment captures the stochastic nature of real-world driving behaviors. This realistic traffic modeling introduces greater environmental uncertainty, providing a more challenging and practical training setting. The decision-making system is built on deep Q-network (DQN) and PPO agents. This study provides a detailed evaluation and comparison of DQN and PPO performance. A multi-objective reward function is designed to jointly consider safety and efficiency, while assigning greater importance to safety,

thereby guiding the agent to minimize collision risk while maintaining an effective travel speed. The contribution of this study is not the first introduction of combined lateral-longitudinal actions or probabilistic risk assessment. Instead, it provides a controlled framework for integrating the same Bayesian clearance-based safety signal into both value-based DQN and policy-gradient PPO agents using an identical five-action interface. Bayesian risk assessment has previously been applied

Table 1. Summary of relevant prior studies.

Refs.	State space	Action space	Reward	Method	Software	Lanes	Simulation setup
[20]	Position, velocity, yaw angle, and yaw rate of ego and surrounding vehicles	Left-turn or right-turn with a small or large value, straight driving without steering	Safety, efficiency	PRDQN + RA	CARLA	2	CARLA autopilot
[34]	Raw sensors data (position, velocity, and acceleration)	Lane keeping, change to the target lane, abort lane change	Safety, efficiency, comfort	PPO	SUMO	3	IDM model
[35]	Position, velocity, and acceleration of ego and surrounding vehicles	Stay, switch to the right/left lane	Safety, efficiency, comfort	D3QN	SUMO	3	IDM model
[39]	Ego and surrounding vehicles dynamics, road curvature	Continuous yaw acceleration	Safety, efficiency, comfort	DDPG	Not mentioned	3	IDM-car following
[40]	Relative distance and absolute speed of front vehicles	Stay or change lane	Safety, efficiency	DQN	MATLAB & VISSIM	2	Without random actions
[41]	Position and velocity of ego and surrounding vehicles	Lane change to left/right, do nothing	Safety, efficiency	DQN	Udacity simulator	3	Without random actions
[42]	Relative position and velocity of surrounding vehicles	speed up, slow down, cruising, turn left/right.	Safety, efficiency, comfort	MA2C	OpenAI Highway-env	2	IDM & MOBIL
[43]	Ego kinematics and position, velocity, heading angle, and size of surrounding vehicles	Continuous acceleration and steering angle	Safety, comfort	FSM and SAC	Not mentioned	2 or 3	IDM & MOBIL
[44]	Ego kinematics and position, and velocity of surrounding vehicles	Stay or change lane	Safety, efficiency	DQN	Julia	2	IDM model
Present study	Relative position and velocity of ego and surrounding vehicles	Speed up, slow down, do nothing, lane change to the left/right	Safety, efficiency	DQN/PPO + RA	OpenAI Highway-env	3	Random acceleration, deceleration, and lane changes

to lane-change prediction and decision-making; therefore, the Bayesian formulation itself is not claimed as a novel contribution. The novelty of this study lies in integrating the Bayesian risk measure as a dense, safety-prioritized reward into a unified action space containing both lateral and longitudinal maneuvers and systematically evaluating its effects on DQN- and PPO-based policies.

The remainder of this paper is organized into five sections. Section 2 defines the research problem and introduces the foundational concepts of DRL, along with the specific algorithms employed. Section 3 describes the risk assessment methodology and explains how risk evaluation is integrated into the agent's decision-making framework. Section 4 presents the simulation setup, including the state and action spaces, reward functions, agent training methodology, and environment configuration. Section 5 discusses the simulation results, analyzes the performance of the algorithms, and compares the proposed approaches. Finally, Section 6 concludes the paper and outlines potential directions for future work.

2. Problem Statement and Preliminaries

One of the most notable applications of DRL is in AD systems, where agents must make optimal decisions in real time within high-dimensional state and action spaces, all while navigating highly dynamic and uncertain environments. This capability highlights DRL's transformative potential in advancing the development of intelligent and adaptive AD technologies [45].

2-1- Deep reinforcement learning

In a DRL-based decision-making framework, an agent interacts with a stochastic environment by selecting and executing actions at each time step. Learning occurs through environmental feedback in the form of rewards, with the goal of maximizing cumulative rewards over time. This process is typically modeled as a Markov decision process (MDP), defined by the following relationship

$$M \triangleq \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle, \quad (1)$$

where $\mathcal{S}$ represents a finite set of states, $\mathcal{A}$ denotes a finite set of actions, $\mathcal{P}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ represents the state transition probability such that $\mathcal{P}(s'|s, a)$ defines the probability of transitioning to state $s' \in \mathcal{S}$ from state $s \in \mathcal{S}$ after executing action $a \in \mathcal{A}$, and $\mathcal{R}: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ signifies the reward function.

A stochastic policy $\pi_\theta: \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$, with parameters θ , defines a conditional probability distribution $\pi_\theta(a|s)$, which governs the agent's action selection in state s . As illustrated in Fig. 1, at each decision epoch t , the agent observes the current state $s_t \in \mathcal{S}$, samples an action $a_t \sim \pi_\theta(a|s_t)$, and applies it to the environment. The environment evolves to a new state $s_{t+1} \sim \mathcal{P}(\cdot | s_t, a_t)$, and the agent receives an

immediate reward $r_{t+1} = r(s_t, a_t, s_{t+1})$. This interaction generates a trajectory given by

$$\Psi = \{s_0, a_0, s_1, r_1, \dots, s_t, a_t, s_{t+1}, r_{t+1}, \dots, s_T, r_T\}, \quad (2)$$

where s_T is terminal state.

The principal objective in solving an MDP is to derive the optimal policy π^* , which maximizes the cumulative reward over time. This optimization problem is articulated by

$$\pi^*(s) = \operatorname{argmax}_{\pi} \mathbb{E}_{\pi} \{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \}, \quad (3)$$

where $\gamma \in [0,1]$ is the discount factor that determines the weight of future rewards. r_{t+k+1} is the reward

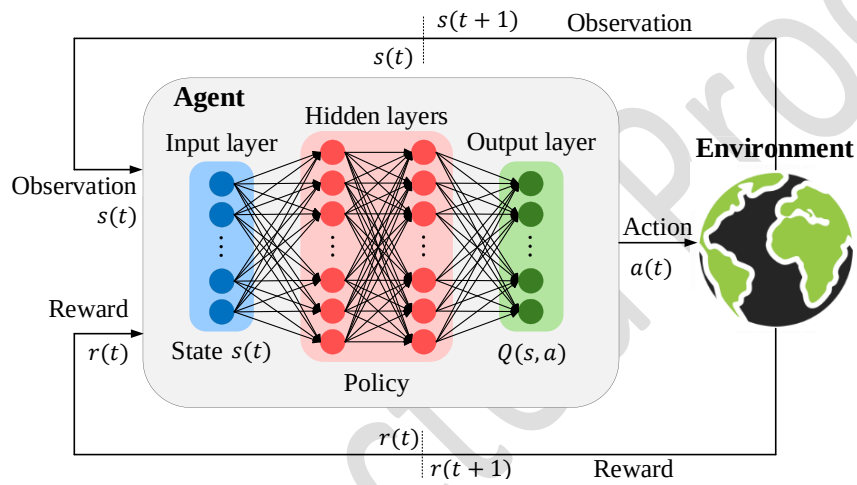


Fig. 1. Interaction between the agent and the environment in MDP for DRL.

received at time $t + k + 1$, computed from a predefined function $r(s_{t+k}, a_{t+k}, s_{t+k+1})$.

To solve the optimization problem in Eq. (3) and derive $\pi^*(s)$, the Q-value function is introduced. This function serves as a key tool in policy improvement and provides a representation of the agent's policy as

$$q_{\pi}(s, a) = \mathbb{E}_{\pi} \{ \sum_{t=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \}, \quad (4)$$

where $q_{\pi}(s, a)$ represents the return that the agent obtains in state $s \in \mathcal{S}$, taking action $a \in \mathcal{A}$, and following policy π . π^* can be derived by selecting the action that maximizes the Q-value for each state as

$$\pi^*(s) = \operatorname{argmax}_a q_*(s, a) = \operatorname{argmax}_a (\max_{\pi} q_{\pi}(s, a)), \quad (5)$$

where $q_*(s, a) = \max_{\pi} q_{\pi}(s, a)$ is the optimal Q-value function, which satisfies the Bellman optimality equation as

$$q_*(s, a) = \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s, a)} \left\{ r(s, a, s') + \gamma \max_a q_*(s', a') \mid s, a \right\} = \sum_{s', r} P(s', r|s, a) \left(r + \gamma \max_a q_*(s', a') \right). \quad (6)$$

Since the agent's knowledge may be limited, it should occasionally deviate from its current policy based on a probability distribution (ϵ -greedy) to explore alternative actions based on the following equation

$$a = \begin{cases} \text{random action} & \text{with probability } \epsilon \\ \operatorname{argmax}_a Q(s, a; \theta) & \text{with probability } 1 - \epsilon \end{cases} \quad (7)$$

allowing it to discover potentially better strategies. Formally, at each decision epoch t , the agent samples an action $a_t \sim \pi_\theta(a|s_t)$, where π_θ is the parameterized policy influenced by the exploration strategy in use. Early in training, $\epsilon \approx 1$ to promote exploration; as learning advances, ϵ decays to prioritize exploitation.

2-2- Deep Q-network algorithm

DQN approximates the optimal Q-value function in large state-action spaces, where tabular Q-learning is infeasible, by learning and optimizing $q(s, a)$ from which the policy is indirectly derived. Two neural networks are employed to approximate $q_\theta(s, a) \approx q_*(s, a)$: Online network $q_\theta(s, a)$ and target network $q_{\theta'}(s, a)$ [46].

The loss function for training online Q-network is defined as

$$\mathcal{L} = \mathbb{E}_{(s, a, r, s') \sim M} \left\{ (y - q(s, a; \theta))^2 \right\}, \quad (8)$$

where (s, a, r, s') represents a sampled transition from memory M , and θ denotes the weights of the online Q-network, which are updated during training. The target y is computed as

$$y = r + \gamma \max_a q(s', a'; \theta'), \quad (9)$$

where θ' represents the weights of the target Q-network, which are periodically synchronized with θ .

The overall framework of the algorithm is illustrated in Fig. 2.

2-3- Proximal policy optimization

As a policy-based algorithm, PPO directly learns and optimizes $\pi_\theta(a|s)$. A key feature of PPO is its constraint on policy updates, preventing large deviations from the previous policy. This constraint promotes stable and efficient learning [47].

The objective function of PPO is defined as

$$\mathcal{L}^{PPO}(\theta) = \mathbb{E}_t\{\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \mu, 1 + \mu)A_t)\}, \quad (10)$$

where $r_t(\theta)$ represents the probability ratio between the new policy $\pi_\theta(a_t|s_t)$ and the old policy $\pi_{\theta_{\text{old}}}(a_t|s_t)$, expressed as

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, \quad (11)$$

and $\text{clip}(r_t(\theta), 1 - \mu, 1 + \mu)$ restricts the policy update to a conservative range as

$$\text{clip}(r_t(\theta), 1 - \mu, 1 + \mu) = \begin{cases} 1 + \mu & \text{if } r_t(\theta) > 1 + \mu \\ r_t(\theta) & \text{if } 1 - \mu \leq r_t(\theta) \leq 1 + \mu \\ 1 - \mu & \text{if } r_t(\theta) < 1 - \mu \end{cases} \quad (12)$$

The hyperparameter $\mu > 0$ determines the clipping range, controlling the allowable magnitude of policy

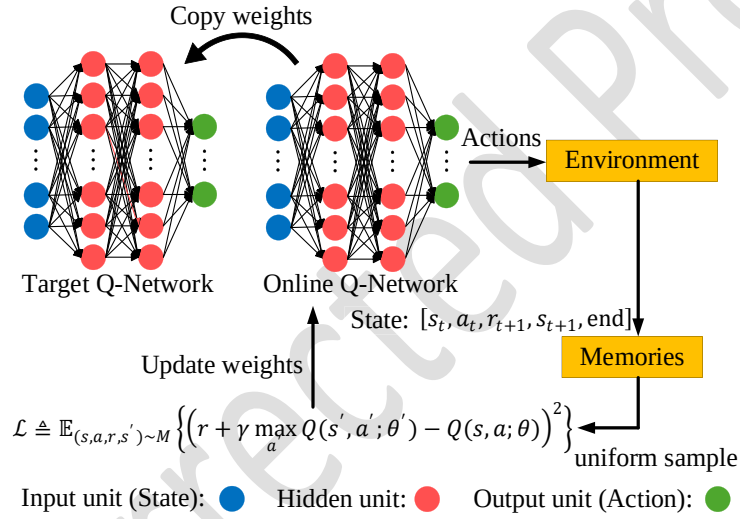


Fig. 2. General framework of the DQN algorithm.

changes. A_t denotes the advantage function, which estimates the relative benefit of taking action a_t in state s_t . The advantage function is computed as

$$A_t = q(s_t, a_t) - V(s_t). \quad (13)$$

While A_t is used to evaluate policy improvements, the primary goal remains policy optimization. The policy parameters θ are updated by optimizing the clipped objective function $\mathcal{L}^{PPO}(\theta)$ using stochastic gradient ascent based on the following update rule

$$\theta_{\text{new}} \leftarrow \theta_{\text{old}} + \alpha \nabla_{\theta} \mathcal{L}^{PPO}(\theta), \quad (14)$$

where α is the learning rate.

3. Decision Making with Risk Assessment

The idea is to assigning lower rewards to higher-risk situations to encourage safe, long-term policies. According to [20], Risk levels can be classified into three distinct categories as $\Omega = \{S, A, D\}$, where Ω is the set of risk levels with S , A , and D denoting safe, attentive, and dangerous situations, respectively. A corresponding risk value mapping is given by $\tau: \Omega \rightarrow \{0, 1, 2\}$, $\tau(S) = 0$, $\tau(A) = 1$, $\tau(D) = 2$. To calculate the conditional probability for each risk level, the following relations can be employed.

$$\begin{cases} P(d|\tau = D) = \begin{cases} 1, & d < d_D \\ \exp(-\frac{\Delta d_D^2}{2\sigma^2}), & \text{otherwise} \end{cases} \\ P(d|\tau = A) = \exp(-\frac{\Delta d_A^2}{2\sigma^2}) \\ P(d|\tau = S) = \begin{cases} 1, & d > d_S \\ \exp(-\frac{\Delta d_S^2}{2\sigma^2}), & \text{otherwise} \end{cases} \end{cases} \quad (15)$$

Here, d represents the agent's relative distance to other vehicles, and the hyperparameters d_D , d_A , d_S , and σ are predefined safety thresholds that influence the smoothness of the risk curves. The term Δd_i is defined as

$$\Delta d_i = |d - d_i|, i \in \Omega. \quad (16)$$

Using Bayesian theorem, the probability of each risk level in a given state can be calculated according to the following relationship

$$P(\tau|d) = \frac{P(d|\tau)P(\tau)}{\sum_{\tau \in \Omega} P(\tau)P(d|\tau)}, \quad (17)$$

where $P(\tau|d)$ represents the probability of risk level d given state d , while $P(d|\tau)$ is the conditional probability from Eq. (15). $P(\tau)$ denotes the prior probability, assumed to be uniform across risk levels, ensuring $\sum_{\tau \in \Omega} P(\tau) = 1$.

Fig. 3 illustrates the risk curves, showing the conditional probability of each state across different risk levels in Fig. 3(a), and the risk level associated with each state in Fig. 3(b).

Since the outcomes of Eq. (17) are not directly applicable to the DRL framework, a risk metric ξ representing the expected assessed risk is introduced as follows.

$$\xi = \mathbb{E}\{\tau\} = \sum_{\tau \in \Omega} \tau P(\tau|d) = \sum_{\tau \in \{2,1\}} \tau P(\tau|d) \quad (18)$$

The optimal strategy minimizing risk is given by

$$\pi^*(s) = \underset{\pi}{\operatorname{argmin}} \mathbb{E}_{\pi} \{ \sum_{k=0}^{\infty} \gamma^k \xi_{t+k+1} \mid s_t = s \}. \quad (19)$$

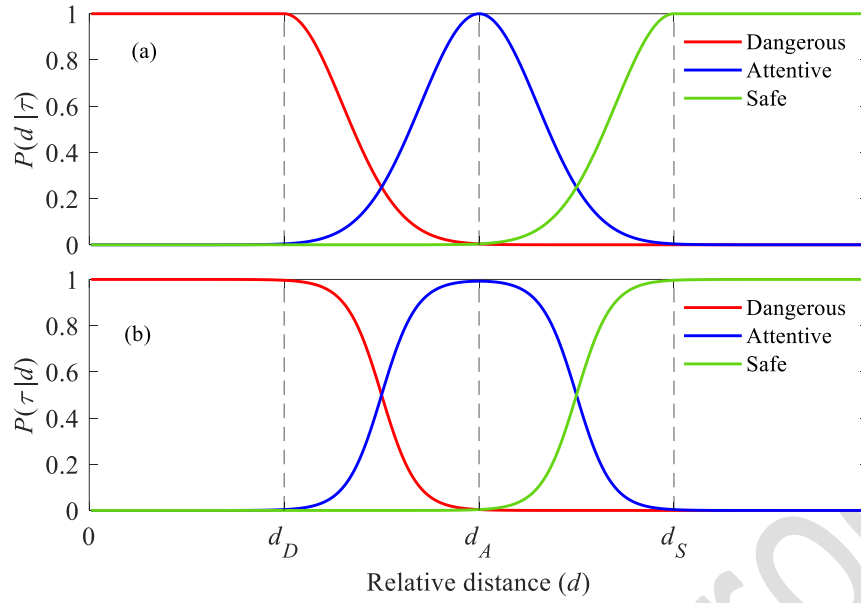


Fig. 3. Risk curves for a) $P(d|\tau)$ and b) $P(\tau|d)$.

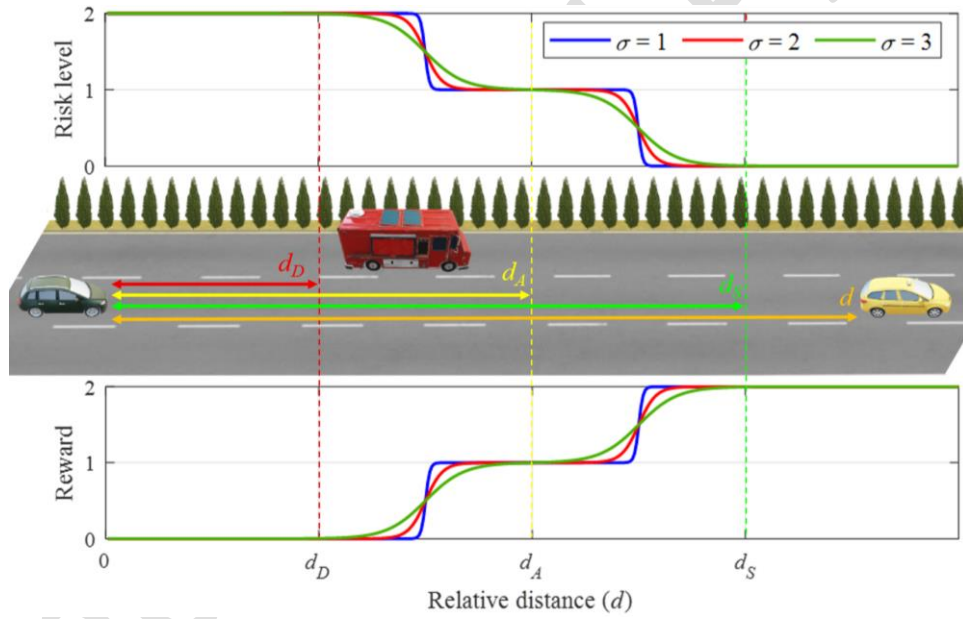


Fig. 4. Overall framework for risk assessment.

Eq. (19) parallels Eq. (3) but aims to minimize the target value. A new risk metric is defined as

$$r_{\text{risk}_{t+k+1}} = \max \xi - \xi_{t+k+1}, \quad (20)$$

where $\max \xi$ represents the maximum risk value, ensuring that $r_{\text{risk}_{t+k+1}}$ remains within $[0, 2]$. Eq.

(19) can now be rewritten as

$$\pi^*(s) = \operatorname{argmax}_{\pi} \mathbb{E}_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{\text{risk}_{t+k+1}} | s_t = s \right\}, \quad (21)$$

Comparing Eq. (21) and Eq. (3) shows that DRL can identify a strategy that maximizes reward or minimizes risk. The corresponding Q-value for this case is given by

$$q_{\pi}(s, a) = \sum_{k=0}^{\infty} \gamma^k (\max \xi - \xi_{t+k+1}) | s_t = s, a_t = a] = \mathbb{E}_{\pi} [\sum_{k=0}^{\infty} \gamma^k r_{\text{risk}_{t+k+1}} | s_t = s, a_t = a]. \quad (22)$$

Eq. (22) is now fully analogous to Eq. (4). The RA framework for the problem is depicted in Fig. 4.

4. Collision Avoidance Decision Making

The simulation environment models dynamic, stochastic behaviors like speed variations and lane changes, creating realistic challenges to assess the agent's adaptability and decision-making. This approach narrows the gap between simulation and real-world driving, supporting the development of robust AD systems.

4-1- State space

To simplify the training process and enhance adaptability, a numerical representation is employed, structured to encapsulate both ego-vehicle and the dynamic states of surrounding vehicles, formulated as

$$s_t \triangleq \left\{ \ell, \left\{ x^{(i)}, y^{(i)}, v_x^{(i)}, v_y^{(i)}, l^{(i)} \right\}_{i=1}^N \right\}, \quad (23)$$

where N denotes the number of nearest neighboring vehicles considered in the state space, $l, l^{(i)} \in \{0,1,2\}$ represent the current lane indices of the ego and the i -th surrounding vehicle, respectively. $x^{(i)}, y^{(i)} \in \mathbb{R}$ represent the relative longitudinal and lateral distances between the host and i -th surrounding vehicle, while $v_x^{(i)}, v_y^{(i)} \in \mathbb{R}$ denote their relative velocities. The state space incorporates observations from the five nearest vehicles in the environment. For each neighboring vehicle $i \in \{1, \dots, N\}$, the ranges of values for $x^{(i)}, y^{(i)}, v_x^{(i)}$ and $v_y^{(i)}$ are defined as $x^{(i)}, y^{(i)} \in [-100, 100]$ and $v_x^{(i)}, v_y^{(i)} \in [-20, 20]$. To promote numerical stability and efficient training, a feature-wise normalization is applied, mapping each continuous variable to the interval $[-1, 1]$. This is achieved via an element-wise transformation

$$\bar{f}^{(i)} = \frac{f^{(i)}}{D_f}, \text{ with } D_f \in \{100, 100, 20, 20\}. \quad (24)$$

4-2- Action space

The action space should balance simplicity and effectiveness, remaining rich enough for robust decision-making while minimal for efficient learning [48]. By delegating lane-change execution to lower-level actions, the focus goes to longitudinal actions as well, resulting in a streamlined and purposeful action space defined as

$$a_t \triangleq \{L, R, O, A^+, A^-\}, \quad (25)$$

where L and R denote lane changes to the left and right, respectively. O denotes holding the current lane and speed. A^+ and A^- refer to longitudinal actions corresponding to acceleration and deceleration, respectively. In the defined action space, certain actions may be restricted in marginal states. To prevent unsafe or undefined transitions and ensuring continuity in the control process and maintaining environmental stability, invalid actions are automatically remapped to the action O as

$$a_t = \begin{cases} O, & a_t \notin \mathcal{A}_{\text{valid}} \\ a_t, & \text{otherwise} \end{cases}, \quad (26)$$

where $\mathcal{A}_{\text{valid}}(s_t) \subseteq \mathcal{A}$ denotes the set of admissible actions at state s_t . Table 2 summarizes the defined action space and its constraints. $v \in [v_{\min}, v_{\max}]$ denotes the current speed of the vehicle, while $v_{\max}$ and $v_{\min}$ represent the maximum and minimum allowed speeds, respectively.

Table 2. Action space and its constraints.

Action	Description	Invalid when
L	Lane change to the left	$l = 0$
R	Lane change to the right	$l = 2$
O	Maintain current state	-
A^+	Increase speed	$v = v_{\max}$
A^-	Decrease speed	$v = v_{\min}$

4-3- Reward function

At each step t , the agent receives a reward r_t defined as the sum of various components designed to encourage safe and efficient driving behavior.

A fixed penalty is applied upon collision with another vehicle as

$$r_{\text{collision}} = \begin{cases} -5, & \text{collision occurs} \\ 0, & \text{otherwise} \end{cases}. \quad (27)$$

To incentivize staying collision-free, a small constant reward is given at each non-terminal step as follows to encourage the agent to survive in the environment.

$$r_{\text{exist}} = \begin{cases} 0.1, & \text{no collision} \\ 0, & \text{otherwise} \end{cases} \quad (28)$$

The speed limit is defined as $15 \leq v_t \leq 25$ for the agent. To encourage higher speeds within the allowable range, the agent is rewarded based on a linear scaling function as

$$r_{\text{speed}} = \begin{cases} 0.04(v_t - 15), & 15 \leq v_t \leq 25 \\ 0, & \text{otherwise} \end{cases}. \quad (29)$$

This yields a maximum speed reward of $r_{\text{speed}_{\text{max}}} = 0.4$ when $v_t = 25$. This mechanism results in more frequent lane changes, providing a better assessment of the agent's decision-making capabilities. Finally, the risk assessment reward is added to the aforementioned rewards. The reward functions, without and with risk assessment, are defined in the following equations, respectively.

$$r = r_{\text{exist}} + r_{\text{speed}} + r_{\text{collision}} \quad (30)$$

$$r = r_{\text{exist}} + r_{\text{risk}} + r_{\text{speed}} + r_{\text{collision}} \quad (31)$$

The proposed reward function is deliberately safety-prioritized rather than equally weighted. Although unit multipliers are applied to its components, their numerical ranges differ substantially. At each non-collision step, the risk-assessment reward lies within $[0, 2]$, whereas the speed reward is limited to $[0, 0.4]$ and the existence reward is fixed at 0.1. Consequently, the maximum risk-assessment contribution is five times the maximum speed contribution. Therefore, the risk-assessment term constitutes the dominant component of the dense reward signal, while the speed and existence terms provide secondary incentives for efficient motion and continued collision-free operation.

4-4- Training of The Agent

The experiments were conducted using HighwayEnv 1.12.1 and Gymnasium 1.3.0. The ego vehicle was modeled as an MDPVehicle and controlled through the five discrete high-level actions defined in Section 4.2. A proportional low-level controller translated the selected target lane and speed into steering and acceleration commands. Surrounding vehicles were modeled as IDMVehicle instances. Their longitudinal behavior was governed by the IDM, whereas lane-changing decisions were determined using the minimizing overall braking induced by lane changes (MOBIL) model. Accordingly, a lane change was executed only when it provided a sufficient acceleration benefit without requiring unsafe braking by the new following vehicle. At the beginning of each episode, vehicle lanes

and traffic-related parameters were sampled using the environment's seeded random-number generator. This procedure generated distinct yet reproducible traffic configurations across different random seeds. The observation included the ego vehicle and its five nearest surrounding vehicles, including vehicles

Table 3. Simulation-environment configuration.

Parameter	Configuration
Simulator	HighwayEnv 1.12.1, Gymnasium 1.3.0
Road geometry	Straight highway, three lanes
Road speed limit	25 m/s
Controlled vehicles	One ego vehicle
Surrounding vehicles	50 IDMVehicle instances
Training traffic-density coefficient	1.0
Approximate initial density	42 vehicles/km across all lanes; approximately 14 vehicles/km/lane
Vehicle geometry	Length 5 m; width 2 m
Vehicle dynamics	Kinematic bicycle model
Ego-vehicle model	MDPVehicle with low-level speed and lane tracking
Ego initial speed	20 m/s
Ego target speeds	[15, 25] m/s
Surrounding longitudinal behavior	Intelligent Driver Model
Surrounding lateral behavior	MOBIL lane-change model
Surrounding initial speed	Uniform distribution, $U(17.5, 20)$ m/s
Initial lane	Randomly selected from the three lanes
Position-spacing multiplier	$U(0.9, 1.1)$
IDM acceleration exponent	$U(3.5, 4.5)$
Desired IDM time gap	1.5 s
Desired standstill distance	10 m
Simulation frequency	15 Hz
Agent decision frequency	1 Hz
Observed neighboring vehicles	Five nearest vehicles
Episode time limit	60 s

located behind it. An episode terminated when the ego vehicle reached the end of the 1000 – m road segment, collided with another vehicle, or exceeded the 60 – s time limit. The complete simulation-environment configuration is summarized in Table 3.

All methods were implemented using Stable-Baselines3 2.9.0 and PyTorch. To account for stochasticity arising from neural-network initialization, action exploration, replay-buffer sampling, policy sampling, and traffic generation, each method was independently trained under a common multi-seed evaluation protocol. This procedure resulted in 25 independent training runs across DQN, risk-aware DQN (SDQN), PPO, risk-aware PPO (SPPO), and realism-constrained SDQN (RSDQN), which limits the

agent's speed to that of surrounding vehicles.

All methods used the same hidden-layer dimensions and ReLU activation functions to support a

Table 4. Training protocol and algorithm-specific hyperparameters.

Parameter	DQN, SDQN, and RSDQN	PPO and SPPO
Independent training runs	5	5
Training seeds	11, 22, 33, 44, 55	11, 22, 33, 44, 55
Environment steps per run	120,000	120,000
Hidden-layer architecture	[32, 128, 256, 128, 32]	[32, 128, 256, 128, 32] for policy and value networks
Hidden activation	ReLU	ReLU
Learning rate	1×10^{-4}	3×10^{-4}
Discount factor (γ)	0.99	0.99
Batch size	128	64
Optimizer	Adam	Adam
Replay-buffer capacity	100,000	N/A
Learning begins	After 5,000 steps	After the first 2,048-step rollout
Training frequency	Every 4 environment steps	After each 2048-step rollout
Gradient updates	1 per training event	10 epochs per rollout
Target-network update	Every 2,000 steps	N/A
Initial exploration probability	1.0	Policy sampling
Final exploration probability	0.10	Policy sampling
PPO rollout length	N/A	2,048 steps
PPO clipping range	N/A	0.20
d_D	2	2
d_A	4	4
d_S	6	6
σ	3	3

consistent architectural comparison. For the DQN-based methods, the network approximated the action-value function. For the PPO-based methods, separate policy and value networks were constructed using the same hidden-layer dimensions. The complete training protocol and algorithm-specific hyperparameters are summarized in Table 4. A fixed training budget ensured equal environment interactions across all methods; convergence was therefore assessed retrospectively rather than used as a stopping criterion.

Experiments were conducted on a 12th Gen Intel® Core™ i9-12900H Processor, 8GB of DDR5 RAM, and an NVIDIA® GeForce RTX™ 3070 Ti Laptop GPU with 8 GB of GDDR6 memory. The wall-clock training time for 120,000 environment steps was measured over five independent runs. The training performance of all methods is presented in Fig. 5, while Fig. 6 reports the mean training time for each method, with error bars representing the sample standard deviation across five independent runs. Additionally, the sensitivity of DQN and PPO to key hyperparameters is analyzed in Fig. 7.

Detailed step-by-step implementation of the proposed methodologies for DQN and PPO are outlined in

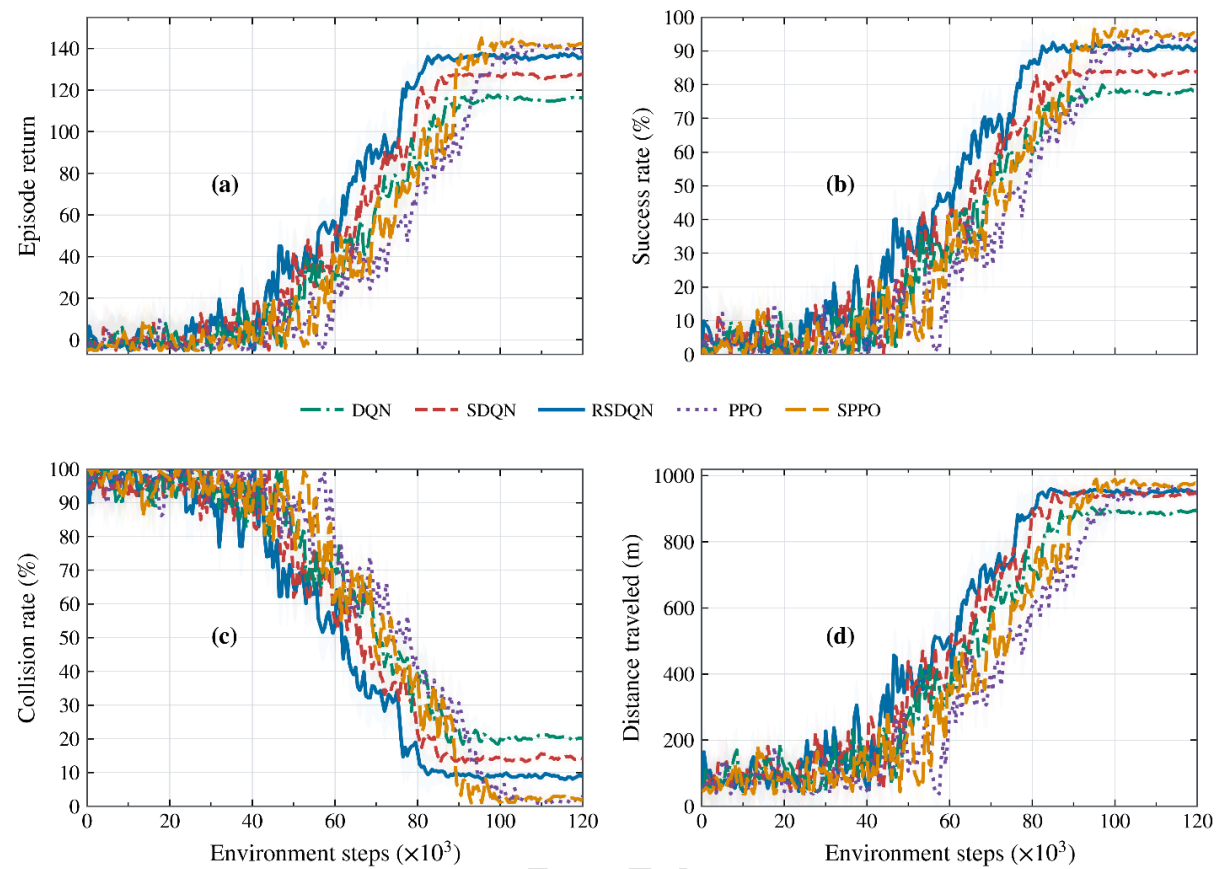


Fig. 5. Training performance: a) episode return, b) success rate, c) collision rate, and d) distance traveled.

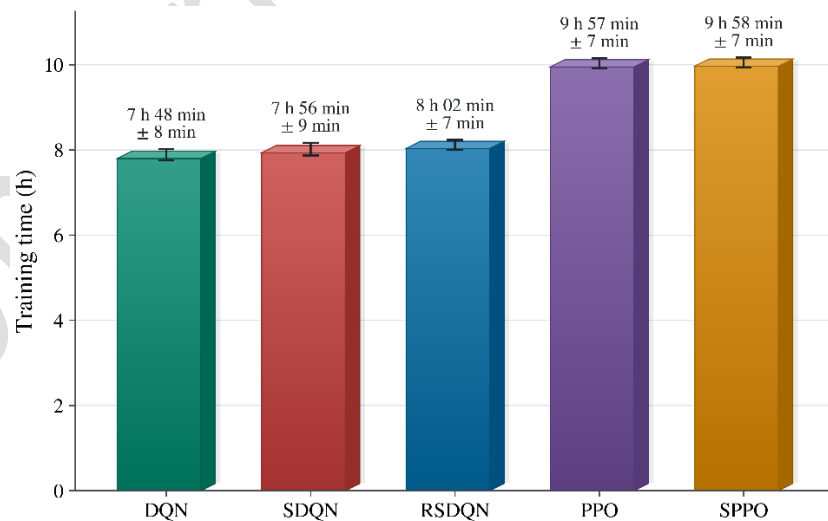


Fig. 6. Mean training time with sample standard deviation across five runs.

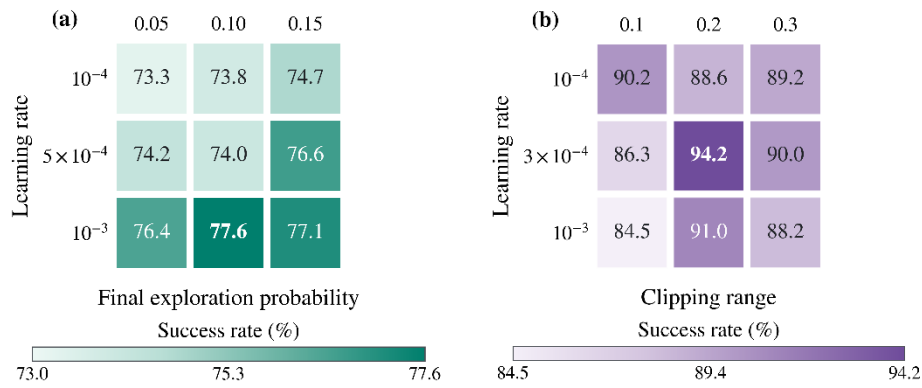


Fig. 7. Hyperparameter sensitivity analysis for a) DQN and b) PPO.

Algorithm 1 and Algorithm 2, respectively. The trained agent from the learning phase is directly utilized in the evaluation phase to assess its performance and efficiency. To ensure a fair comparison, both methods are trained and evaluated under identical conditions. This consistent framework guarantees that the obtained results are reliable and compelling [20].

Algorithm 1. Training the DQN agent for highway driving

Input: Number of episodes M , maximum timesteps per episode T , replay buffer capacity N

Output: Learned Q-network $q(s, a; \theta)$ representing optimal policy $\pi^*(s)$

- 1: Initialize online Q-network $q(s, a; \theta)$ with random weights θ
 - 2: Initialize target Q-network $q'(s, a; \theta')$ with $\theta' \leftarrow \theta$
 - 3: Initialize replay buffer D with capacity N
 - 4: **for** episode $e = 1$ to M **do**
 - 5: Initialize environment and observe initial state s_0
 - 6: **for** $t = 1$ to T **do**
 - 7: With probability ε select a random action a_t
 - 8: Otherwise select $a_t = \underset{a}{\operatorname{argmax}} q_\pi(s, a; \theta)$
 - 9: Execute a_t , observe r_{t+1} and next state s_{t+1}
 - 10: Store transition $(s_t, a_t, r_{t+1}, s_{t+1})$ in D
 - 11: Sample a mini-batch of transitions (s_j, a_j, r_j, s'_j) from D
 - 12: Compute target $y_j = r_j + \gamma \max_{a'} q(s'_j, a'; \theta')$
 - 13: Update Q-network by minimizing loss $L(\theta) = (y_j - q(s_j, a_j; \theta))^2$
 - 14: Every C steps, update target network $\theta' \leftarrow \theta$
 - 15: **end for**
 - 16: **end for**
 - 17: Return trained Q-network $q(s, a; \theta)$
-

Algorithm 2. Training the PPO agent for highway driving

Input: Number of episodes M , maximum timesteps per episode T , learning rate α , clipping range μ

Output: Trained policy network $\pi(s; \theta)$

- 1: Initialize policy network $\pi(s; \theta)$ and value network $V(s; \theta)$ with random parameters
 - 2: Initialize environment E
 - 3: **for** episode $e = 1$ to M **do**
 - 4: Initialize state s_0
-

```

5:   Initialize memory buffer  $B \leftarrow \varphi$ 
6:   for  $t = 1$  to  $T$  do
7:     Sample action  $a_t \sim \pi(\cdot | s_t; \theta)$ 
8:     Execute  $a_t$ , observe  $r_{t+1}$  and next state  $s_{t+1}$ 
9:     Store transition  $(s_t, a_t, r_{t+1}, s_{t+1})$  in  $B$ 
10:     $s_t \leftarrow s_{t+1}$ 
11:  end for
12:  Compute advantage estimates  $A_t$  with  $V(s_t; \varphi)$ 
13:  Compute  $R_t$  and old action probabilities  $\pi_{\theta_{\text{old}}}(a_t | s_t)$ 
14:  for  $k$  epochs do
15:    for each mini-batch from  $B$  do
16:      Compute probability ratio  $r_t(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)}$ 
17:      Compute clipped surrogate objective  $\mathcal{L}^{ppo}(\theta) = \mathbb{E}_t\{\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \mu, 1 + \mu)A_t)\}$ 
18:      Update  $\theta$  using gradient ascent  $\theta \leftarrow \theta + \alpha \nabla_{\theta} \mathcal{L}^{ppo}(\theta)$ 
19:      Update  $\varphi$  by minimizing value loss  $L_{V(\varphi)} = (V(s_t; \varphi) - R_t)^2$ 
20:    end for
21:  end for
22: end for
23: Return trained policy  $\pi(s; \theta)$ 

```

5. Simulation Results

After training, each saved policy was evaluated deterministically for 500 episodes. Evaluation seeds 1,000,000 through 1,000,499 were used for every trained model, ensuring that competing methods encountered the same initial lanes, vehicle positions, traffic speeds, and randomized IDM parameters. The evaluation environment, traffic density, vehicle dynamics, and outcome definitions were identical for all methods. No exploratory actions or parameter updates were permitted during evaluation. Performance was evaluated in terms of success rate, average distance traveled per episode, and average number of vehicles safely passed per episode. An episode was considered successful if the ego vehicle reached the end of the 1000 m road segment without collision within the 60 s time limit; otherwise, it

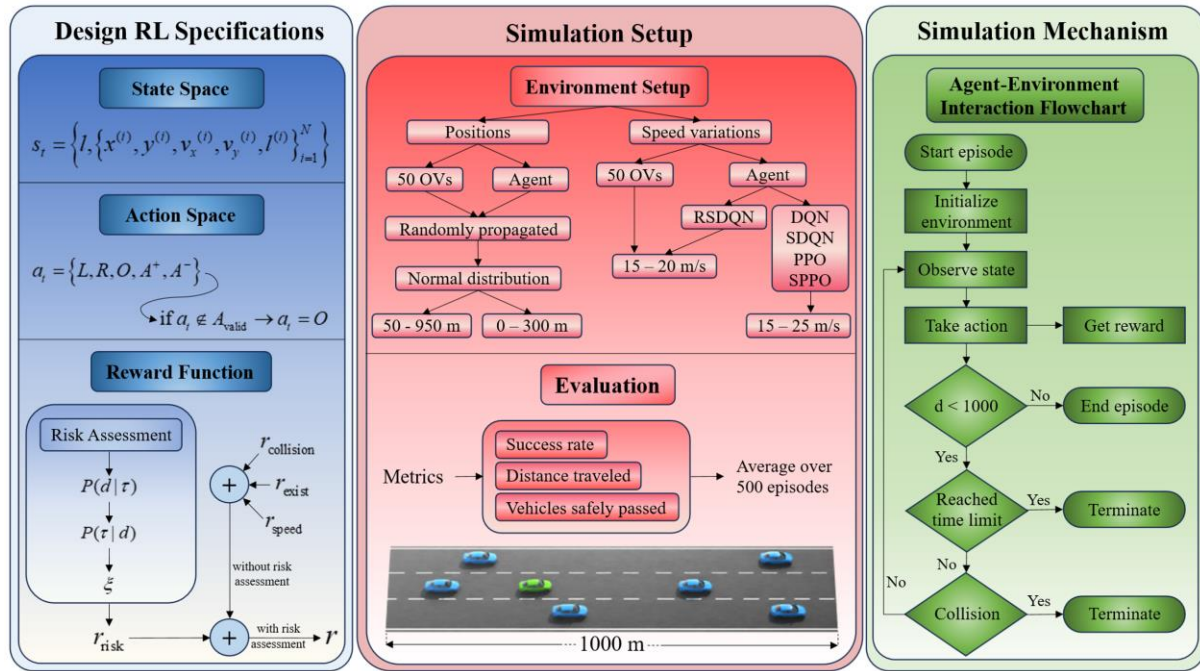


Fig. 8. Schematic of the design, simulation setup and interaction mechanism.

was considered unsuccessful. The number of vehicles safely passed was defined as the number of unique surrounding vehicles overtaken by the ego vehicle during an episode. Each vehicle could be counted only once. A vehicle was classified as safely passed if it was initially ahead of the ego vehicle and subsequently remained behind it for at least two consecutive decision steps. An overview of the simulation setup is presented in Fig. 8.

The evaluation results for all methods under different traffic densities are presented in Fig. 9. Analysis

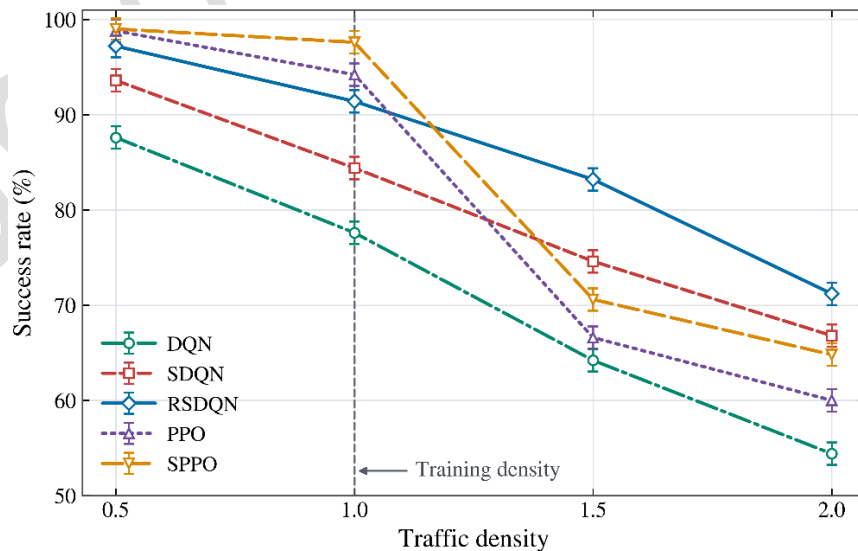


Fig. 9. Success rates under unseen traffic densities with 95% confidence intervals.

of the data related to the DQN and PPO agents reveals distinct performance characteristics. DQN-based agents exhibit more aggressive behavior. PPO agents adopt a more conservative approach, but they achieve a higher overall success rate.

DQN performs a relatively high number of overtakes. The aggressive overtaking behavior is advantageous for maintaining high speeds and facilitating rapid transitions between lanes. However, this aggressive strategy may lead to high-risk maneuvers. r_{speed} also contributes to this aggressive performance.

The findings underscore the need for risk assessment to ensure more reliable performance. The incorporation of risk assessment to DQN has led to a reduction from 10.4 to 4.7 in the average number of vehicles safely passed. In effect, the agent has learned that certain overtaking maneuvers are risky and has adopted a more conservative behavior. The number of overtakes for SPPO remains low, showing a slight and negligible decrease from 0.33 to 0.29 compared to PPO. This indicates that SPPO operates even a little bit more conservatively than PPO. Number of vehicles safely passed in RSDQN is even less than SDQN (1.55) and this is reasonable because the agents speed is mapped to the speed zone of other vehicles. Incorporating risk assessment improved DQN from 77.6% to 84.4% (SDQN), a 6.8% gain. SPPO improves PPO's performance by 3.4%, from 94.2% to 97.6% completion. RSDQN successfully reaches the end of the route in 91.4% of cases, 7% increase compared to SDQN. The enhanced performance of RSDQN highlights its effectiveness in balancing efficiency and safety. Quantitative results, including performance metrics for each method, are presented in Table 5. Most of the collisions occurred were rear-end collisions, where the front vehicle was hit by the agent from behind. This may be attributed to a common tendency to overlook longitudinal actions, leading to a reliance on for example IDM system for these considerations. An example of such a collision is depicted in Fig. 10.

Table 5. Quantitative results of proposed methods.

Method	Distance traveled	Vehicles safely passed	Success rate (%)
DQN	887.22	10.4	77.6
PPO	974.91	0.33	94.2

SDQN	945.15	4.7	84.4
SPPO	993.69	0.29	97.6
RSDQN	956.07	1.55	91.4

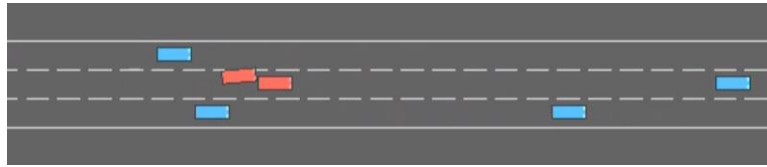


Fig. 10. An example of rear-end collisions.

Although randomized initial conditions and behavioral parameters increase traffic diversity, surrounding vehicles still follow structured IDM/MOBIL models that cannot fully capture unpredictable human behavior, such as hesitation, aggressive gap acceptance, abrupt braking, or unexpected lane changes. The simulator also assumes accurate kinematic observations and simplified vehicle dynamics, without explicitly modeling sensor noise, occlusion, perception and control delays, actuator limitations, road-friction variations, adverse weather, or heterogeneous mixed traffic. Furthermore, the experiments are restricted to a straight three-lane highway, and the Bayesian risk module does not explicitly account for uncertainty in perception, driver intentions, or future multi-agent trajectories. Therefore, the reported results demonstrate comparative performance under the specified decision-level simulation conditions rather than real-world deployment readiness. Future work will include broader domain randomization, evaluation under unseen traffic and behavioral conditions, naturalistic-data calibration, mixed human–autonomous traffic, cross-simulator validation, higher-fidelity vehicle dynamics, hardware- and vehicle-in-the-loop experiments, and controlled proving-ground testing.

6. Conclusion

This study proposed a risk-aware DRL framework for safe and efficient decision-making in AD on highways. By integrating Bayesian risk assessment with DQN and PPO, the framework enhances decision-making by accounting for uncertainties in surrounding vehicle behaviors, improving both safety and efficiency. Simulation results demonstrated that incorporating risk assessment significantly improves safety. While DQN exhibited frequent overtaking, it also performed riskier maneuvers. PPO and SPPO prioritized stability and safety but at the cost of fewer overtaking opportunities. The proposed

framework contributes to advancing DRL-based autonomous driving strategies. Future work can extend this framework to more complex traffic environments, incorporating vehicle-to-vehicle communication. Our findings indicate that rear-end collisions, where the front vehicle is struck from behind, were predominant. This may be attributed to a common tendency to overlook longitudinal actions, leading to a reliance on for example IDM system for these considerations. One potential approach is to integrate classic control-based and data-driven methodologies, leveraging their complementary strengths. Specifically, model predictive control can be employed for longitudinal actions, ensuring optimal speed and acceleration control, while DRL can be utilized for lane-change decision making and lateral maneuvers.

References

- [1] S. Adelipour, E.A.D. Razgahi, M. Haeri, Vulnerability mitigation of urban traffic control against cyberattacks using secure multi-party computation, *IEEE Transactions on Intelligent Transportation Systems*, (2025).
- [2] D.S. Sarwatt, Y. Lin, J. Ding, Y. Sun, H. Ning, Metaverse for intelligent transportation systems (ITS): A comprehensive review of technologies, applications, implications, challenges and future directions, *IEEE Transactions on Intelligent Transportation Systems*, 25(7) (2024) 6290-6308.
- [3] B.B. ELallid, N. Benamar, M. Bagaa, N. Mrani, Enhancing autonomous vehicle control in complex scenarios with deep transformer reinforcement learning, *Engineering Applications of Artificial Intelligence*, 158 (2025) 111483.
- [4] J. Gao, B. Yu, Y. Chen, S. Bao, K. Gao, L. Zhang, An ADAS with better driver satisfaction under rear-end near-crash scenarios: A spatio-temporal graph transformer-based prediction framework of evasive behavior and collision risk, *Transportation research part C: emerging technologies*, 159 (2024) 104491.
- [5] J. Wu, C. Huang, H. Huang, C. Lv, Y. Wang, F.-Y. Wang, Recent advances in reinforcement learning-based autonomous driving behavior planning: A survey, *Transportation Research Part C: Emerging Technologies*, 164 (2024) 104654.
- [6] S.S. Hosseini Najafabadi, M. Haeri, Traffic responsive urban predictive control for signal splitting

- in large-scale congested urban road networks, *Journal of Intelligent Transportation Systems*, (2025) 1-15.
- [7] Y. Xing, Y. Wu, H. Wang, L. Wang, L. Li, Y. Peng, Failed lane-changing detection and prediction using naturalistic vehicle trajectories, *Transportation Research Part C: Emerging Technologies*, 170 (2025) 104939.
- [8] O. Sharma, N. Sahoo, N. Puan, Kernelized convolutional and transformer based hierarchical spatio-temporal attention network for autonomous vehicle trajectory prediction, *International Journal of Intelligent Transportation Systems Research*, 23(2) (2025) 676-701.
- [9] S.U. Ratre, B. Joshi, Trajectory and motion prediction of autonomous vehicles driving assistant system using distributed discriminator based bi-directional long short term memory model, *International Journal of Intelligent Transportation Systems Research*, 23(1) (2025) 245-258.
- [10] Y.-Z. Lei, Y. Cheng, X.T. Yang, An optimization-free approximation framework for connected and automated vehicles eco-trajectory planning under limited computing capacity, *Transportation Research Part C: Emerging Technologies*, 171 (2025) 104949.
- [11] T. Zhang, Z. Wang, H. Wang, J. Li, Intelligent defensive driving for autonomous vehicles: Framework, strategy and verification, *Accident Analysis & Prevention*, 226 (2026) 108355.
- [12] M. Soleimani, A.A. Saria, Towards Autonomy: A Comprehensive Technical and Ethical Review of Automated Vehicle Safety, *Multimodal Transportation*, (2025) 100272.
- [13] Y. Du, J. Chen, C. Zhao, C. Liu, F. Liao, C.-Y. Chan, Comfortable and energy-efficient speed control of autonomous vehicles on rough pavements using deep reinforcement learning, *Transportation Research Part C: Emerging Technologies*, 134 (2022) 103489.
- [14] X. Zhao, Y. Fang, H. Min, X. Wu, W. Wang, R. Teixeira, Potential sources of sensor data anomalies for autonomous vehicles: An overview from road vehicle safety perspective, *Expert Systems with Applications*, 236 (2024) 121358.
- [15] K. Gao, X. Li, B. Chen, L. Hu, J. Liu, R. Du, Y. Li, Dual transformer-based prediction for lane change intentions and trajectories in mixed traffic environment, *IEEE Transactions on Intelligent Transportation Systems*, 24(6) (2023) 6203-6216.

- [16] Y. Wang, Z. Han, Y. Xing, S. Xu, J. Wang, A survey on datasets for the decision making of autonomous vehicles, *IEEE Intelligent Transportation Systems Magazine*, 16(2) (2024) 23-40.
- [17] T. Wang, M. Ma, S. Liang, J. Yang, Y. Wang, Robust lane change decision for autonomous vehicles in mixed traffic: A safety-aware multi-agent adversarial reinforcement learning approach, *Transportation Research Part C: Emerging Technologies*, 172 (2025) 105005.
- [18] H. Lu, J. Yang, M. Zhu, C. Lu, X. Chen, X. Zheng, H. Yang, A knowledge-driven, generalizable decision-making framework for autonomous driving via cognitive representation alignment, *Transportation Research Part C: Emerging Technologies*, 172 (2025) 105030.
- [19] P. Lamei, M. Haeri, Lane Change Decision Making Using Deep Reinforcement Learning, in: 2025 33rd International Conference on Electrical Engineering (ICEE), IEEE, 2025, pp. 357-362.
- [20] G. Li, Y. Yang, S. Li, X. Qu, N. Lyu, S.E. Li, Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness, *Transportation research part C: emerging technologies*, 134 (2022) 103452.
- [21] M. Ataollahi, M. Farrokhi, A Bibliographic and Qualitative Analysis on Navigation Concepts of Mobile Robots, *AUT Journal of Electrical Engineering*, 58(1) (2026) 3-30.
- [22] G. Li, Y. Yang, T. Zhang, X. Qu, D. Cao, B. Cheng, K. Li, Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios, *Transportation research part C: emerging technologies*, 122 (2021) 102820.
- [23] K. Wang, C. Shen, X. Li, J. Lu, Uncertainty quantification for safe and reliable autonomous vehicles: A review of methods and applications, *IEEE Transactions on Intelligent Transportation Systems*, (2025).
- [24] S. Noh, Decision-making framework for autonomous driving at road intersections: Safeguarding against collision, overly conservative behavior, and violation vehicles, *IEEE Transactions on Industrial Electronics*, 66(4) (2018) 3275-3286.
- [25] D. Shin, B. Kim, K. Yi, A. Carvalho, F. Borrelli, Human-centered risk assessment of an automated vehicle using vehicular wireless communication, *IEEE Transactions on Intelligent Transportation Systems*, 20(2) (2018) 667-681.

- [26] G. Li, Y. Chen, D. Cao, X. Qu, B. Cheng, K. Li, Extraction of descriptive driving patterns from driving data using unsupervised algorithms, *Mechanical Systems and Signal Processing*, 156 (2021) 107589.
- [27] Y.S. Son, W. Kim, Cooperation-based risk assessment prediction for rear-end collision avoidance in autonomous lane change maneuvers, in: *Actuators*, MDPI, 2022, pp. 98.
- [28] F. Yixuan, Z. Huanhuan, Y. Minjie, W. Hongchao, Risk Assessment Model of Emergency Lane Change Behavior for Intelligent Vehicles, *Recent Patents on Mechanical Engineering*, 17(3) (2024) 208-221.
- [29] A. Sivaprakasam, S. Yogarayan, J.N. Mogan, S.F. Abdul Razak, A. Azman, K. Raman, Driver Drowsiness and Alcohol Detection for Automotive Safety Systems, *Civil Engineering Journal*, 11(7) (2025) 2686-2700.
- [30] A. Almomany, Z. Marouf, A. Jarrah, M. Sutcu, Real-Time FPGA-Based ADAS Solution for Driver Drowsiness Detection and Autonomous Stopping, *Emerging Science Journal*, 9(5) (2025) 2687-2703.
- [31] A. Irshayyid, J. Chen, G. Xiong, A review on reinforcement learning-based highway autonomous vehicle control, *Green Energy and Intelligent Transportation*, 3(4) (2024) 100156.
- [32] J. Duan, Y. Guan, S.E. Li, Y. Ren, Q. Sun, B. Cheng, Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors, *IEEE transactions on neural networks and learning systems*, 33(11) (2021) 6584-6598.
- [33] P. Wang, C.-Y. Chan, A. de La Fortelle, A reinforcement learning based approach for automated lane change maneuvers, in: *2018 IEEE intelligent vehicles symposium (IV)*, IEEE, 2018, pp. 1379-1384.
- [34] F. Ye, X. Cheng, P. Wang, C.-Y. Chan, J. Zhang, Automated lane change strategy using proximal policy optimization-based deep reinforcement learning, in: *2020 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2020, pp. 1746-1752.
- [35] H. Gao, M. Zhao, X. Zheng, C. Wang, L. Zhou, Y. Wang, L. Ma, B. Cheng, Z. Wu, Y. Li, An improved hierarchical deep reinforcement learning algorithm for multi-intelligent vehicle lane

- change, *Neurocomputing*, 609 (2024) 128482.
- [36] J. Duan, S. Eben Li, Y. Guan, Q. Sun, B. Cheng, Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data, *IET Intelligent Transport Systems*, 14(5) (2020) 297-305.
- [37] S. Wu, D. Tian, X. Duan, J. Zhou, D. Zhao, D. Cao, Continuous decision-making in lane changing and overtaking maneuvers for unmanned vehicles: A risk-aware reinforcement learning approach with task decomposition, *IEEE Transactions on Intelligent Vehicles*, 9(4) (2024) 4657-4674.
- [38] K. Mokhtari, A.R. Wagner, Don't get yourself into trouble! risk-aware decision-making for autonomous vehicles, *arXiv preprint arXiv:2106.04625*, (2021).
- [39] P. Wang, H. Li, C.-Y. Chan, Continuous control for automated lane change behavior based on deep deterministic policy gradient algorithm, in: *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 1454-1460.
- [40] J. Wang, C. Hu, J. Zhao, S. Zhang, Y. Han, Deep Q-network-based efficient driving strategy for mixed traffic flow with connected and autonomous vehicles on urban expressways, *Transportation research record*, 2677(10) (2023) 324-338.
- [41] J. Wang, Q. Zhang, D. Zhao, Y. Chen, Lane change decision-making through deep reinforcement learning with rule-based constraints, in: *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1-6.
- [42] W. Zhou, D. Chen, J. Yan, Z. Li, H. Yin, W. Ge, Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic, *Autonomous Intelligent Systems*, 2(1) (2022) 5.
- [43] S. Hwang, K. Lee, H. Jeon, D. Kum, Autonomous vehicle cut-in algorithm for lane-merging scenarios via policy-based reinforcement learning nested within finite-state machine, *IEEE Transactions on Intelligent Transportation Systems*, 23(10) (2022) 17594-17606.
- [44] M. Bouton, A. Nakhaei, D. Isele, K. Fujimura, M.J. Kochenderfer, Reinforcement learning with iterative reasoning for merging in dense traffic, in: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2020, pp. 1-6.

- [45] B.R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A.A. Al Sallab, S. Yogamani, P. Pérez, Deep reinforcement learning for autonomous driving: A survey, *IEEE transactions on intelligent transportation systems*, 23(6) (2021) 4909-4926.
- [46] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fiedjeland, G. Ostrovski, Human-level control through deep reinforcement learning, *nature*, 518(7540) (2015) 529-533.
- [47] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, *arXiv preprint arXiv:1707.06347*, (2017).
- [48] M. Eppe, P.D. Nguyen, S. Wermter, From semantics to execution: Integrating action planning with reinforcement learning for robotic causal problem-solving, *Frontiers in Robotics and AI*, 6 (2019) 123.