Reward Machine Reinforcement Learning for Autonomous Highway Driving—A Unified Framework for Safety and Performance

Abstract

Developing a safe and highly effective policy for autonomous vehicles (AVs) on highways continues to pose a significant challenge in contemporary reinforcement learning. In this study, we introduce a novel reinforcement learning approach based on reward machines. By tailoring the reward function to the specific needs of AVs in highway scenarios, we enable them to make more informed and efficient decisions. Our focus is on designing a reward function that formalizes traffic rules, which is crucial for achieving safe and reliable AV behavior on highways. To address this problem, we propose several innovative ideas that go beyond existing algorithmic techniques, specifically aimed at facilitating exploration and exploitation. This is the first reinforcement learning autonomous highway driving algorithm that inherently follows the Vienna Convention on road traffic. Experimental results demonstrate the effectiveness of the proposed approach, which significantly improves AVs' safety and performance on highways.

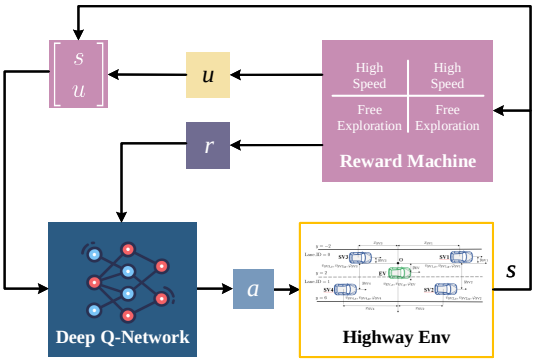
1 Introduction

Autonomous driving (AD) is a cutting-edge technology that seamlessly integrates the automotive industry with new-generation information technologies such as artificial intelligence (AI), 5G communication, and high-performance computing in the field of transportation [1–3]. However, existing autonomous driving systems are primarily suitable for straightforward, well-defined traffic scenarios, and possess limited self-learning abilities, making it difficult to guarantee safety [4, 5]. Consequently, autonomous vehicles (AVs) struggle to navigate unfamiliar, intricate, and time-varying traffic environments, which severely restricts their rapid deployment and development in the real world. As such, ensuring the safety of machine learning for AVs in complex, uncertain, and open traffic scenarios has become an increasingly crucial scientific problem [6].

Reinforcement learning (RL) is a robust machine learning framework capable of learning complex action-decision policies in high-dimensional traffic environments [7, 8]. A standard assumption in RL is that the agent does not have access to the environment model but interacts with the environment and learns from its experience. Although assuming that the transition probabilities are unknown seems reasonable, there is less justification for hiding the reward function from the agent. For AVs on highways, automotive engineers must program reward functions by a case-by-case way [5, 9–11], while the agent does not have access to the structures or high-level ideas that the engineers may have used in defining it. We consider a unified framework to do so in this paper.

In previous work, efforts to equip an agent with knowledge about reward functions have centered around defining a task specification language, typically based on hierarchical structures [12, 13] or linear temporal logic [14, 15], and then generating a reward function aimed at fulfilling that specification. Recently, the authors in [16–18] introduced a novel solution in the form of a finite state machine, known as the Reward Machine (RM), for defining rewards. The RM methodology affords the ability to combine different reward functions in flexible ways, ultimately outputting the reward function that the agent should utilize at any given time. This approach facilitates the decomposition of complex problems and significantly expedites the learning process.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.



(a)

	Lane Keeping	Lane Change
Obey Traffic Rules	High Speed Reward	High Speed Reward
Violate Traffic Rules	Zero Reward	Zero Reward

(b)

Figure 1: Visualization of the proposed approach: (a) The reward machine reinforcement learning framework; (b) Reward machine with traffic rules.

The work presented in this paper offers two key contributions. Firstly, we introduce a novel approach to reward machine reinforcement learning, specifically tailored for AVs on highways. As illustrated in Figure 1, our proposed scheme leverages the internal structure of reward machines to break down complex tasks related to AD in open traffic scenarios. Reward machine reinforcement learning provides a distinct advantage by allowing the agent to integrate traffic rules with RL's self-evolving capabilities. This not only enhances driving safety, but also improves overall performance and sample efficiency.

Our second contribution is to introduce a quantifiable traffic rule, called the safety distance [19–22]. This rule enables us to obtain precisely defined specifications from the Vienna Convention on road traffic [6]. The safety distance has also been utilized in [23] to verify RL during operation. However, no previous works have focused on designing RL using the safety distance. Therefore, we demonstrate how the safety distance can be combined with a reward machine to enhance AVs' safety.

Finally, our experiments in the highway-env environment [9] demonstrate that the proposed algorithm outperforms conventional RL methods. Specifically, the collision rate was reduced to nearly zero, while sacrificing minimal speed.

2 Preliminaries

2.1 Reinforcement Learning (RL)

The problem of reinforcement learning (RL) involves an agent interacting with an environment whose dynamics are unknown. Typically, the environment is modeled as a Markov Decision Process (MDP), which is a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker. An MDP is defined by a tuple $\mathcal{M} =< S, A, r, p, \gamma >$, where S is a finite set of states, A is a finite set of actions, $r : S \times A \times S \to \mathbb{R}$ is the reward function, $p(s'|s, a)$ is the probability distribution of the next state s' given the current state s and action a, and $\gamma \in (0, 1]$ is the discount factor that determines the trade-off between immediate and future rewards. The MDP framework provides a structured way to analyze the RL problem and develop algorithms for finding optimal policies that maximize the expected cumulative reward over time.

In RL, a policy $\pi(a|s)$ is a probability distribution over the actions $a \in A$ given a state $s \in S$. At each time step, the agent, when in a particular state $s \in S$, selects an action a according to $\pi(\cdot|s)$ and executes it. The agent then receives the next state $s' \sim p(\cdot|s, a)$ and the current reward $r(s, a, s')$ from the environment. This process repeats from s'. The goal of the agent is to find a policy π^* that maximizes the expected discounted future reward from every state in S. In other words, the agent seeks to select actions that maximize the cumulative reward it expects to receive over time, taking into account the discount factor γ. By finding an optimal policy, the agent can act in a way that maximizes its long-term performance in the environment.

Furthermore, the Q-function $Q_\pi(s,a)$ under a policy π is defined as the expected discounted future reward of taking action a in state s and then following policy π. It represents the quality of taking a particular action in a particular state, given the policy that is being followed. It is worth noting that every optimal policy π^* satisfies the Bellman equations, which relate the value of the Q-function at a given state-action pair to the values of the Q-function at the next state-action pairs:

$$Q^*(s,a) = \sum_{s' \in S} p(s'|s,a) \left(r(s,a,s') + \gamma \max_{a' \in A} Q^*(s',a') \right) \tag{1}$$

for each state $s \in S$ and action $a \in A$, where $Q^* = Q_{\pi^*}$. The optimal policy can be computed by selecting the action a with the highest value of $Q^*(s,a)$ in each state s.

2.2 Deep Q-Networks (DQN)

Deep Q-Network (DQN) is a powerful algorithm used in RL that approximates $Q(s,a) \approx Q_\theta(s,a)$ using a deep neural network with parameters θ. This method has been shown to be effective in solving complex problems, especially when the state and action spaces are continuous.

The training process involves randomly sampling mini-batches of experiences (s,a,r,s') from an experience replay buffer. These experiences are then used to minimize the square error between $Q_\theta(s,a)$ and the Bellman equations' estimate $r(s,a,s') + \gamma \max_{a'} Q_{\theta'}(s',a')$. The updates are made with respect to a target network with parameters θ'. During the training process, the parameters θ' are held fixed when minimizing the square error but updated to θ after a certain number of training updates. The target network's role is to stabilize learning and prevent the algorithm from overfitting.

One of the advantages of DQN is that it inherits the off-policy behavior from Q-learning. However, it is no longer guaranteed to converge to an optimal policy. Despite this limitation, DQN has been successfully applied to a variety of domains like games, robotics, and particularly autonomous driving.

2.3 Reward Machines (RM)

In this section, we introduce an innovative type of finite state machine called the Reward Machine (RM). The RM takes abstracted environment descriptions as input and produces reward functions as output. The primary goal of the RM is to enable the agent to receive different rewards based on the transitions made within the RM. This allows for the creation of temporally extended tasks and behaviors that are non-Markovian relative to the environment.

Formally, an RM can be defined as a finite state machine that takes an abstracted environment description as input and generates a reward function as output.

Reward Machine. [18] Given a set of propositional symbols $\mathcal{P}$, a set of states S, and a set of actions A, a Reward Machine (RM) is a tuple $\mathcal{R}_{\text{PSA}} = \langle U, u_0, \delta_u, \delta_r \rangle$ where U is a finite set of states, $u_0 \in U$ is an initial state, $\delta_u : U \times 2^{\mathcal{P}} \to U$ is the state-transition function and $\delta_r : U \times U \to [S \times A \times S \to \mathbb{R}]$ is the reward-transition function.

The behavior of an RM $\mathcal{R}_{\text{PSA}}$ can be described as follows: it starts in a particular state u_0 and subsequently transitions to other states $u \in U$ at each time step. At every step, the RM receives a truth assignment σ_t as input, which is a set of propositions from $\mathcal{P}$ that are true in the current state s. The RM then moves to the next state $u' = \delta_u(u,\sigma)$ based on the state-transition function, and outputs a reward function $r = \delta_r(u,u')$ based on the reward-transition function. Consequently, the RM operates by taking in a truth assignment as input, transitioning to a new state, and producing a reward based on the transition.

A RM allows us to define a single reward function over a larger state space, which provides a powerful means of describing complex reward structures. By leveraging the power of RMs, we can define reward functions that are tailored to the specific needs of AVs in highway scenarios, making them more capable and efficient in their decision-making processes.

3 The Proposed Approach

In this section, we introduce Reward Machine Reinforcement Learning, a novel approach for learning complex decision-making policies for AVs on highways. We begin by presenting the problem

formulation of autonomous driving and formalizing the traffic rules on the highway. Next, we design a reward machine that aids in the decision-making of AVs on the highway. Finally, we demonstrate how to combine the reward machine with deep reinforcement learning.

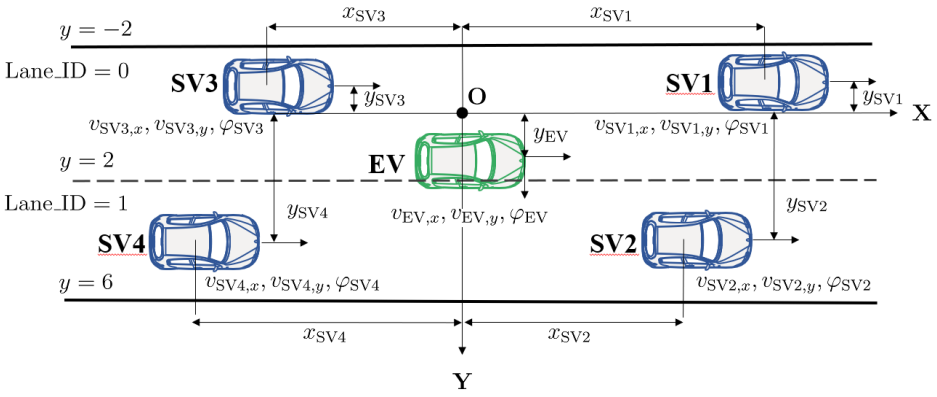


Figure 2: Visualization of the ego vehicle and surrounding vehicles on a two-lane highway.

3.1 Problem Formulation of AVs on Highways

The experimental scenarios for AVs on highways are configured using the highway-env environment. As depicted in Figure 2, we model a typical two-lane highway with an ego vehicle (EV) and several surrounding vehicles (SVs). Each lane has a width of 4 meters, and the speed limit of the highway is 30 meters per second. We set the origin of coordinates at the center of the left lane, following the longitudinal center of the EV.

The initial positions of the SVs are uniformly distributed, and their initial speeds are randomly selected from the range of 23 to 25 meters per second. The driver model for the SVs is a combination of the IDM and MOBILE models [24, 25]. As illustrated in Figure 2, four SVs are positioned to collect states: one in front of the EV, one behind the EV, one ahead of the EV on the other lane, and one behind the EV on the other lane. The states considered for each SV include their longitudinal positions $x_{\text{SV}i}$, lateral positions $y_{\text{SV}i}$, longitudinal speeds $v_{\text{SV}i,x}$, lateral speeds $v_{\text{SV}i,y}$, and heading angles $\varphi_{\text{SV}i}$, with $i = 1, \ldots, 4$.

The EV is initialized with a longitudinal speed of 25 meters per second. Its initial lateral position is randomly sampled from the set $\text{Lane\_ID}(\cdot) = \{0, 1\}$. Note that $\text{Cur\_ID} = \text{Lane\_ID}(\text{EV}) = 0$ indicates that the EV is in the left lane, while $\text{Tar\_ID} = \text{Lane\_ID}(\text{EV}') = 1$ indicates that the target lane of the EV is the right lane. The longitudinal relative position of the EV is always fixed at $x_{\text{EV}} = 0$ meters, while its lateral position is denoted by y_{EV}. The longitudinal and lateral speeds, as well as the heading angle of the EV, are denoted by $v_{\text{EV},x}$, $v_{\text{EV},y}$, and φ_{EV}, respectively. The term $\text{Mid\_Lane}(y_{\text{EV}}) \in \{0, 1\}$ is used to indicate whether the EV is near the centerline of the road, which is defined as follows:

$$\text{Mid\_Lane}(y_{\text{EV}}) = \begin{cases} 0 & \text{If } |y_{\text{EV}} - 2| \geq 1 \\ 1 & \text{Otherwise.} \end{cases} \tag{2}$$

The decision-making actions for the EV consist of the following discrete variables:

$$a = \{\text{Faster}, \ \text{Idle}, \ \text{Slower}, \ \text{Lane\_Left}, \ \text{Lane\_Right}\} \tag{3}$$

where Faster : $\bar{v}_{\text{EV},x} = v_{\text{EV},x} + \Delta v_1$, Idle : $\bar{v}_{\text{EV},x} = v_{\text{EV},x}$ and Slower : $\bar{v}_{\text{EV},x} = v_{\text{EV},x} - \Delta v_2$ are the decisions for increasing, holding and decreasing the target speed $\bar{v}_{x,\text{EV}}$ of the EV, respectively. Note that Δv_1 and Δv_2 represent two speed increments.

Lane_Left and Lane_Right are the lane change decisions for modifying the target lane of the EV. The computation logic is defined as follows:

$$\text{Tar\_ID} = \begin{cases} \text{Cur\_ID} - 1 & \text{If } (a = \text{Lane\_Left}) \wedge (\text{Cur\_ID} = 1) \\ \text{Cur\_ID} + 1 & \text{If } (a = \text{Lane\_Right}) \wedge (\text{Cur\_ID} = 0) \\ \text{Cur\_ID} & \text{Otherwise.} \end{cases} \tag{4}$$

After obtaining the target lane Tar_ID and target speed $\bar{v}_{\mathrm{EV},x}$ through the decision-making module, they are then passed on to the control module, which utilizes the IDM and MOBILE models to compute continuous speed control and steering signals. However, due to space limitations, the details of both models will not be discussed in this paper.

3.2 Formalized Traffic Rules

When it comes to deploying AVs in the real world, safety is the most crucial aspect to consider. In particular, ensuring the safety of autonomous driving means being able to adhere to road boundaries and avoid collisions with obstacles and other traffic participants. To achieve this for AVs on highways, formalized traffic rules based on the Vienna Convention [26] on Road Traffic have been introduced, with the safety distance being at the core of these rules.

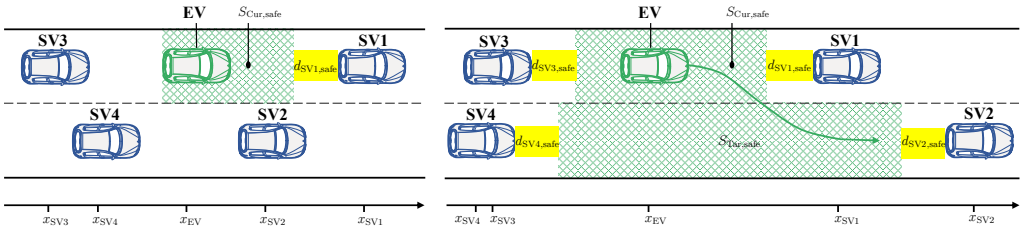


Figure 3: Visualization of the safe space and safety distances: (a) the safe space (shaded area) and the safety distance (yellow area) of a leading vehicle during the lane keeping case; (b) the safe spaces (shaded area) and the safety distances (yellow area) of SVs during the lane change case.

Figure 3 demonstrates the concept of the safe distance using the examples of $\{\mathrm{EV}, \mathrm{SV}i\}$. Without loss of generalization, the safe distance $d_{\mathrm{SV1,safe}} > 0$ between the ego vehicle and the leading vehicle (SV1) must be large enough for the ego vehicle to stop behind SV1 if SV1 performs an emergency brake with the maximum absolute acceleration $a_{\mathrm{SV1,max}}$. It's important to note that the future position of a vehicle at a point in time $t \geq 0$ can be described by the general motion equation: $d(t) = d_0 + vt + 0.5at^2$, where $d_0 \in \mathbb{R}$ represents the position of the vehicle at t_0. If the positions of the ego vehicle and leading vehicle SV1 are equal for some $t \geq 0$, i.e., $\exists t \geq 0 : d_{\mathrm{EV}}(t) = d_{\mathrm{SV1}}(t)$, then they collide.

For any SVi with $i = 1, \ldots, 4$, the definition of safety distance can be extended to include reaction times δ_t. In this case, the minimum required safety distance between the ego vehicle and surrounding vehicles depends on the following condition [21]:

For $i = 1$ and 2 (front vehicles)

$$(d_{\mathrm{SV}i}(\delta_t) \leq d_{\mathrm{EV,max}}) \wedge (|a_{\mathrm{SV}i,\mathrm{max}}| < |a_{\mathrm{EV,max}}|) \wedge (v^*_{\mathrm{SV}i} < v_{\mathrm{EV}}) \wedge \left(t_{\mathrm{EV,stop}} < t^*_{\mathrm{SV}i,\mathrm{stop}}\right) \tag{5}$$

For $i = 3$ and 4 (rear vehicles)

$$(d_{\mathrm{EV}}(\delta_t) \leq d_{\mathrm{SV}i,\mathrm{max}}) \wedge (|a_{\mathrm{EV,max}}| < |a_{\mathrm{SV}i,\mathrm{max}}|) \wedge (v^*_{\mathrm{EV}} < v_{\mathrm{SV}i}) \wedge \left(t_{\mathrm{SV}i,\mathrm{stop}} < t^*_{\mathrm{EV,stop}}\right) \tag{6}$$

Note that $d_{\mathrm{SV}i}(\delta_t)$ represents the future position of the SVi at time δ_t. The variable $d_{\mathrm{EV,max}}$ is the stopping distance of the EV during the reaction time δ_t, while $v^*_{\mathrm{SV}i}$ represents the velocity of SVi at time δ_t after starting emergency braking. The stopping time of the EV is calculated as $t_{\mathrm{EV,stop}} = v_{\mathrm{EV}}/|a_{\mathrm{EV,max}}|$, and the stopping time of SVi is calculated as $t^*_{\mathrm{SV}i,\mathrm{stop}} = v^*_{\mathrm{SV}i}/|a_{\mathrm{SV}i,\mathrm{max}}|$. The symbols used in (6) are comparable to those in (5) with exception that EV and SVi are interchanged.

Then, the safety distances between EV and SVi are defined as [22]

$$d_{\mathrm{SV}i,\mathrm{safe\_1}} = v_{\mathrm{EV}}\delta_t + \frac{v^2_{\mathrm{EV}}}{2a_{\mathrm{EV}}} \tag{7}$$

$$d_{\mathrm{SV}i,\mathrm{safe\_2}} = d_{\mathrm{SV}i,\mathrm{safe\_1}} - v_{\mathrm{SV}i}\delta_t + \frac{1}{2}a_{\mathrm{SV}i}\delta_t^2 \tag{8}$$

$$d_{\mathrm{SV}i,\mathrm{safe\_3}} = \begin{cases} \frac{(v_{\mathrm{SV}i} - |a_{\mathrm{SV}i,\mathrm{max}}|\delta_t - v_{\mathrm{EV}})^2}{-2(|a_{\mathrm{SV}i,\mathrm{max}}| - |a_{\mathrm{EV,max}}|)} - v_{\mathrm{SV}i}\delta_t + \frac{1}{2}|a_{\mathrm{SV}i,\mathrm{max}}|\delta_t^2 + v_{\mathrm{EV}}\delta_t & \text{If (5) is true,} \\ \frac{v^2_{\mathrm{SV}i}}{-2|a_{\mathrm{SV}i,\mathrm{max}}|} - \frac{v^2_{\mathrm{EV}}}{-2|a_{\mathrm{EV,max}}|} + v_{\mathrm{EV}}\delta_t & \text{Otherwise.} \end{cases} \tag{9}$$

for $i = 1$ and 2 (front vehicles), and

$$d_{\text{SV}i,\text{safe\_1}} = v_{\text{SV}i}\delta_t + \frac{v_{\text{SV}i}^2}{2a_{\text{SV}i}} \tag{10}$$

$$d_{\text{SV}i,\text{safe\_2}} = d_{\text{SV,safe\_1}} - v_{\text{EV}}\delta_t + \frac{1}{2}a_{\text{EV}}\delta_t^2 \tag{11}$$

$$d_{\text{SV}i,\text{safe\_3}} = \begin{cases} \frac{(v_{\text{EV}}-|a_{\text{EV,max}}|\delta_t - v_{\text{SV}i})^2}{-2(|a_{\text{EV,max}}|-|a_{\text{SV}i,\text{max}}|)} - v_{\text{EV}}\delta_t + \frac{1}{2}|a_{\text{EV,max}}|\delta_t^2 + v_{\text{SV}i}\delta_t & \text{If (6) is true,} \\ \frac{v_{\text{EV}}^2}{-2|a_{\text{EV,max}}|} - \frac{v_{\text{SV}i}^2}{-2|a_{\text{SV}i,\text{max}}|} + v_{\text{SV}i}\delta_t & \text{Otherwise.} \end{cases} \tag{12}$$

for $i = 3$ and 4 (rear vehicles).

A traffic rule can be established for the ego vehicle to maintain a safe distance from the vehicle SVi by introducing the longitudinal relative distance $x_{\text{SV}i}$. Referring to [21, 22], we denote the relative distance $d_{\text{SV}i} = x_{\text{FV}i} - x_{\text{RV}i}$ between the front vehicle and the rear vehicle and propose a new formulated traffic rule that satisfies the Vienna Convention as follows:

$$\text{Rule\_SV}i = \begin{cases} 0 & \text{If } (d_{\text{SV}i} \le 0) \vee \left(d_{\text{SV}i} \le v_{\text{RV}i}\delta_t - v_{\text{FV}i}\delta_t + \frac{1}{2}a_{\text{FV}i}\delta_t^2\right) \\ 1 & \text{If } (d_{\text{SV}i} > d_{\text{SV}i,\text{safe\_1}}) \vee (\delta_t \le t_{\text{FV}i,\text{stop}}) \wedge (d_{\text{SV}i} > d_{\text{SV}i,\text{safe\_2}}) \\ d_{\text{SV}i} > d_{\text{SV}i,\text{safe\_3}} & \text{Otherwise.} \end{cases} \tag{13}$$

where $i = 1, \ldots, 4$, and the subscripts $(\text{FV}i, \text{RV}i)$ either equal to $(\text{EV}, \text{SV}i)$ for $i = 1, 2$ or equal to $(\text{SV}i, \text{EV})$ for $i = 3, 4$.

Note that the main difference from the original traffic rule in [21, 22] is using the logical condition $(x_{\text{SV}i} \le 0) \vee \left(x_{\text{SV}i} \le v_{\text{RV}i}\delta_t - v_{\text{FV}i}\delta_t + \frac{1}{2}a_{\text{FV}i}\delta_t^2\right)$ to characterize the condition where the ego vehicle position at time δ_t is already in front of the surrounding vehicles. If the EV continuously drives within the safety space, i.e., $\forall t \ge 0 : p_{\text{EV}} = [x_{\text{EV}}, y_{\text{EV}}]^T \in \mathcal{S}_{\text{safe}}$, safety is guaranteed for an infinite time horizon as t approaches infinity, and the ego vehicle will not be held responsible for any collision. The upcoming subsection will use the formalized traffic rules for safety as presented in equations (5)-(13), to design the reward machine for AVs on highways.

3.3 Design of A Reward Machine for AVs on Highways

In this subsection, we present a novel Reward Machine (RM) that takes abstracted descriptions of complex autonomous driving environments as input and outputs reward functions for reinforcement learning (RL). The motivation behind designing an RM is to ensure that the ego vehicle (agent) is rewarded differently for different maneuvers. For AVs on highways, there are mainly two types of maneuvers: lane keeping and lane changing. In accordance with formalized traffic rules, we have two traffic conditions: the safety condition and the safety-free condition.

A lane keeping maneuver, as shown in Figure 3 (a), only takes into consideration the safety distance of the leading vehicle. If the traffic rule for the lane keeping maneuver is met and the current lane of the EV is the same as its target lane, the EV follows the first RM state u_1. If the EV does not meet the safety condition for the lane keeping maneuver, the RM transitions to the second RM state u_2.

In contrast to lane keeping, the lane change maneuver requires considering the safety distances of all four SVs simultaneously, as illustrated in Figure 3 (b). Therefore, we define state u_3 represents the EV taking a lane change maneuver while meeting the safety condition, while state u_4 indicates that the EV violates the formalized traffic rules during the lane change maneuver.

To conclude, the state of the reward machine for AVs on highways is presented as

$$u_1 = \delta_u(\sigma_1),\ \sigma_1 = \neg\text{Mid\_Lane}(y_{\text{EV}}) \wedge (|y_{\text{EV}} - \text{Tar\_ID} \times \text{Lane\_Width}| < 2) \wedge \text{Rule\_SV1} \tag{14}$$

$$u_2 = \delta_u(\sigma_2),\ \sigma_2 = \neg\text{Mid\_Lane}(y_{\text{EV}}) \wedge (|y_{\text{EV}} - \text{Tar\_ID} \times \text{Lane\_Width}| < 2) \wedge \neg\text{Rule\_SV1} \tag{15}$$

$$u_3 = \delta_u(\sigma_3),\ \sigma_3 = (\text{Mid\_Lane}(y_{\text{EV}}) \vee (|y_{\text{EV}} - \text{Tar\_ID} \times \text{Lane\_Width}| \ge 2)) \wedge \text{Rule\_SV1} \wedge \text{Rule\_SV2} \wedge \text{Rule\_SV3} \wedge \text{Rule\_SV4} \tag{16}$$

$$u_4 = \delta_u(\sigma_4),\ \sigma_4 = (\text{Mid\_Lane}(y_{\text{EV}}) \vee (|y_{\text{EV}} - \text{Tar\_ID} \times \text{Lane\_Width}| \ge 2)) \wedge (\neg\text{Rule\_SV1} \vee \neg\text{Rule\_SV2} \vee \neg\text{Rule\_SV3} \vee \neg\text{Rule\_SV4}) \tag{17}$$

where $|y_{\text{EV}} - \text{Tar\_ID} \times \text{Lane\_Width}|$ is used to check whether the target lane is the same with the current lane of the EV, and the safety conditions are determined by the formalized traffic rules in (13).

Based on the above RM states, we can now formally define the final reward as follows:

$$r = \delta_r(u) = \begin{cases} \frac{v_{\text{EV}}}{v^*_{\text{EV}}} & \text{If } u = u_1 \\ 0 & \text{If } u = u_2 \\ \frac{v_{\text{EV}}}{v^*_{\text{EV}}} & \text{If } u = u_3 \\ 0 & \text{If } u = u_4 \end{cases} \tag{18}$$

$$v^*_{\text{EV}} = \begin{cases} v_{\text{SV1}} & \text{If } (x_{\text{SV1}} \leq d_{\text{ACC}}) \wedge \left(\vee_{i=1}^{4} \neg \text{Rule\_SV}i\right) \\ v_{\max} & \text{Otherwise.} \end{cases}$$

where d_{ACC} is the threshold for adaptive cruise control (ACC). It is worth noting that the reward function in Equation (18) can support multi-task learning: we incentivize the agent to move at high speed within the safe space S_{safe}, while maintaining a zero reward if any traffic rule is violated.

3.4 Reward Machine Reinforcement Learning

By defining the RM, we can utilize it to reward an agent. Algorithm 1 shows pseudo-code for solving reward machine reinforcement learning. The main difference with standard reinforcement learning is that it learns Q-values over the cross-product $Q(s, u, a)$.

Algorithm 1: The reward machine reinforcement learning

Input : $S, A, \gamma \in (0, 1], \alpha \in (0, 1], \epsilon \in (0, 1], P, \sigma, U, u_0, F, \delta_u, \delta_r$

For all $s \in S$, $u \in U$, and $a \in A$, initialize $Q(s, u, a)$ arbitrarily
for $l \leftarrow 0$ *to num episodes* **do**
  Initialize $u \leftarrow u_0$ and $s \leftarrow EnvInitialState()$
  while s *is not terminal* **do**
    Choose action a from (s, u) using policy derived from Q (e.g., ε-greedy)
    Take action a and observe the next state s' Compute the next RM state $u' \leftarrow \delta_u(\sigma)$ in (14)-(17) and the current reward $r \leftarrow \delta_r(u)$ in (18)
    if s' *is terminal* **then**
      $Q(s, u, a) \leftarrow Q(s, u, a) + \alpha(r - Q(s, u, a))$
    end
    else
      $Q(s, u, a) \leftarrow Q(s, u, a) + \alpha\left(r + \gamma \max_{a' \in A} Q(s', u', a') - Q(s, u, a)\right)$
    end
    Update $s \leftarrow s'$ and $u \leftarrow u'$
  end
end

4 Experimental Evaluation

4.1 Evaluation Metrics and Baselines

In this section, we present a quantifiable evaluation of the proposed approach within the highway-env environment under various traffic conditions, as shown in Table 1. Our experiments are designed to explore and measure the following metrics:

1. **Completion**: This metric measures the number of successfully completed episodes and the number of instances where the agent crashed. To convert it into an error measure, the metric is computed as follows:

$$\mathcal{M}_C = \frac{n_{\text{collision}}}{n_{\text{total}}} \times 100\% \tag{19}$$

where $n_{\text{collision}}$ is the number of collisions and n_{total} is the total number of episodes.

Table 1: Traffic levels of simulation environment.

Traffic Level	Density (N_v/km)	Average Headway (m)	Average Speed (km/h)	Flow (N_v/sec)
Level A	13∼18	57.83∼77.10	75.60∼86.40	0.319
Level B	18∼24	43.37∼57.83	75.60∼86.40	0.418
Level C	24∼31	32.53∼43.37	75.60∼86.40	0.531
Level D	31∼41	24.40∼32.53	75.60∼86.40	0.665
Level E	41∼55	18.30∼24.40	75.60∼86.40	0.794
Level F	55∼73	13.72∼18.30	75.60∼86.40	0.958

2. **Average Speed**: This metric measures the average speed at which the agent completes an episode. It is intended to capture the efficiency of the autonomous driving algorithm:

$$\mathcal{M}_S = \frac{1}{n_{\text{total}}} \sum_{i=1}^{n_{\text{total}}} \left(\frac{1}{n_{i,\max}} \sum_{j=1}^{n_{i,\max}} |v_{\text{EV}}(i,j)| \right) \tag{20}$$

where $n_{i,\max}$ represents the number of samples taken during the i-th episode until it reaches termination, either due to the vehicle crashing or reaching the time limit.

To evaluate the performance of the proposed algorithm, we compare it against several baseline approaches in various simulation conditions, including different traffic levels. The baselines used are as follows:

1. **Baseline 1 (IDM+MOBILE)**: This baseline employs a classical driver model-based algorithm. It combines the IDM algorithm [24] for longitudinal control and the MOBILE algorithm [25] for lateral control. Baseline 1 prioritizes safety but suffers from drawbacks such as reduced speed and limited intelligence.
2. **Baseline 2 (DQN without Reward Machine)**: Baseline 2 utilizes a DQN algorithm but does not incorporate a reward machine. In this baseline, the reward is determined by a weighting function based on speed and collision [9].
3. **Baseline 3 (Human Tester)**: Baseline 3 involves the participation of ten students (five boys and five girls) who play the highway-env game under the same conditions. The average performance of the human testers is evaluated. Due to the high diversity among human testers, Baseline 3 simulates real-life highway scenarios with more randomness.

4.2 Experimental Setup

We conducted randomized experiments to test each algorithm, with the number of surrounding vehicles set to 10, a time limit of 40 seconds, and a policy frequency of 8 Hz. The initial positions of the surrounding vehicles were uniformly distributed, while the initial lateral position of the ego vehicle was randomly selected from the set Lane_ID$(\cdot) = \{0, 1\}$. The initial speed of the ego vehicle was set to 25 m/s, and the initial speeds of the surrounding vehicles were randomly chosen from the range of 21 to 24 m/s. Traffic levels A and D are used for training neural networks.

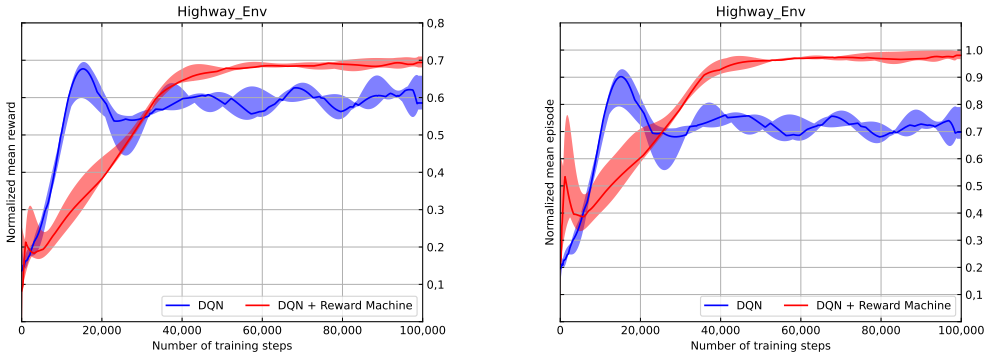
The proposed approach uses a Q function with an input vector consisting of 34 states at each sampling time. These states not only include the states of the ego vehicle and the closest four SVs to the agent (x_{EV}, y_{EV}, $v_{\text{EV},x}$, $v_{\text{EV},y}$, φ_{EV}, $x_{\text{SV}i}$, $y_{\text{SV}i}$, $v_{\text{SV}i,x}$, $v_{\text{SV}i,y}$ and $\varphi_{\text{SV}i}$, where $i = 1, 2, 3, 4$) but also consider Cur_ID, Tar_ID, Mid_Lane, Tar_Speed, u_i, where $i = 0, \ldots, 4$, Tar_Speed represents the target speed of the ego vehicle, and u_0 is the initial state for the reward machine. In contrast, the baseline approaches only consider the states of the ego vehicle and the closest four SVs to the agent. The DQN algorithms were trained over 1×10^5 steps. After training, each algorithm was independently tested over 5×10^3 runs, and we report the median performance and percentiles 14 to 86 over the runs.

4.3 Experimental Results

The proposed approaches were evaluated within the highway-env environment. During each training iteration, we utilized specific parameter values to ensure effective learning. For exploration, we set

$\varepsilon = 0.1$, while the discounted rate was set to $\gamma = 0.8$. Additionally, a learning rate of $\alpha = 5 \times 10^{-4}$ was employed. The training results of the proposed method and Baseline 2 (DQN without reward machine) are depicted in Figures 4. These figures demonstrate the effectiveness of the proposed approach in learning optimal and smooth driving tasks. In contrast, the baseline method converges to suboptimal policies and exhibits significant fluctuations. This observation highlights the insufficiency of the training steps undertaken by the baseline method to discover an optimal policy.

Figure 5 presents a comparative analysis of the performance between the proposed RL algorithm and Baselines 1-3 at varying traffic levels. The assessment of these algorithms is based on the balance between completion ($\mathcal{M}_C$) and average speed ($\mathcal{M}_S$). The top-right area of the graph indicates superior performance. It is important to highlight that Baseline 1 (IDM+MOBILE) ensures collision-free driving but consistently operates at low speeds. Baselines 2 (RL) and 3 (human tester) achieve higher average speeds than Baseline 1 under low traffic levels. However, as the traffic level increases, both the RL algorithm and human testers experience a significant decline in completion rates, raising concerns about the safety of autonomous driving. Among the compared algorithms, the proposed reward machine RL algorithm consistently achieves completion rates exceeding 99.95% in all traffic scenarios. The proposed unified framework consistently occupies the top-right corner of the graph, demonstrating its superior performance compared to the other algorithms and human.



(a) Average rewards of DQNs.

(b) Average episodes of DQNs.

Figure 4: Training results of DQNs in the highway-env environment.

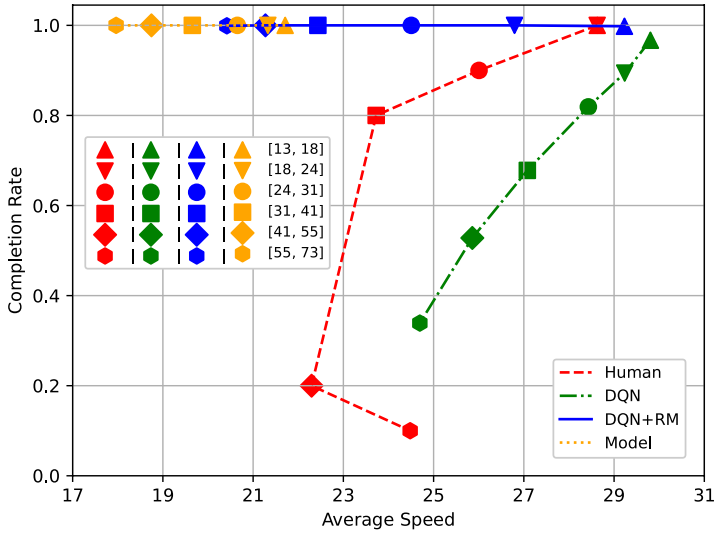


Figure 5: Comparison of the algorithms and human by the trade-off between completion rate and average speed at various traffic levels.

5 Conclusion

In this paper, we developed a reinforcement learning scheme that utilizes state machines to specify the reward function of an AV agent operating on highways. Our proposed reward machine allows for the specification of high speed and sparse rewards, specifically for temporally-based lane keeping and lane change maneuvers. By formalizing a subset of the traffic rules elicited from the Vienna Convention on road traffic for highways, our reward machine significantly improves safety, performance and sample efficiency for autonomous highway driving.

The research on reinforcement learning technologies in AVs is just at the trough of disillusionment. Numerous problems need to be considered to design more general learning-based autonomous driving models that can operate in open traffic environment. In future research, there should be a focus on developing more comprehensive reinforcement learning algorithms that can handle complex traffic scenarios, such as multilane highway and merging area.

References

[1] J. Wang, J. Liu, N. Kato, Networking and communications in autonomous driving: A survey, IEEE Commun. Surv. Tutor 21 (2) (2019) 1243–1274.

[2] L. Claussmann, M. Revilloud, D. Gruyer, S. Glaser, A review of motion planning for highway autonomous driving, IEEE trans Intell Transp Syst 21 (5) (2020) 1826–1848.

[3] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, W. Shi, Computing systems for autonomous driving: State of the art and challenges, IEEE Internet Things J. 8 (8) (2021) 6469–6486.

[4] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, J. Li, Deep learning for LiDAR point clouds in autonomous driving: A review, IEEE trans Neural Netw Learn Syst 32 (8) (2020) 3412–3432.

[5] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, P. Perez, Deep reinforcement learning for autonomous driving: A survey, IEEE trans Intell Transp Syst 23 (6) (2022) 4909–4926.

[6] N. Mehdipour, M. Althoff, R. D. Tebbens, C. Belta, Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges, Automatica 152 (2023) 110692.

[7] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction (2nd Edition), MIT Press Cambridge, 1999.

[8] L. Fridman, J. Terwilliger, B. Jenik, Deeptraffic: Crowdsourced hyperparameter tuning of deep reinforcement learning systems for multi-agent dense traffic navigation, in: Neural Information Processing Systems (NIPS 2018) Deep Reinforcement Learning Workshop.

[9] E. Leurent, An environment for autonomous driving decision-making, `https://github.com/eleurent/highway-env` (2018).

[10] M. Zhou, Y. Yu, X. Qu, Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach, IEEE trans Intell Transp Syst 21 (1) (2020) 433–443.

[11] O. Nassef, L. Sequeira, E. Salam, T. Mahmoodi, Building a lane merge coordination for connected vehicles using deep reinforcement learning, IEEE Internet Things J. 8 (4) (2020) 2540–2557.

[12] S. P. Singh, Transfer of learning by composing solutions of elemental sequential tasks, Machine Learning 8 (3-4) (1992) 323–339.

[13] R. S. Sutton, D. Precup, S. Singh, Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning, Artificial Intelligence 112 (1-2) (1999) 181–211.

[14] X. Li, C.-l. Vasile, C. Belta, Reinforcement learning with temporal logic rewards, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 3834–3839.

[15] R. Toro Icarte, T. Q. Klassen, R. Valenzano, S. A. McIlraith, Teaching multiple tasks to an RL agent using LTL, in: 17th International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 452–461.

[16] R. Toro Icarte, T. Q. Klassen, R. Valenzano, S. A. McIlraith, Using reward machines for high-level task specification and decomposition in reinforcement learning, in: 35th International Conference on Machine Learning (ICML), 2018, pp. 2107–2116.

[17] R. Toro Icarte, E. Waldie, T. Q. Klassen, R. Valenzano, M. P. Castro, S. A. McIlraith, Learning reward machines for partially observable reinforcement learning, in: 35th Conference on Neural Information Processing System (NeurIPS 2019), 2019.

[18] R. Toro Icarte, T. Q. Klassen, R. Valenzano, S. A. McIlraith, Reward machines: Exploiting reward function structure in reinforcement learning, J Artif Intell Res 73 (2022) 173–208.

[19] A. Rizaldi, M. Althoff, Formalising traffic rules for accountability of autonomous vehicles, in: 2015 IEEE 18th international conference on intelligent transportation systems, 2015.

[20] A. Rizaldi, F. Immler, M. Althoff, A formally verified checker of the safe distance traffic rules for autonomous vehicles, in: NASA Formal Methods: 8th International Symposium, 2016, pp. 175–190.

[21] A. Rizaldi, J. Keinholz, M. Huber, J. Feldle, F. Immler, M. Althoff, E. Hilgendorf, T. Nipkow, Formalizing and monitoring traffic rules for autonomous vehicles in Isabelle/HOL, in: Integrated Formal Methods: 13th International Conference, IFM 2017, 2017.

[22] C. Pek, P. Zahn, M. Althoff, Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules, in: 2017 IEEE Intelligent Vehicles Symposium (IV), 2017.

[23] B. Mirchevska, C. Pek, M. Werling, M. Althoff, J. Boedecker, High-level decision making for safe and reasonable autonomous lane-changing with reinforcement learning, in: 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 2156–2162.

[24] M. Treiber, A. Hennecke, D. Helbing, Congested traffic states in empirical observations and microscopic simulations, Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics 62 (2) (2000) 1805–1824.

[25] A. Kesting, M. Treiber, D. Helbing, General lane-changing model MOBIL for car-following models, Transportation Research Record 1999 (1) (2007) 86–94.

[26] E. C. for Europe, Vienna convention on road traffic, `http://www.unece.org/fileadmin/DAM/trans/conventn/crt1968e.pdf` (1968).