machines

MDPI

Article

Autonomous Vehicle Decision and Control through Reinforcement Learning with Traffic Flow Randomization

Yuan Lin, Antai Xie * and Xiao Liu

Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 510641, China; yuanlin@scut.edu.cn (Y.L.); 202121060431@mail.scut.edu.cn (X.L.)
* Correspondence: 202220159307@mail.scut.edu.cn

Abstract: Most of the current studies on autonomous vehicle decision-making and control based on reinforcement learning are conducted in simulated environments. The training and testing of these studies are carried out under the condition of rule-based microscopic traffic flow, with little consideration regarding migrating them to real or near-real environments. This may lead to performance degradation when the trained model is tested in more realistic traffic scenes. In this study, we propose a method to randomize the driving behavior of surrounding vehicles by randomizing certain parameters of the car-following and lane-changing models of rule-based microscopic traffic flow. We trained policies with deep reinforcement learning algorithms under the domain-randomized rule-based microscopic traffic flow in freeway and merging scenes and then tested them separately in rule-based and high-fidelity microscopic traffic flows. The results indicate that the policies trained under domain-randomized traffic flow have significantly better success rates and episodic rewards compared to those trained under non-randomized traffic flow.

Keywords: autonomous vehicle; reinforcement learning; decision and control; traffic flow; domain randomization

Citation: Lin, Y.; Xie, A.; Liu, X. Autonomous Vehicle Decision and Control through Reinforcement Learning with Traffic Flow Randomization. *Machines* **2024**, *12*, 264. https://doi.org/10.3390/machines12040264

Academic Editor: Yahui Liu

Received: 7 March 2024
Revised: 1 April 2024
Accepted: 15 April 2024
Published: 16 April 2024

1. Introduction

In recent years, autonomous vehicles have received increasing attention as they have the potential to free drivers from the fatigue of driving and facilitate efficient road traffic [1]. With the development of machine learning, rapid progress has been achieved in the development of autonomous vehicles. In particular, reinforcement learning, which enables vehicles to learn driving tasks through trial and error, continuously improves the learned policies. Compared to supervised learning, reinforcement learning does not require the manual labeling or supervision of sample data [2–5]. However, reinforcement learning models require tens of thousands of trial-and-error iterations for policy learning, and real vehicles on the road can hardly withstand so many trials. Therefore, the current mainstream research on autonomous driving with reinforcement learning focuses on using virtual driving simulators for training.

Lin et al. [6] utilized deep reinforcement learning within a driving simulator, Simulation of Urban Mobility (SUMO), to train autonomous vehicles, enabling them to merge safely and smoothly at on-ramps. Peng et al. [7] also employed deep reinforcement learning algorithms within a SUMO to train a model for lane changing and car following. They tested the model by reconstructing scenes using NGSIM data, and the results indicate that the models based on reinforcement learning demonstrate higher efficacy than those based on rule-based approaches. Mirchevska et al. [8] used fitted Q-learning for high-level decisionmaking on a busy simulated highway. However, the microscopic traffic flows of these studies are based on rule-based models, such as the Intelligent Driver Model (IDM) [9–11] and the Minimize Overall Braking Induced by Lane Change (MOBIL) model. These are mathematical models based on traffic flow theory [12]. They tend to simplify

vehicle motion behavior and do not consider the interaction of multiple vehicles. Autonomous vehicles trained with reinforcement learning in such microscopic traffic flows may perform exceptionally well when tested in the same environments. However, when the trained model is applied to more realistic or real-world traffic flows, their performance may significantly deteriorate, and they could even cause traffic accidents. This is due to the discrepancies between simulated and real-world traffic flows.

For research on sim-to-real transfer, numerous methods have been proposed to date. For instance, robust reinforcement learning has been explored to develop strategies that account for the mismatch between simulated and real-world scenes [13]. Meta-learning is another approach that seeks to learn adaptability to potential test tasks from multiple training tasks [14]. Additionally, the domain randomization method used in this article is acknowledged as one of the most extensively used techniques to improve the adaptability to real-world scenes [15]. Domain randomization relies on randomized parameters aimed at encompassing the true distribution of real-world data. Sheckells et al. [16] applied domain randomization to vehicle dynamics, using stochastic dynamic models to optimize the control strategies for vehicles maneuvering on elliptical tracks. Real-world experiments indicated that the strategy was able to maintain performance levels similar to those achieved in simulations. However, few studies have applied domain randomization to microscopic traffic flows and investigated its efficacy.

In recent years, many driving simulators have been moving towards more realistic scenes. One type includes data-based driving simulators (InterSim [17] and TrafficGen [18]), which train neural network models by extracting vehicle motion characteristics from real-world traffic datasets, resulting in interactive microscopic traffic flows. However, the simulation time is much longer than for most rule-based driving simulators due to the complexity of the models. The other kind includes theory-based interactive traffic simulators, which can generate long-term interactive high-fidelity traffic flows by combining multiple modules (LimSim [19]). The traffic flow generated by LimSim closely resembles an actual dataset with a normal distribution, sharing similar means and standard deviations [20].

This paper proposes a domain randomization method for rule-based microscopic traffic flows for reinforcement learning-based decision and control. The parameters of the car-following and lane-changing models are randomized with Gaussian distributions, making the microscopic traffic flows more random and behaviorally uncertain, thus exposing the agent to a more complex and variable driving environment during training. To investigate the impact of domain randomization, this paper will train and test agents using microscopic traffic flow without randomization, high-fidelity microscopic traffic flow, and domain-randomized traffic flow for freeway and merging scenes.

The rest of this paper is structured as follows: Section 2 introduces the relevant microscopic traffic flows. Section 3 describes the proposed domain randomization method. Section 4 presents the simulation experiments and the analysis of the results for the freeway and merging scenes. Finally, the conclusions are drawn in Section 5.

2. Microscopic Traffic Flow

Microscopic traffic flow models take individual vehicles as the research subject and mathematically describe the driving behaviors of the vehicles, such as acceleration, overtaking, and lane changing.

2.1. Rule-Based Microscopic Traffic Flow

This paper utilizes IDM and SL2015 as the default car-following and lane-changing models, respectively. The following is a detailed introduction to them.

2.1.1. IDM Car-Following Model

IDM was originally proposed by Treiber in [9], capable of describing various traffic states from free flow to complete congestion with a unified formulaic approach. The model takes the preceding vehicle's speed, the ego vehicle's speed, and the distance to the

preceding vehicle as inputs to output the ego vehicle's safe acceleration. The acceleration of the ego vehicle at each timestep is

$$\dot{v}(t) = a\left[1 - \left(\frac{v(t)}{v_0}\right)^{\delta} - \left(\frac{s^*(v(t), \Delta v(t))}{s}\right)^2\right], \tag{1}$$

where a represents the maximum acceleration of the ego vehicle, $v(t)$ is the current speed of the ego vehicle, v_0 is the desired speed of the ego vehicle, δ is the acceleration exponent, $\Delta v(t)$ is the speed difference between the ego vehicle and the preceding vehicle, s is the current distance between the ego vehicle and the preceding vehicle, and $s^*(v(t), \Delta v(t))$ is the desired following distance. The desired distance is defined as follows:

$$s^*(v(t), \Delta v(t)) = s_0 + \max\left(0, v(t) * T + \frac{v(t) * \Delta v(t)}{2\sqrt{ab}}\right), \tag{2}$$

where s_0 is the minimum gap, T is the bumper-to-bumper time gap, and b represents the maximum deceleration.

2.1.2. SL2015 Lane-Changing Model

The safety distance required for the lane-changing process is calculated as follows:

$$d_{\mathrm{lc,veh}}(t) = \begin{cases} v(t) * a_1 + 2l_{\mathrm{veh}}, & \text{if } v(t) \le v_{\mathrm{c}}, \\ v(t) * a_2 + 2l_{\mathrm{veh}}, & \text{if } v(t) > v_{\mathrm{c}}, \end{cases} \tag{3}$$

where $d_{\mathrm{lc,veh}}(t)$ denotes the safety distance required for lane changing, $v(t)$ represents the velocity of the vehicle at time t, l_{veh} is the length of the vehicle, a_1 and a_2 are safety factors, and the threshold speed v_{c} differentiates between urban roads and highways.

The profit $b_{ln}(t)$ at time t for changing lanes is calculated as follows:

$$b_{ln}(t) = \frac{v(t, ln) - v(t, lc)}{v_{\max}(lc)}, \tag{4}$$

where $v(t, ln)$ is the velocity of the vehicle in the target lane at the next timestep, $v(t, lc)$ is the safe velocity in the current lane, and $v_{\max}(lc)$ is the maximum velocity allowed in the current lane. The goal here is to maximize the velocity difference, thereby increasing the benefit of changing lanes.

If the profit $b_{ln}(t)$ for the current timestep is greater than zero, then this profit will be added to the cumulative profit. Conversely, if the profit for the current timestep is less than zero, the cumulative profit will be halved to moderate the desire to change to the target lane. If the cumulative profit is larger than a threshold, lane change can be initiated.

2.2. LimSim High-Fidelity Microscopic Traffic Flow

The study employs the LimSim driving simulation platform's high-fidelity microscopic traffic flow. The high-fidelity microscopic traffic flow in LimSim is based on optimal trajectory in the Frenet frame [21]. Within the circular area around the ego vehicle, the microscopic traffic flow is updated based on each optimal trajectory.

2.2.1. Trajectory Generation

In the Frenet coordinate system, the motion state of a vehicle can be described by the tuple $[s, \dot{s}, \ddot{s}, d, \dot{d}, \ddot{d}]$, where s represents the longitudinal displacement, $\dot{s}$ the longitudinal velocity, $\ddot{s}$ the lateral acceleration, d represents the lateral displacement, $\dot{d}$ the lateral velocity, and $\ddot{d}$ the latitudinal acceleration.

Lateral Trajectory Generation

The lateral trajectory curve can be expressed by the following fifth-order polynomial:

$$d(t) = a_{d0} + a_{d1}t + a_{d2}t^2 + a_{d3}t^3 + a_{d4}t^4 + a_{d5}t^5. \quad (5)$$

The trajectory start point is known as $D_0 = [d_0, \dot{d}_0, \ddot{d}_0]$, and a complete polynomial trajectory can be determined once the end point $D_1 = [d_1, \dot{d}_1, \ddot{d}_1]$ is specified. As vehicles travel on the road, they use the road centerline as the reference line for navigation, and the optimal state should be moving parallel to the centerline, which means the end point would be $D_1 = [d_1, 0, 0]$. Equidistant sampling points are selected between the start point and end point, and the multiple polynomial segments are connected to form many complete lateral trajectories.

Longitudinal Trajectory Generation

Longitudinal trajectory curve can be expressed with a fourth-degree polynomial:

$$s(t) = a_{s0} + a_{s1}t + a_{s2}t^2 + a_{s3}t^3 + a_{s4}t^4. \quad (6)$$

$S_0 = [s_0, \dot{s}_0, \ddot{s}_0]$ is the start point and $S_1 = [\dot{s}_1, \ddot{s}_1]$ is the end point. Equidistant sampling points are selected between the start point and end point, and the multiple polynomial segments are connected to form many complete longitudinal trajectories.

2.2.2. Optimal Trajectory Selection

The trajectory selection process involves evaluating a cost function that includes key components: trajectory smoothness, which is determined by the heading and curvature differences between the actual and reference trajectories; vehicle stability, indicated by the differences in acceleration and jerk between the actual and reference trajectories; collision risk, assessed by the risk level of collision with surrounding vehicles; speed deviation, gauged by the velocity difference between the actual trajectory and the reference speed; and lateral trajectory deviation, measured by the lateral distance difference between the actual trajectory and the reference trajectory.

The total cost function is utilized to evaluate the set of candidate trajectories in Section 2.2.1, followed by an assessment of their compliance with vehicle dynamics constraints, such as turning radius and speed/acceleration limits. The trajectory that not only satisfies the vehicle dynamics constraints but also incurs the minimum cost is selected as the final valid trajectory.

Vehicles within a 50 m perception range of the ego vehicle will be subject to the Frenet optimal trajectory control, with a trajectory being planned every 0.5 s and having a duration of 5 s.

3. Domain Randomization for Rule-Based Microscopic Traffic Flow

The domain randomization method is based on randomizing the model parameters in the IDM car-following model and the SL2015 lane-changing model. The randomized parameters are shown in Table 1 and are described below.

There are five randomized parameters in the IDM model. "δ" is the acceleration exponent and "T" is the time gap in the IDM model, respectively. "a_{max}", "a_{min}", and "v_{max}" are the upper and lower limits of vehicle acceleration and the upper limit of vehicle speed, respectively.

There are two randomized parameters in the SL2015 model. "`lcSpeedGain`" indicates the degree to which a vehicle is eager to change lanes to gain speed; the larger the value, the more inclined the vehicle is to change lanes. "`lcAssertive`" is another parameter that significantly influences the driver's lane-changing model [22]; a lower "`lcAssertive`" value makes the vehicle more inclined to accept smaller lane-changing gaps, leading to more aggressive lane-changing behavior.

Ref. [23] found that the parameters δ, T, a_{max}, a_{min}, and v_{max} are close to Gaussian distributions. Consequently, we adopt Gaussian distributions for all the domain-randomized parameters. All the randomized parameters follow Gaussian distributions within the interval $[s_{min}, s_{max}]$, with distribution $s\ (\mu, \sigma^2)$. Here, s_{max} and s_{min} are the upper and lower bounds of the randomization interval. μ is set to be $(s_{max} + s_{min})/2$, and σ is set to be $(s_{max} - s_{min})/6$. Thus, when a vehicle is generated, the probability that its randomized parameter value will fall within $[s_{min}, s_{max}]$ is 99.73%.

When each vehicle is initialized on the road for each episode, these randomized parameters are generated and assigned to it.

Table 1. Domain randomization parameters.

Parameter	Default Value	Randomization Interval
δ	4	$[3.5, 4.5]$
T	1 s	$[0.5, 1.5]$ s
a_{max}	2.6 m/s^2	$[1.8, 3.4]$ m/s^2
a_{min}	-4.5 m/s^2	$[-5.5, -3.5]$ m/s^2
v_{max}	8.33 m/s	$[7.33, 9.33]$ m/s
`lcSpeedGain`	1	$[0, 100]$
`lcAssertive`	1	$[1, 5]$

4. Simulation Experiment

In this section, we create freeway and merging environments in the open-source SUMO driving simulator [24] and establish the communication between SUMO and the reinforcement learning algorithm via TraCI [25]. The timestep for the agent to select actions and observe environment state is set at 0.1 s. We create non-randomized microscopic traffic flow, the high-fidelity microscopic traffic flow of LimSim, and the domain-randomized microscopic traffic flows. We train the reinforcement learning-based autonomous vehicles under different microscopic traffic flows in freeway and merging scenes, respectively.

4.1. Merging

4.1.1. Merging Environment

We establish the merging environment inspired by Lin et al. [6]. A control zone for the merging vehicle is established, spanning 100 m to the rear of the on-ramp's merging point and 100 m to the front of the merging point, as depicted in Figure 1. The red vehicle, operating under reinforcement learning control, is tasked with executing smooth and safe merging within the designated control area.

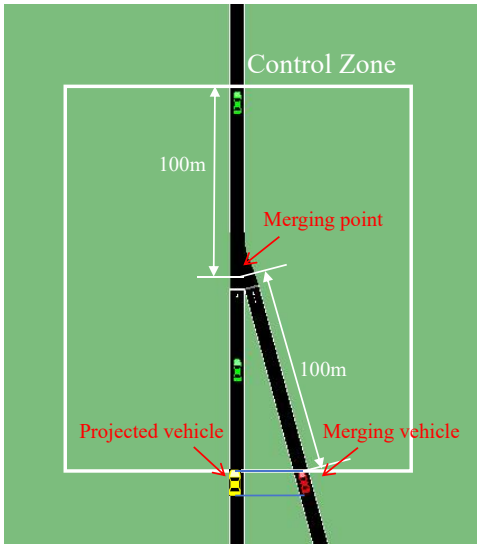

Figure 1. Merging in SUMO.

State

In defining the state of the reinforcement learning environment, the merging vehicle is projected onto the main road to produce the projected vehicle, and then a total of five vehicles are considered: two vehicles before the projected vehicle, two vehicles after the projected vehicle, and the projected vehicle. In order to utilize the observable information reasonably, the distance ($d_t^{p2}, d_t^{p1}, d_t^{f1}, d_t^{f2}, d_t^m$) of these five vehicles to the merging point, as well as their velocities ($v_t^{p2}, v_t^{p1}, v_t^{f1}, v_t^{f2}, v_t^m$), are included in the state representation. These parameters form a state representation with eleven variables, defined as

$$s_t = [d_t^{p2}, v_t^{p2}, d_t^{p1}, v_t^{p1}, d_t^m, v_t^m, a_t^m, d_t^{f1}, v_t^{f1}, d_t^{f2}, v_t^{f2}] \in S. \tag{7}$$

Action

The action space we have defined is a continuous variable: acceleration within $[-4.5, 2.5]$ m/s^2. This range is consistent with the normal acceleration range of surrounding vehicles.

$$a_t = \{acc_t^m\} \in A. \tag{8}$$

Reward

We aim for the merging vehicle to maintain a safe distance from the preceding and following vehicles, ensure comfort, and avoid coming to a stop or making the following vehicle brake sharply. Therefore, the reward function is expressed as follows:

$$R_{\text{total}} = R_m + R_b + R_j + R_{\text{stop}} + R_{\text{success}} + R_{\text{collision}}. \tag{9}$$

After merging, the merging vehicle is safer when its position is in the middle between the preceding and following vehicles. The corresponding penalizing reward is defined as

$$R_m = w_m * \left(|w| + \frac{\left| \frac{(v_{p1} + v_{f1})}{2} - v_m \right|}{\Delta v_{\max}} \right), \tag{10}$$

where w_m represents the weight factor, and $\Delta v_{\max}$ is the maximum allowable speed difference. The variable w is defined to measure the distance gap among the merging vehicle, its first preceding vehicle, and its first following vehicle. The details are as follows:

$$w = \frac{\left| d_{p1} - d_m - l_{p1} \right| - \left| d_m - d_{f1} - l_m \right|}{\left| d_{p1} - d_{f1} - l_{p1} - l_m \right|}, \tag{11}$$

where l_{p1} and l_m represent the lengths of the first preceding vehicle and the merging vehicle, both measuring at 5 m. When the first following vehicle performs braking in the control zone, a penalizing reward is defined as

$$R_b = w_b * \frac{\left| a_{f1} \right|}{\max(|a_{\min}|, a_{\max})}, \tag{12}$$

where w_b is the weight and a_{f1} is acceleration of the first following vehicle. In order to improve the comfort level of the merging vehicle, we define a penalizing reward for jerk:

$$R_j = w_j * \frac{|j_m|}{j_{\max}} = -w_j * \frac{|\dot{a}_m|}{j_{\max}}, \tag{13}$$

where w_j is the weight, $j_{\max}$ is maximum allowed jerk, and $\dot{a}_m$ is jerk of the merging vehicle.

In addition, if the merging vehicle comes to a stop, a penalty of $R_{\text{stop}} = -0.5$ is imposed. When a merged vehicle collides with any vehicle, a penalty of $R_{\text{collision}} = -1$ is applied. Conversely, if the merging vehicle successfully reaches its destination, a reward of $R_{\text{success}} = 1$ is granted. Table 2 shows the values of the above-mentioned parameters of the merging vehicle.

Table 2. Parameter values for the merging vehicle.

Parameter	Value
Weight for merging midway w_m	-0.015
Weight for penalizing first following's braking w_b	-0.015
Weight for penalizing jerk w_j	-0.015
Maximum allowed speed difference $\Delta v_{\max}$	5 m/s
Maximum allowed jerk value $j_{\max}$	3 m/s^3

4.1.2. Soft Actor–Critic

SAC is the reinforcement learning algorithm used for training in merging scenes [26]. The SAC algorithm uses the classical framework of reinforcement learning, actor–critic, which helps to optimize the value function and the policy at the same time, and it consists of a parameterized soft-Q function $Q_\theta(s_t, a_t)$ and a tractable policy $\pi_\phi(a_t|s_t)$. The parameters of these networks are θ and ϕ. This approach considers a more general maximum entropy objective that not only seeks to maximize rewards but also maintains a degree of randomness in action selection, as follows:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \gamma^t [r(s_t, a_t) + \alpha H(\pi(\cdot|s_t))], \tag{14}$$

where ρ_π denotes the state–action distribution under the policy π, while $H(\pi(\cdot|s_t))$ signifies the entropy of the policy at state s_t, thereby enhancing the unpredictability of the chosen actions. The temperature parameter α plays a pivotal role as it calibrates the balance between entropy and reward within the objective function, subsequently influencing the formulation of the optimal policy. The hyperparameters of SAC are the same as in Ref. [6].

4.1.3. Results under Different Microscopic Traffic Flows

Training

In the merging environment, we trained 200,000 timesteps in each of three different microscopic traffic flows. The training was carried out on an NVIDIA RTX 3060 graphics card paired with an Intel i7-12700F processor. It required approximately 1 h to complete the training using both SUMO's default non-randomized and domain-randomized traffic flows. In contrast, the training under the condition of high-fidelity traffic flow took 3.5 h. Vehicle generation probability is 0.56, and the traffic density on the main road was approximately 16 vehicles per kilometer.

Testing

The trained policy was tested with 1000 episodes in the merging environment. We evaluated the trained policy based on the merging vehicle's success rate defined by the completion of an episode without any collisions and the average reward value over the entire testing period.

Comparison and Analysis

The training curves depicted in Figure 2 suggest that there is minimal visible difference in the rate of convergence and the rewards achieved by strategies trained under different microscopic traffic flows.

Table 3 shows that the policy trained under rule-based traffic flow without randomization and high-fidelity microscopic traffic flow yields poor results when adapted to

domain-randomized rule-based traffic flow. Conversely, the policy trained under domain-randomized rule-based traffic flows consistently achieves success rates above 90% when tested across all three traffic flows.

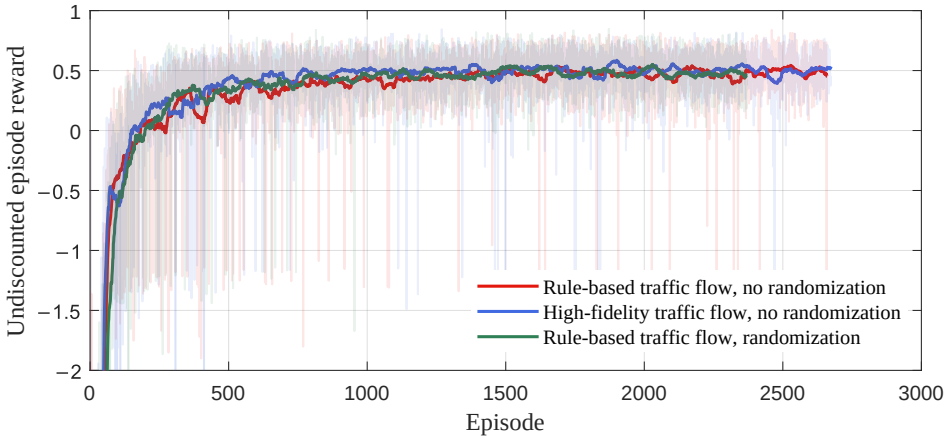

Figure 2. Undiscounted episode reward during training under three traffic flows.

Table 3. The results of testing the trained policies regarding merging.

			Traffic Flows for Training		
			Rule-Based, No Randomization	**High-Fidelity, No Randomization**	**Rule-Based, Randomization**
Testing	**rule-based, no randomization**	Average reward	0.0058	0.0002	−0.0018
		Success rate	98.50%	76.40%	91.60%
	high-fidelity, no randomization	Average reward	0.0029	0.0055	−0.0069
		Success rate	95.20%	97.50%	98.20%
	rule-based, randomization	Average reward	−0.0089	−0.0065	0.0057
		Success rate	56.00%	66.70%	99.30%

4.1.4. Generalization Results for Increased Traffic Densities

High-fidelity microscopic traffic flows closely resemble actual traffic scenes, so we used them as the test traffic flow with increased traffic densities. The impact of changes in traffic density is shown in Table 4. It can be observed that the policy trained under rule-based traffic flow without randomization experiences a gradual decline in success rates and rewards as traffic density increases. In contrast, the policy trained under domain-randomized rule-based traffic flow consistently maintains a higher success rate.

Table 4. The impact of traffic densities on three trained policies with high-fidelity traffic flow.

			Traffic Density for Testing under High-Fidelity Traffic Flow		
			$\phi = 0.56$	$\phi = 0.72$	$\phi = 0.89$
Training under rule-based traffic flow ($\phi = 0.56$)	**no randomization**	Average reward	0.0039	0.0017	0.0006
		Success rate	95.90%	93.30%	91.90%
	randomization	Average reward	−0.0001	−0.0002	−0.0001
		Success rate	98.20%	98.50%	98.20%

ϕ is the vehicle generation probability of the microscopic traffic flow, defined as the number of vehicles that are generated from the lane starting point per second.

4.1.5. Ablation Study

In order to strengthen the understanding of individual domain-randomized parameters' role in the model's performance, we analyzed their individual impact on the training

outcomes through an ablation study. For the ablation study, we separately ablated each of the domain-randomized parameters. Subsequently, policies were individually trained under the traffic flows with domain-randomized parameter ablation. Finally, the trained policies were tested under both the domain-randomized (all parameters randomized) and high-fidelity traffic flows. The results of the ablation study are shown in Table 5.

Table 5. The results of the ablation study.

Training under Rule-Based Traffic Flow		Traffic Flows for Testing	
		Rule-Based, Randomization	**High-Fidelity, No Randomization**
randomization—no δ	Average reward Success rate	0.0049 99.90%	−0.0023 95.60%
randomization—no T	Average reward Success rate	0.0018 92.50%	−0.0049 94.10%
randomization—no a_{max}	Average reward Success rate	0.0038 97.30%	−0.0038 97.70%
randomization—no a_{min}	Average reward Success rate	0.0025 94.20%	−0.0026 96.80%
randomization—no v_{max}	Average reward Success rate	−0.0070 65.00%	0.0016 93.70%
no randomization	Average reward Success rate	−0.0089 56.00%	0.0029 95.20%
randomization—all parameters	Average reward Success rate	0.0057 99.30%	−0.0069 98.20%

It can be observed that a decline occurs in the performance of the policies trained under the traffic flows with ablations when tested under the high-fidelity traffic flow. Moreover, the ablation of v_{max} significantly affects performance.

4.2. Freeway

4.2.1. Freeway Environment

We used a straight two-lane freeway measuring 1000 m in length, inspired by Lin et al. [27]. Figure 3 depicts a standard lane-changing scenario in SUMO, where the ego vehicle is indicated by the red car and the surrounding vehicles are represented by the green cars.

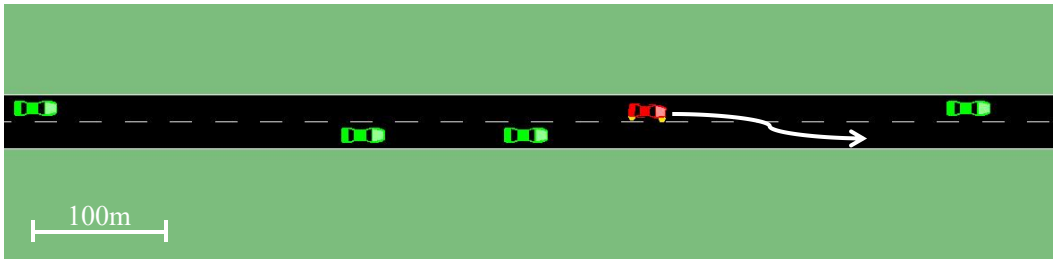

Figure 3. The ego vehicle overtakes along the arrow trajectory in the freeway.

State

The state of the environment is centered on the ego vehicle and four nearby vehicles: one directly in the front and one directly behind it in the same lane, and two similarly positioned vehicles in the adjacent lane. At time t, the state is defined by the longitudinal distance $(d_t^p, d_t^f, d_t^{adjacent_p}, d_t^{adjacent_f})$ of these four vehicles from the ego vehicle, their respec-

tive speeds ($v_t^p, v_t^f, v_t^{adjacent_p}, v_t^{adjacent_f}$), and the speed and acceleration (v_t^{ego}, a_t^{ego}) of the ego vehicle. These parameters form a state representation with ten variables as follows:

$$s_t = (d_t^p, d_t^f, d_t^{adjacent_p}, d_t^{adjacent_f}, v_t^p, v_t^f, v_t^{adjacent_p}, v_t^{adjacent_f}, v_t^{ego}, a_t^{ego}) \in S. \tag{15}$$

Action

The action space is defined as follows:

$$a_t = \{acc_t^{ego}, 0, 1\} \in A. \tag{16}$$

where acc_t^{ego} is a continuous action that indicates the acceleration of ego vehicle. Meanwhile, the discrete actions '0' and '1' dictate lane-changing behavior. The '0' means to keep the current lane and the '1' means an instantaneous lane change to the other lane.

Reward

We have formulated a reward function aligned with practical driving objectives, incentivizing behaviors such as avoiding collisions, obeying speed limits, preserving comfortable driving conditions, and maintaining a safe following distance. The total reward R_{total} is expressed as follows:

$$R_{\text{total}} = R_{\text{act}} + R_{\text{distance}} + R_{\text{jerk}} + R_v + R_{\text{collision}}. \tag{17}$$

In order to penalize frequent lane changes, the penalty R_{act} is defined as follows:

$$R_{\text{act}} = \begin{cases} \omega_0, & \text{if } |y_{t-1} - y_t| \neq 0 \text{ and } d_t^p < d_{\text{safe}}, \\ \omega_1, & \text{if } |y_{t-1} - y_t| \neq 0 \text{ and } d_t^p > d_{\text{safe}}, \\ 0, & \text{other}, \end{cases} \tag{18}$$

where y_t is ego vehicle's lateral position and $\omega_0 < \omega_1$. If the vehicle changes lanes within the safety distance d_{safe}, it incurs a penalty of ω_0. Alternatively, changing lanes outside d_{safe} results in a penalty of ω_1.

It is essential to ensure that the ego vehicle maintains a safe following distance from the preceding vehicle, and the corresponding penalizing reward R_{distance} is defined as

$$R_{\text{distance}} = \omega_2 * \left| \frac{d_t^p - d_{\text{safe}}}{d_{\text{safe}}} \right|, \text{if } d_t^p < d_{\text{safe}}. \tag{19}$$

The objective of R_{jerk} is to ensure driving comfort. It is defined as

$$R_{\text{jerk}} = \omega_3 * |(a_t - a_{t-1})/0.1|, \tag{20}$$

where a_t and a_{t-1} denote the acceleration at the current and previous moments, respectively.

In order to promote the ego vehicle speed that enables overtaking, the penalty R_v is defined as follows:

$$R_v = \begin{cases} \omega_4 * \left| \frac{v_t^{ego} - v_{\text{stable}}}{v_{\text{safe}}} \right|, & \text{if } v_{\text{stable}} < v_t^{ego} < v_{\text{safe}} \text{ and } d_t^p < d_{\text{safe}} + d^*, \\ \omega_5 * \left| \frac{v_t^{ego} - v_{\text{safe}}}{v_{\text{safe}}} \right|, & \text{if } v_t^{ego} > v_{\text{safe}} \text{ and } d_t^p < d_{\text{safe}} + d^*, \\ \omega_6 * \left| \frac{v_t^{ego} - v_{\text{stable}}}{v_{\text{stable}}} \right|, & \text{if } v_t^{ego} < v_{\text{stable}} \text{ and } d_t^p < d_{\text{safe}} + d^*, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

When there is no opportunity to overtake the vehicle ahead, the ego vehicle should travel at a steady speed similar to that of the preceding vehicle. Consequently, we introduce a threshold d^*. As long as $d^p \in [d_{\text{safe}}, d_{\text{safe}} + d^*]$, the ego vehicle will not incur penalties of R_{distance} or R_v.

In the equations presented above, ω_i denotes the corresponding weights. The key parameters for the freeway scene are presented in Table 6.

Table 6. Parameters for freeway simulation.

Parameters	Value	Weights	Value
a_{min}	-4.5 m/s^2	w_0	-5
a_{max}	2.6 m/s^2	w_1	-2
v_{safe}	16.89 m/s	w_2	-10
v_{stable}	8.89 m/s	w_3	-0.005
d_{safe}	25 m	w_4	1
$R_{\text{collision}}$	-200	w_5	-0.5
d^*	2.5 m	w_6	-0.5

4.2.2. Parameterized Soft Actor–Critic

The SAC algorithm in Section 4.1.2 can only solve continuous-action space problems. When dealing with continuous-discrete hybrid action space for freeway lane change, we adopt the Parameterized SAC (PASAC) algorithm, inspired by Lin et al. [27].

PASAC is based on SAC. The actor network produces continuous outputs, which include both continues actions and the weights for the discrete actions. An argmax function is utilized to select the discrete action associated with the maximum weight.

The freeway environment, having a hybrid continuous-discrete action space, requires the agent to be trained with the PASAC algorithm. The hyperparameters of PASAC are the same as SAC.

4.2.3. Results under Different Microscopic Traffic Flows

Training

In the freeway environment, we trained 400,000 timesteps in each of three different microscopic traffic flows. It required 1.5 h to complete the training using both rule-based traffic flows without randomization and with domain randomization. In contrast, the training under the condition of high-fidelity traffic flow took 5 h. Vehicle generation probability is 0.14 vehicles per second, and the traffic density on the main road was approximately 11 vehicles per kilometer on each straightaway.

Testing

The trained policy was tested with 1000 episodes in the freeway environment. We evaluated the trained policy based on the ego vehicle's success rate defined by the completion of an episode without any collisions, and the average reward value over the entire testing period.

Comparison and Analysis

In Figure 4, it can be observed that the policies all tend to converge around 200 episodes. Throughout the training process, aside from the initially lower reward of the domain-randomized traffic flows, the convergence rates and final rewards of the three curves are closely aligned.

The results of testing are shown in Table 7. It can be observed that the policy trained under domain-randomized rule-based traffic flows has the highest success rates when tested under different microscopic traffic flows. The policy trained under rule-based and high-fidelity traffic flows without randomization cannot adapt to domain-randomized rule-based traffic flow.

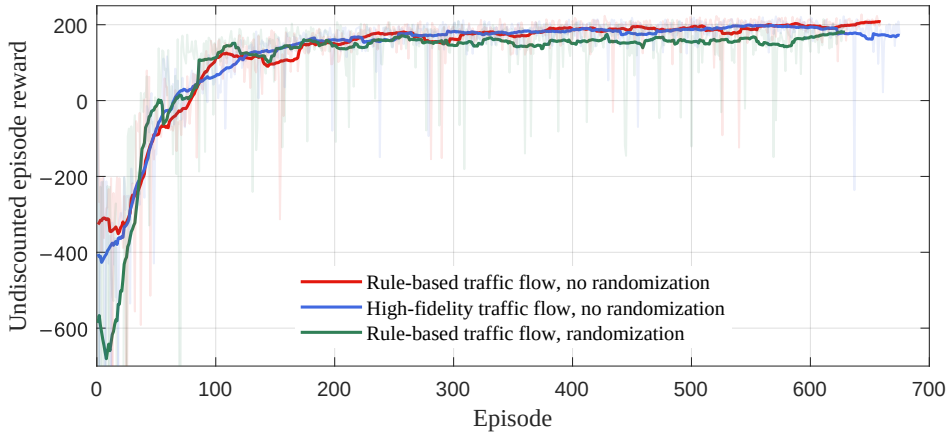

Figure 4. Undiscounted episode reward during training under three traffic flows.

Table 7. The results of testing the trained policies regarding freeway condition.

			Traffic Flows for Training		
			Rule-Based, No Randomization	**High-Fidelity, No Randomization**	**Rule-Based, Randomization**
Testing	**rule-based, no randomization**	Average reward	200.50	205.32	197.15
		Success rate	100%	99.70%	100%
	high-fidelity, no randomization	Average reward	187.10	208.85	202.48
		Success rate	98.90%	99.60%	99.90%
	rule-based, randomization	Average reward	126.27	160.06	186.72
		Success rate	80.40%	89.00%	99.40%

4.2.4. Generalization Results for Increased Traffic Densities

The impact of changes in high-fidelity traffic density is shown in Table 8. It can be observed that the success rate and reward of the policy trained under microscopic traffic flow without randomization decreases significantly as traffic density increases. In contrast, the policy trained under domain-randomized traffic flow maintains a success rate near 100%.

Table 8. The impact of changes in traffic density on three trained policies under high-fidelity traffic flow.

			Traffic Density for Testing under High-Fidelity Traffic Flow		
			$\phi = 0.14$	$\phi = 0.18$	$\phi = 0.20$
Training under rule-based traffic flow ($\phi = 0.14$)	**no randomization**	Average speed	16.39	15.89	15.77
		Average reward	187.10	189.85	109.49
		Success rate	98.90%	93.90%	66.50%
	randomization	Average speed	15.83	15.42	15.17
		Average reward	202.48	197.48	191.51
		Success rate	99.90%	99.80%	99.90%

ϕ is the vehicle generation probability of the microscopic traffic flow, defined as the number of vehicles that are generated from the lane starting point per second.

4.2.5. Ablation Study

In the freeway environment, we also conducted an ablation study to enhance our understanding of the role that an individual domain-randomized parameter plays in the model's performance. The results of the ablation study are shown in Table 9.

Table 9. The results of the ablation study.

Training under Rule-Based Traffic Flow		Traffic Flows for Testing	
		Rule-Based, Randomization	**High-Fidelity, No Randomization**
randomization—no δ	Average reward	170.46	187.65
	Success rate	97.20%	99.90%
randomization—no T	Average reward	169.62	178.95
	Success rate	98.50%	99.60%
randomization—no a_{max}	Average reward	181.49	202.52
	Success rate	98.30%	99.90%
randomization—no a_{min}	Average reward	170.72	185.81
	Success rate	99.40%	100%
randomization—no v_{max}	Average reward	96.99	150.10
	Success rate	89.90%	99.50%
randomization—no `lcSpeedGain`	Average reward	132.04	156.61
	Success rate	100%	100%
randomization—no `lcAssertive`	Average reward	173.71	188.57
	Success rate	97.70%	99.90%
no randomization	Average reward	126.27	187.10
	Success rate	80.40%	98.90%
randomization—all parameters	Average reward	186.72	202.48
	Success rate	99.40%	99.90%

In the freeway environment, the collision rates of the different policies are close to zero, so we mainly compare the average rewards of the different policies. It can be found that the conclusions are similar to those of merging, where the performance of the policies trained under traffic flows with domain-randomized parameter ablation declines to varying degrees. The ablation of v_{max} has a large impact on the performance.

5. Conclusions

In this study, we introduce a method for randomizing lane-changing and car-following model parameters to generate randomized microscopic traffic flows, and we evaluate and compare the policies trained by reinforcement learning algorithms in freeway and merging environments. The results show that

- The policy trained under the condition of domain-randomized rule-based microscopic traffic flow is able to maintain high rewards and success rates when tested with different microscopic traffic flows. However, the policy trained under the condition of microscopic traffic flow without randomization or high-fidelity microtraffic flow performs significantly worse when tested under microscopic traffic flows that are different from those of training. This indicates that domain randomization enables reinforcement learning agents to adapt to different types of traffic flow.
- The policy trained under the condition of domain-randomized rule-based microscopic traffic flow performs well when tested under high-fidelity microscopic traffic flow with different traffic densities. The policy trained under microscopic traffic flow without randomization decreases significantly with increasing traffic density. This indicates that the domain-randomized traffic flow possesses strong generalization to changes in traffic density.
- Although high-fidelity microscopic traffic flow is close to real microscopic traffic flows, the results show that not only does it considerably increase simulation time but policies trained under the condition of microscopic traffic flow also do not generalize well to different microscopic flows. Therefore, high-fidelity microscopic traffic flow is more suitable for testing rather than training.

In summary, the policies trained under domain-randomized rule-based microscopic traffic flow demonstrate robust performance when transferred to environments that closely resemble real-world traffic conditions. The future work includes testing a policy trained under the condition of domain-randomized rule-based microscopic traffic flow on a real autonomous vehicle.

Author Contributions: Conceptualization, Y.L.; methodology, A.X., Y.L. and X.L.; formal analysis, A.X. and Y.L.; investigation, A.X. and Y.L.; data curation, A.X.; writing—original draft preparation, A.X.; writing—review and editing, A.X.; supervision, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Guangzhou Basic and Applied Basic Research Program under Grant 2023A04J1688, and in part by South China University of Technology faculty start-up fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data can be obtained upon reasonable request from the corresponding author.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Le Vine, S.; Zolfaghari, A.; Polak, J. Autonomous cars: The tension between occupant experience and intersection capacity. *Transp. Res. Part C Emerg. Technol.* **2015**, *52*, 1–14. [CrossRef]
2. Sallab, A.E.; Abdou, M.; Perot, E.; Yogamani, S. Deep reinforcement learning framework for autonomous driving. *arXiv* **2017**, arXiv:1704.02532.
3. Hoel, C.J.; Wolff, K.; Laine, L. Automated speed and lane change decision making using deep reinforcement learning. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2148–2155.
4. Ye, Y.; Zhang, X.; Sun, J. Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment. *Transp. Res. Part C Emerg. Technol.* **2019**, *107*, 155–170. [CrossRef]
5. Ye, F.; Cheng, X.; Wang, P.; Chan, C.Y.; Zhang, J. Automated lane change strategy using proximal policy optimization-based deep reinforcement learning. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1746–1752.
6. Lin, Y.; McPhee, J.; Azad, N.L. Anti-jerk on-ramp merging using deep reinforcement learning. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 7–14.
7. Peng, J.; Zhang, S.; Zhou, Y.; Li, Z. An Integrated Model for Autonomous Speed and Lane Change Decision-Making Based on Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 21848–21860. [CrossRef]
8. Mirchevska, B.; Blum, M.; Louis, L.; Boedecker, J.; Werling, M. Reinforcement learning for autonomous maneuvering in highway scenarios. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium, Los Angeles, CA, USA, 11–14 June 2017.
9. Treiber, M.; Hennecke, A.; Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* **2000**, *62*, 1805. [CrossRef] [PubMed]
10. Liu, J.; Zeng, W.; Urtasun, R.; Yumer, E. Deep structured reactive planning. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 4897–4904.
11. Huang, X.; Rosman, G.; Jasour, A.; McGill, S.G.; Leonard, J.J.; Williams, B.C. TIP: Task-informed motion prediction for intelligent vehicles. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 11432–11439.
12. Punzo, V.; Simonelli, F. Analysis and comparison of microscopic traffic flow models with real traffic microscopic data. *Transp. Res. Rec.* **2005**, *1934*, 53–63. [CrossRef]
13. Tessler, C.; Efroni, Y.; Mannor, S. Action robust reinforcement learning and applications in continuous control. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6215–6224.
14. Wang, J.X.; Kurth-Nelson, Z.; Tirumala, D.; Soyer, H.; Leibo, J.Z.; Munos, R.; Blundell, C.; Kumaran, D.; Botvinick, M. Learning to reinforcement learn. *arXiv* **2016**, arXiv:1611.05763.
15. Andrychowicz, O.M.; Baker, B.; Chociej, M.; Jozefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **2020**, *39*, 3–20. [CrossRef]
16. Sheckells, M.; Garimella, G.; Mishra, S.; Kobilarov, M. Using data-driven domain randomization to transfer robust control policies to mobile robots. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3224–3230.

17. Sun, Q.; Huang, X.; Williams, B.C.; Zhao, H. InterSim: Interactive traffic simulation via explicit relation modeling. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 11416–11423.
18. Feng, L.; Li, Q.; Peng, Z.; Tan, S.; Zhou, B. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 3567–3575.
19. Wenl, L.; Fu, D.; Mao, S.; Cai, P.; Dou, M.; Li, Y.; Qiao, Y. LimSim: A long-term interactive multi-scenario traffic simulator. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 24–28 September 2023; pp. 1255–1262.
20. Zheng, O.; Abdel-Aty, M.; Yue, L.; Abdelraouf, A.; Wang, Z.; Mahmoud, N. CitySim: A drone-based vehicle trajectory dataset for safety oriented research and digital twins. *arXiv* **2022**, arXiv:2208.11036.
21. Werling, M.; Ziegler, J.; Kammel, S.; Thrun, S. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, Alaska, 3–8 May 2010; pp. 987–993.
22. Berrazouane, M.; Tong, K.; Solmaz, S.; Kiers, M.; Erhart, J. Analysis and initial observations on varying penetration rates of automated vehicles in mixed traffic flow utilizing sumo. In Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), Graz, Austria, 4–8 November 2019; pp. 1–7.
23. Kusari, A.; Li, P.; Yang, H.; Punshi, N.; Rasulis, M.; Bogard, S.; LeBlanc, D.J. Enhancing SUMO simulator for simulation based testing and validation of autonomous vehicles. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 5–9 June 2022; pp. 829–835.
24. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent development and applications of SUMO-Simulation of Urban MObility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138.
25. Wegener, A.; Piórkowski, M.; Raya, M.; Hellbrück, H.; Fischer, S.; Hubaux, J.P. TraCI: An interface for coupling road traffic and network simulators. In Proceedings of the 11th Communications and Networking Simulation Symposium, Ottawa, ON, Canada, 14–17 April 2008; pp. 155–163.
26. Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. Soft actor-critic algorithms and applications. *arXiv* **2018**, arXiv:1812.05905.
27. Lin, Y.; Liu, X.; Zheng, Z.; Wang, L. Discretionary Lane-Change Decision and Control via Parameterized Soft Actor-Critic for Hybrid Action Space. *arXiv* **2024**, arXiv:2402.15790.