sensors

MDPI

Article

Enhanced Safety in Autonomous Driving: Integrating a Latent State Diffusion Model for End-to-End Navigation

De-Tian Chu [1,†], Lin-Yuan Bai [1,†], Jia-Nuo Huang [2], Zhen-Long Fang [3], Peng Zhang [1], Wei Kang [1] and Hai-Feng Ling [1,*]

[1] Field Engineering College, Army Engineering University of PLA, Nanjing 210007, China; detian_chu@aeu.edu.cn (D.-T.C.); linyuan_bai@aeu.edu.cn (L.-Y.B.); shuiyuantou2519@163.com (P.Z.); w_kang2023@163.com (W.K.)

[2] School of Computing and Data Science, Xiamen University Malaysia, Sepang 43900, Malaysia; jasm4041@163.com

[3] School of Mathematics and Computer Sciences, Nanchang University, Nanchang 330031, China; zhenlongfang0808@163.com

[*] Correspondence: hfling@compintell.cn

[†] These authors contributed equally to this work.

Abstract: Ensuring safety in autonomous driving is crucial for effective motion planning and navigation. However, most end-to-end planning methodologies lack sufficient safety measures. This study tackles this issue by formulating the control optimization problem in autonomous driving as Constrained Markov Decision Processes (CMDPs). We introduce an innovative, model-based approach for policy optimization, employing a conditional Value-at-Risk (VaR)-based soft actor-critic (SAC) to handle constraints in complex, high-dimensional state spaces. Our method features a worst-case actor to ensure strict compliance with safety requirements, even in unpredictable scenarios. The policy optimization leverages the augmented Lagrangian method and leverages latent diffusion models to forecast and simulate future trajectories. This dual strategy ensures safe navigation through environments and enhances policy performance by incorporating distribution modeling to address environmental uncertainties. Empirical evaluations conducted in both simulated and real environments demonstrate that our approach surpasses existing methods in terms of safety, efficiency, and decision-making capabilities.

Keywords: end-to-end driving; safe navigation; motion planning

check for updates

Citation: Chu, D.-T.; Bai, L.-Y.; Huang, J.-N.; Fang, Z.-L.; Zhang, P.; Kang, W.; Ling, H.-F. Enhanced Safety in Autonomous Driving: Integrating a Latent State Diffusion Model for End-to-End Navigation. *Sensors* **2024**, *24*, 5514. https://doi.org/10.3390/s24175514

Academic Editor: Wenling Li

Received: 28 June 2024
Revised: 13 August 2024
Accepted: 19 August 2024
Published: 26 August 2024

1. Introduction

In the rapidly evolving field of autonomous driving, ensuring vehicle safety during the exploration phase is critical [1,2]. Traditional end-to-end methods often fail to guarantee safety in complex, high-dimensional environments [3,4]. The increased complexity of such scenarios makes sampling and learning inefficient and hinders the pursuit of globally optimal policies. Inadequate safety measures can lead to severe consequences, including system damage and significant threats to human life.

Reinforcement learning (RL) has achieved remarkable success across various domains [5,6]. Deep learning (DL) excels in perception, while RL is proficient in decision-making. The integration of DL and RL, known as deep reinforcement learning (DRL), addresses decision-making in complex obstacle avoidance scenarios. Unlike traditional motion planning methods, DRL can enhance adaptability and generalization across diverse scenarios, overcoming the limitations of conventional approaches and providing a more efficient and effective solution. Mnih et al. proposed a Deep Q-Network (DQN) model that combines convolutional neural networks and Q-learning from traditional RL to address high-dimensional perception-based decision problems [6]. This widely adopted approach serves as a primary driver for deep RL. DRL equips robots with perceptual and decision-making capabilities by processing input data to generate the output in an end-to-end manner [7]. The end-to-end motion planning approach treats the system holistically, enhancing its robustness [8,9]. Moreover, DRL can manage high-dimensional and nonlinear environments by utilizing neural networks to learn intricate state and action spaces [10,11]. In contrast, traditional heuristic algorithms require a manual design of the state and action spaces and may need rules to be redesigned for new scenarios, leading to algorithm limitations and performance bottlenecks. Previous research in RL and DL has shown progress in addressing complex scenarios, whereas traditional methods are hindered by their dependence on manual design for state and action spaces, resulting in adaptability issues and reduced performance in novel situations.

Recognizing these challenges, various methods have been proposed. One such method is the framework of Constrained Markov Decision Processes (CMDPs), which seeks to balance reward maximization and risk mitigation by optimizing the trade-off between exploration and exploitation [12,13]. Building on prior research, this study redefines the control optimization problem within the CMDP framework. We propose an innovative, model-based policy optimization framework that integrates the Augmented Lagrangian method, and latent diffusion models to effectively manage safety constraints in autonomous navigation tasks while optimizing navigation proficiency.

Our methodology begins with a latent variable model designed to produce extended-horizon trajectories, enhancing our system's ability to predict future states accurately. To further strengthen the safety, we incorporate a conditional Value-at-Risk (VaR) within the soft actor-critic (SAC) framework, ensuring safety constraints are met through the Augmented Lagrangian method for efficiently solving the safety-constrained optimization challenges.

For extreme risk scenarios, our approach includes a worst-case scenario planning method that employs a "worst-case actor" during policy exploration to ensure the safest outcomes in potentially dangerous conditions. To enhance performance, we integrate a latent diffusion model for state representation learning, refining our system's ability to interpret complex environmental data.

The efficacy of our proposed approach is validated through extensive experiments. Our empirical findings verify our approach's capability to manage and mitigate risks effectively in high-dimensional state spaces, enhancing the safety and reliability of autonomous vehicles in complex driving environments.

Our main contributions are as follows.

- We integrate latent diffusion models for state representation learning, enabling the forecasting of future observations, rewards, and actions. This capability allows for the simulation of future trajectories within the model framework, facilitating the proactive assessment of rewards and risks through controlled model roll-outs.
- We further extend our approach to include advanced prediction of future state value distributions, incorporating the estimation of worst-case scenarios. This ensures that our model predicts and prepares for potential adverse conditions, enhancing system reliability.
- Our experimental results demonstrate the efficacy of our proposed approaches in simulated and real-world environments, which can also guarantee safe policy exploration in unpredictable scenarios.

2. Related Work

2.1. Safe Reinforcement Learning

Safe reinforcement learning (Safe RL) integrates safety measures with the standard learning process for agents in complex environments [14]. Developing safe and reliable autonomous driving systems necessitates a comprehensive focus on safety throughout control and decision-making processes. Safe RL is a powerful tool for training controllers and planners to navigate dynamic environments while adhering to safety constraints.

Safe RL algorithms tackle the safety challenge through various methods. Examples include Constrained Policy Optimization (CPO) [8], which maximizes policies within clearly defined safety constraints to avoid unsafe actions. Trust Region Policy Optimization (TRPO) [15] and Proximal Policy Optimization (PPO) [16] refine these algorithms to integrate safety considerations by implementing penalty terms or barrier functions. Gaussian Processes (GPs) model [17] environmental uncertainty, enabling safe exploration by quantifying the risk associated with different actions. Model Predictive Control (MPC)-based safe RL [18,19] predicts future states and maximizes actions over a finite horizon while complying with safety constraints.

Safe RL plays a critical role in ensuring the safety of autonomous driving. Its applications include high-level decision-making tasks such as lane changing [13] and route planning [20], factoring in the behavior of surrounding vehicles and potential conflicts. Low-level control merges safe RL with conventional methods like MPC, ensuring adherence to safety constraints while following the planned path [21,22]. Combining RL with rule-based systems that encode traffic laws ensures that learned policies comply with rules. Safety verification techniques provide assurances that learned policies maintain safety in the presence of uncertainties and dynamic obstacles [23,24].

The variety of safe RL algorithms and their applications highlight their crucial role in developing adaptable and secure control systems. Ongoing advancements in these approaches will be essential for the successful real-world deployment of autonomous vehicles.

2.2. Reinforcement Learning with Latent State

Latent dynamic models, pivotal in modeling time-series data within reinforcement learning, enable a deep understanding of concealed states and dynamic changes in intricate environments [7,25,26]. These models capture crucial relationships between unobservable internal states and observed data, substantially enhancing the predictive capabilities of RL agents.

In the context of RL, latent dynamic models operate under probabilistic frameworks, utilizing Bayesian inference or maximum likelihood estimation to accurately deduce both model parameters and hidden states [27,28]. These frameworks facilitate environment modeling that aligns with the probabilistic nature of real-world dynamics.

Mathematically, latent dynamic models are characterized by the following:

- **State Transition Equation:**

  $$s_{t+1} = f(s_t, a_t, \epsilon_t)$$

  where s_t denotes the latent state at time t; a_t signifies the action taken; ϵ_t represents the stochasticity inherent in the environment. The function f may be deterministic or stochastic, encapsulating the uncertainty of state transitions.
- **Observation Equation:**

  $$o_t = g(s_t, \delta_t)$$

  where o_t represents the observed output linked to the hidden state s_t; δ_t signifies the observation noise, linking theoretical models to real-world observations.
- **Reward Function:**

  $$r_t = R(s_t, a_t)$$

  defines the immediate reward received after executing action a_t in state s_t, essential for policy optimization in RL.

The primary objective of using latent dynamic models in RL is to infer the sequence of hidden states $s_1, s_2, \ldots, s_T$ from observations $o_1, o_2, \ldots, o_T$, actions $a_1, a_2, \ldots, a_T$, and rewards $r_1, r_2, \ldots, r_T$. This modeling approach predicts future states or actions and integrates environmental dynamics to enhance decision-making processes and optimize policy outcomes [11,29,30]. Additionally, these models provide a deeper understanding of environmental complexities, enabling RL agents to make more informed and effective decisions [31,32].

2.3. Diffusion-Model-Based Reinforcement Learning

Recent advances in diffusion models have significantly impacted RL, especially offline RL, where interaction with the environment is limited. Originally successful in generative tasks, diffusion models are adapted to address challenges such as distributional shift and extrapolation errors in offline settings [33–35].

Research [36] indicates that diffusion probabilistic models can effectively generate plausible future states and actions, facilitating robust policy learning from static datasets. Additionally, latent diffusion models (LDMs) reduce computational demands and enhance learning efficiency by encoding trajectories into a compact latent space before diffusion, thus capturing complex decision dynamics [37]. These approaches improve the stability and efficacy of Q-learning algorithms by ensuring that generated actions remain within the behavioral policy's support, mitigating the risk of policy deviation due to poor sampling [38].

Integrating diffusion techniques into RL frameworks represents a promising frontier for developing more capable and reliable autonomous systems, particularly in environments where conventional learning approaches are inadequate; however, how to improve safety is the main key problem.

3. Problem Modeling

Autonomous navigation involves modeling the interaction between an autonomous agent and a dynamic, uncertain environment using a finite-horizon Markov Decision Process (MDP), denoted by the tuple $\mathcal{M} \sim (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, r, \gamma)$. Here, $\mathcal{S} \subset \mathbb{R}^n$ represents a high-dimensional continuous state space, and $\mathcal{A} \subset \mathbb{R}^m$ represents the action space. State transitions, defined by $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$, capture the environment's stochastic characteristics.

Observations $\mathcal{O}$, derived from the state space, are high-dimensional images captured by sensors and analyzed via a latent diffusion model to refine state representation. This detailed comprehension of environmental dynamics is vital for navigating complex scenarios. The reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ and the discount factor $\gamma \in [0, 1]$ inform the agent's policy π_θ, which produces actions based on analyzed observations o_t.

We prioritize safety by incorporating a conditional VaR metric into our decision-making framework, addressing worst-case scenarios through latent state diffusion for robust state representation learning. Safety protocols are formalized with a subset $S_u \subset \mathbb{R}^n$, where entering a state $s_t \in S_u$ signifies a potential safety breach, monitored by a safety function κ. The objective extends beyond maximizing cumulative rewards to minimizing safety violations.

$$\max_\theta J(\pi_\theta) = \mathbb{E}_{\mathcal{P}(\cdot|s_t, a_t)}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, s_{t+1})\right], \text{ s.t. }, \sum_{t=0}^{T} \kappa(s_t) \leq D, a_t \sim \pi_\theta(\cdot \mid o_t) \tag{1}$$

where $\kappa(s_t) \in \{0, 1\}$ indicates safety violations, with $D \in \mathbb{R}$ representing the maximum allowable safety violations, aiming for $D \to 0$ to enhance operational safety. This safety-constrained MDP framework enables the agent to learn navigation policies that maximize efficiency while ensuring safety, effectively balancing high performance with adherence to critical safety constraints.

4. Methodology

4.1. Constrained Markov Decision Process Formulation

Reinforcement learning involves continuous interaction between the agent and the environment. This study focuses on the safety of autonomous driving, requiring a trade-off between reward and safety. We formulate the control optimization problem as a Constraint Markov Decision Process (CMDP), defined by $(S, A, p, r, c, d, \gamma)$, including the state space, action space, transition model, reward, cost, constraint, and discount factor. At each step, the agent receives a reward r and cost c. The optimization objective is to maximize the reward subject to the safety constraint in Equation (2).

$$\begin{aligned} &\max_{\pi} \underset{(s_t,a_t)\sim\tau_\pi}{\mathrm{E}}\left[\sum_t \gamma^t r(s_t,a_t)\right] \\ &\text{s.t.} \underset{(s_t,a_t)\sim\tau_\pi}{\mathrm{E}}\left[\sum_t \gamma^t c(s_t,a_t)\right] \le d \end{aligned} \tag{2}$$

where π and τ_π denote the policy and the trajectory distribution of the policy, respectively. Current CMDP methods, while effective, often struggle with the high-dimensional, nonlinear state and action spaces inherent in complex traffic environments. Moreover, it lacks a more generalizable safety constraint. Therefore, an improved method is introduced in the following sections.

4.2. Build the Latent Diffusion Model for State Representation

In this section, we elaborate on the latent state space representation and the diffusion process employed in our Enhanced Safe Navigation Autonomous Driving (ESAD-LEND) model.

As shown in Figure 1, the model begins by capturing the state (S) and reward (r) from the environment based on the actions taken by the autonomous driving agent. This information is then encoded into a latent state (z) using a Variational Autoencoder (VAE). The VAE consists of an encoder and a decoder, where the encoder transforms the input state into a latent representation, and the decoder attempts to reconstruct the original input from this latent space. The latent state space is depicted in the top right section of Figure 1, where the encoder encodes the observed state into the latent representation z. This representation z is then combined with the current state (S) to inform the policy network $\pi(a|S,z)$, which generates the appropriate action (a) to be taken by the autonomous vehicle. The purpose of using a VAE is to ensure that the latent space captures the essential features of the input state while allowing for effective reconstruction, thereby creating a robust and informative latent representation.

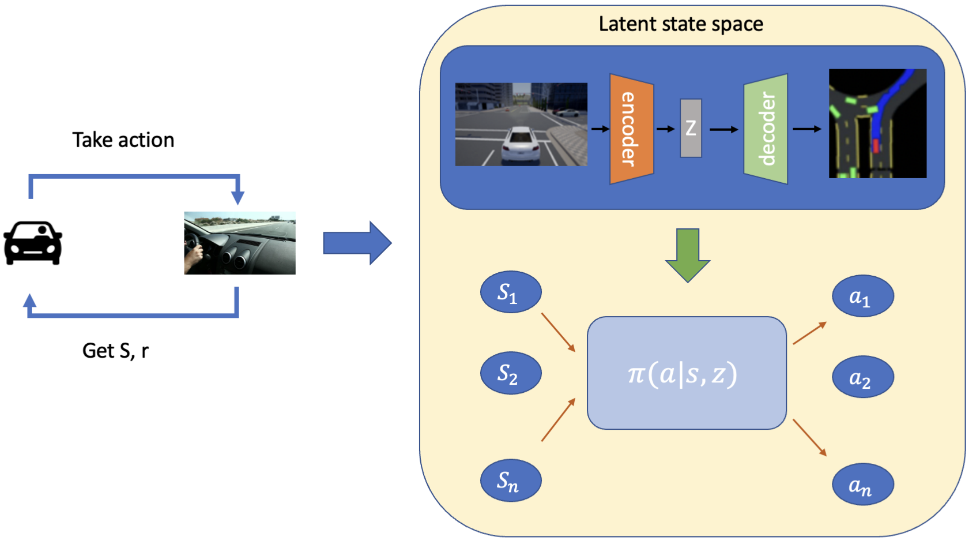


Figure 1. Our proposed framework, the Enhanced Safe Navigation Autonomous Driving model with Latent State End-to-Navigation Diffusion model (ESAD-LEND).

Furthermore, as shown in Figure 2, the training process for the diffusion-based model is illustrated. The latent state z_0, derived from the VAE encoder in Figure 1, undergoes a forward diffusion process. This process gradually adds noise to the latent state, resulting in a highly noisy latent state z_T. The noisy latent state z_T is then passed through a denoising network, which iteratively denoises it over multiple steps to recover the original latent state z_0. This iterative denoising process ensures that the latent space is robust to noise and can effectively capture the essential features required for safe navigation. The forward diffusion and subsequent denoising make the latent representation resilient to variations and disturbances, enhancing the model's performance in real-world scenarios.

The new method proposed in this paper integrates the latent model, as illustrated in Figure 2, containing three critical components: a representation model, a transition model, and a reward model. We consider the latent state as extracted and depicted in Figure 1. These models are trained to work in synergy for the accurate prediction and navigation in complex environments, utilizing both observed data and imagined trajectories.

- **Representation Model:** The representation model establishes a robust latent space based on past experiences. The representation model is formalized as $p(s_\tau \mid s_{\tau-1}, a_{\tau-1}, o_\tau)$, predicting the next state by integrating information from the current state, action, and observation. The representation loss is quantified by assessing the accuracy of state and reward predictions.
- **Transition Model:** This model outputs a Gaussian distribution, defined as $q(s_\tau \mid s_{\tau-1}, a_{\tau-1})$. The transition model's accuracy is evaluated using the Kullback–Leibler (KL) divergence between the predicted and actual distributions, signifying the latent imagination and the environment's real response, respectively.
- **Reward Model:** The reward model enhances learning by computing expected rewards based on the current state, $q(r_\tau \mid s_\tau)$. This model is crucial for the agent to enhance actions and maximize environmental returns.

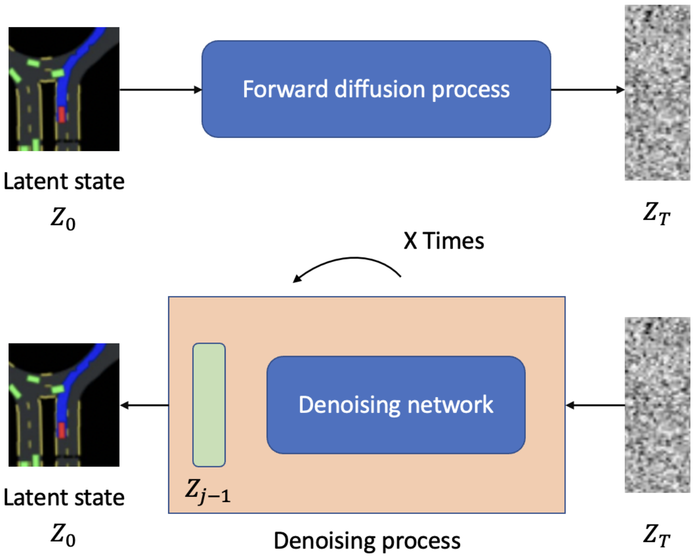


Figure 2. The training process for a diffusion-based model.

In our framework, p denotes the distribution from environment interactions, while q represents latent space predictions. Time steps in the latent space are indexed by τ. Our model's core innovation lies in its ability to create and refine a latent imagination space for predicting future trajectories. This enables the agent to explore safely and learn optimal behaviors by iteratively refining the latent space to avoid unsafe states and maximize efficiency.

4.3. Build Safety Guarantee

Figure 3 illustrates the policy generation process within a diffusion-based control system. The process begins with the state of the environment, input into a diffusion model denoted as $P(z|s)$. This model generates multiple candidate latent states, representing potential actions or decision paths.

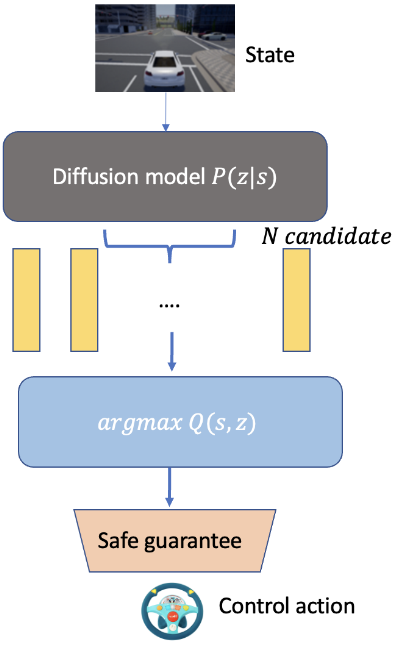


Figure 3. The policy generation process in a diffusion-based control system.

These candidates are evaluated using a Q-function, $\arg\max Q(s,z)$, which chooses the optimal latent state for action execution by maximizing expected rewards and ensuring compliance with safety standards. The chosen action is optimal in performance and satisfies predefined safety criteria, ensuring adherence to safety standards before execution.

This structured approach guarantees robust safety by integrating performance optimization and rigorous safety compliance, which is crucial for autonomous systems operating in dynamic and uncertain environments.

A similar approach can be found in latent imagination, such as Dreamer [39], which achieved excellent performance. However, Dreamer did not consider safety constraints. Although Dreamer reached high reward targets, the absence of safety measures could lead to irreversible incidents in pursuit of higher rewards. Therefore, it is crucial to introduce safety constraints to balance reward and risk.

In our latent imagination framework, we leverage distributional reinforcement learning to address the safety-constrained RL problem [40]. Instead of following a policy π to obtain an action value, distributional RL focuses on the expectation of value and cost [41]. We concentrate on the actor-critic policy and develop a model-based algorithm incorporating latent imagination. The soft actor-critic (SAC) [42] introduces maximum entropy to balance exploitation and exploration.

$$\pi^* = \underset{\pi}{\text{argmax}} \sum_{t=0}^{T} E_{(s_t,a_t)\sim\tau_\pi}\left[\gamma^t(r(s_t,a_t) + \beta\mathcal{H}(\pi(. \mid s_t))\right] \tag{3}$$

where π^* denotes the optimal policy, and β represents the stochasticity of π.

Agents can learn the optimal policy without safety concerns; however, irreversible situations such as collisions are unacceptable in autonomous driving. We address these safety constraints by formulating them using a Lagrangian method with constraints.

$$\begin{aligned} &\max_{\pi} \mathop{\mathbb{E}}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_t \gamma^t r(s_t,a_t)\right] \\ &\text{s.t.}\begin{cases} \mathop{\mathbb{E}}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_t \gamma^t c(s_t,a_t)\right] \leq d \\ \mathop{\mathbb{E}}_{(s_t,a_t)\sim\rho_\pi}\left[-\log(\pi_t(a_t \mid s_t))\right] \geq \mathcal{H}_0 \quad \forall t \\ h(s_t) \leq 0 \quad \text{(Safety Constraint)} \\ h(s_{t+1}) \leq (1-\alpha)h(s_t) \quad \text{(Control Barrier Function)} \end{cases} \end{aligned} \tag{4}$$

To enhance the utilization of safety constraints, a barrier function is employed in distributional RL to adjust the risk assessment, allowing the agent to determine optimal exploration and conservatism strategies. In this formulation, for a constraint with relative degree m, the generalized control barrier function is defined as follows:

$$h(s_{t+m}) \leq (1-\alpha)h(s_t) \tag{5}$$

For a specified risk level α, we optimize the policy until Γ_π meets the following condition:

$$\begin{gathered} \Gamma_\pi(s,a,\alpha) \doteq CVaR_\alpha = Q_\pi^c(s,a) + \alpha^{-1}\phi\left(\Phi^{-1}(\alpha)\right)\sqrt{V_\pi^c(s,a)} \\ \Gamma_\pi(s_{t+m},a_{t+m},\alpha) \leq (1-\alpha)\Gamma_\pi(s_t,a_t,\alpha) \end{gathered} \tag{6}$$

where α signifies the conservativeness coefficient.

Next, the standard soft actor-critic (SAC) framework is enhanced by integrating safety constraints using a barrier function, which adjusts risk assessment and balances exploration with conservatism.

4.4. VaR-Based Soft Actor-Critic for Safe Exploration

As shown in Figure 4, it illustrates the process of world model learning within our autonomous system, which is pivotal for the development and refinement of our latent imagination model. Following Figure 3, the *world model learning process* involves iterative updates to the representation, transition, and reward models. The transition model, in particular, is used to predict future states and rewards, which are essential for planning and decision-making.

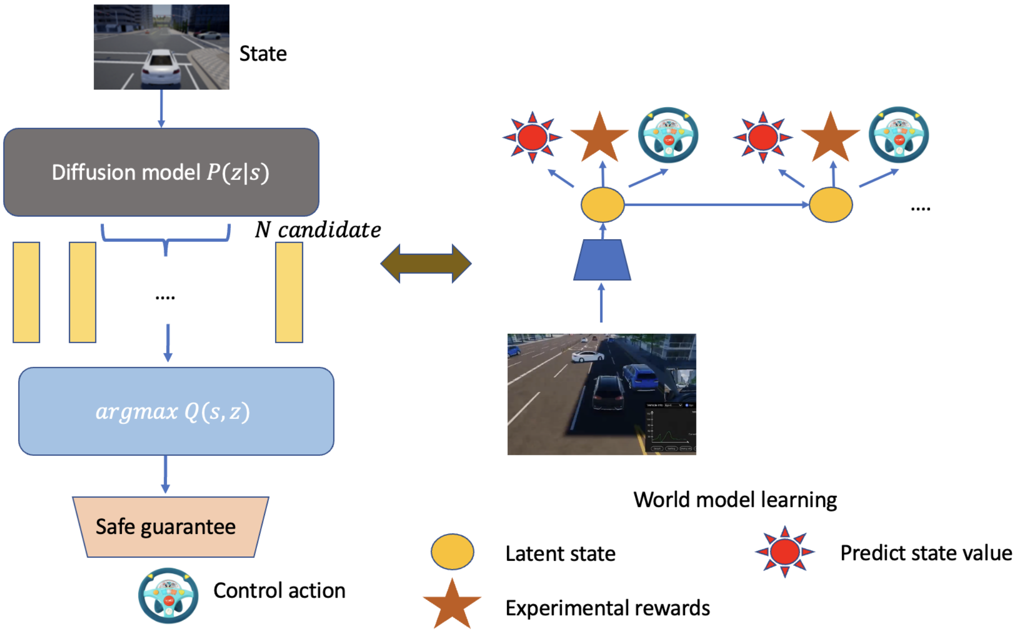


Figure 4. The world model learning process in our autonomous system.

As depicted in Algorithm 1, we construct the latent imagination through iterative updates of the representation, transition, and reward models. Future trajectories are predicted using the transition model, serving as the basis for the latent imagination in continuous actor-critic policy optimization. Upon defining a specific risk-awareness level, new safety constraints are introduced. By considering the expected cumulative cost and reward, the model is updated using distributional RL. Following iterative learning within the latent imagination over discrete time steps, the agent interacts with the environment to acquire new reward and cost data.

Algorithm 1 Pseudocode for ESAD-LEND

```
Inputs: Initial parameters $\alpha, \psi, \mu, \eta, \theta$, and $\tau$
Initialize: Target networks: $\langle \bar{\psi}, \bar{\mu}, \bar{\eta} \rangle \leftarrow \langle \psi, \mu, eta \rangle$
Initialize: Dataset $\mathcal{D}$ with $S$ random seed episodes.
while not onverged do
    for update step $t = 1 \dots T$ do
        // World Model learning with Latent Diffusion Representation
        Sample a sequence batch of $\{(o_t, a_t, r_t, c_t)\}_{t=k}^{k+L}$ from $\mathcal{D}$.
        Encode observed states to latent space: $z_t \sim q_\theta(z_t \mid o_t, h_{t-1})$ using a latent diffusion model.
        Predict next state and reward using latent representations: $h_t = f_\theta(h_{t-1}, z_t, a_{t-1})$.
        Optimize $\theta$ by minimizing the equation $\mathcal{L}(\theta)$.
        // Behavior learning
        Imagine trajectories $\{(s_\tau, a_\tau)\}_{\tau=t}^{t+H}$ from each $s_t$.
        Compute safety measure $\Gamma_\pi(s, a, \alpha)$ based on $\Gamma_\pi(s, a, \alpha) \doteq CVaR_\alpha = Q_\pi^c(s, a) + \alpha^{-1}\phi(\Phi^{-1}(\alpha))\sqrt{V_\pi^c(s, a)}$
        $\psi_i \leftarrow \psi_i - \lambda_R \hat{\nabla}_{\psi_i} J_R(\psi_i)$ for $i \in \{1, 2\}$
        $\theta \leftarrow \theta - \lambda_\pi \hat{\nabla}_\theta J_\pi(\theta)$
        $\mu, \eta, \beta, \kappa \leftarrow \mu - \lambda_C \hat{\nabla}_\mu J_C(\mu), \eta - \lambda_V \hat{\nabla}_\eta J_V(\eta), \beta - \lambda_\beta \hat{\nabla}_\beta J_e(\beta), \kappa - \lambda_\kappa \hat{\nabla}_\kappa J_s(\kappa)$
        // Update target network weights
        $\bar{\psi}_i \leftarrow \tau\psi_i + (1 - \tau)\bar{\psi}_i$ for $i \in \{1, 2\}$
        $\bar{\mu}, \bar{\eta} \leftarrow \tau\mu + (1 - \tau)\bar{\mu}, \tau\eta + (1 - \tau)\bar{\eta}$
    end for
    // Environment interaction
    $o_1 \leftarrow$ env · reset ()
    for time step $t = 1 \dots T$ do
        Update state $s_t$ using latent diffusion model: $s_t \sim p_\theta(s_t \mid s_{t-1}, a_{t-1}, z_t)$.
        Compute action $a_t \sim \pi_\theta(a_t \mid s_t)$ using the action model.
        Add exploration noise to action.
        $r_t, o_{t+1} \leftarrow$ env · step$(a_t)$.
    end for
    Add experience to dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{(o_t, a_t, r_t)_{t=1}^{T}\right\}$.
end while
```

5. Experiments

5.1. Environmental Setup

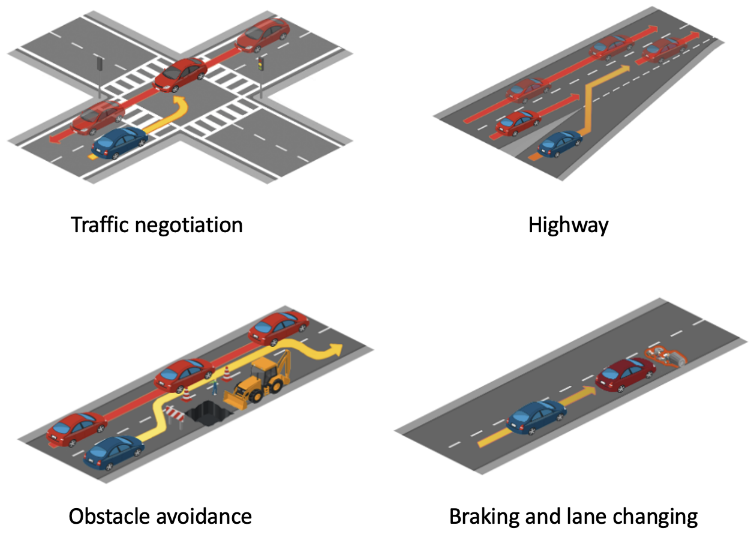
Experimental Setup in CARLA Simulator

In our study, we employed the CARLA simulator to construct and evaluate various safety-critical scenarios challenging the response capabilities of autonomous driving systems. CARLA offers a comprehensive, open-source environment specifically designed for autonomous driving research, featuring realistic urban simulations.

Specific scenario

We designed specific scenarios in CARLA to evaluate various aspects of autonomous vehicle behavior, as illustrated in Figure 5. These scenarios included the following:

- **Traffic Negotiation:** Multiple vehicles interact at a complex intersection, testing the vehicle's ability to negotiate right-of-way and avoid collisions.
- **Highway:** Simulates high-speed driving conditions with lane changes and merges, assessing the vehicle's decision-making speed and accuracy.
- **Obstacle Avoidance:** Challenges the vehicle to detect and navigate around sudden obstacles such as roadblocks.
- **Braking and Lane Changing:** Tests the vehicle's response to emergency braking scenarios and rapid lane changes to evade potential hazards.


Traffic negotiation

Highway

Obstacle avoidance

Braking and lane changing

Figure 5. Illustration of various safety-critical scenarios developed to assess the response capabilities of autonomous driving systems.

These scenarios are essential for validating the robustness and reliability of safety protocols in autonomous vehicles across diverse urban conditions.

Urban Driving Environments

Additionally, we tested the vehicles across three distinct urban layouts in CARLA, as depicted in Figure 6. These towns were selected based on the following criteria:

- **Town 6:** Features a typical urban grid that simplifies navigation while testing adherence to basic traffic rules.
- **Town 7:** Incorporates winding roads and a central water feature, introducing complexity to navigation tasks and necessitating advanced path planning.
- **Town 10:** Represents a dense urban environment with numerous intersections and limited maneuvering space, ideal for testing advanced navigation strategies.

The comprehensive simulation environments offered by CARLA, coupled with the designed scenarios, facilitate thorough testing of autonomous driving algorithms, ensuring their safe and efficient operation in real-world conditions.

Considering the generalizability of our model, we selected Town 10 for its urban complexity and Town 7 for its natural landscape as the primary training environment. In random scenarios, vehicles, including the ego vehicle, can appear at any location on the map. In fixed scenarios, vehicle appearances are limited to a predefined range. However, both scenarios adhere to CARLA's randomization protocols in each training and evaluation episode.

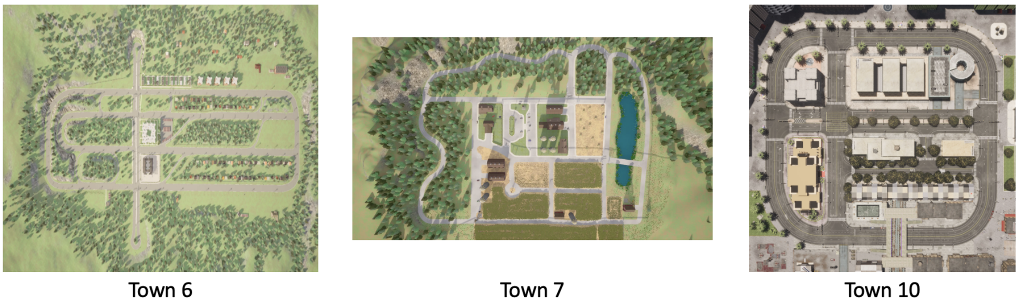


Figure 6. Aerial views of urban environments in the CARLA simulator: Town 6, Town 7, and Town 10. Town 6 features a typical urban grid with straightforward navigation challenges; Town 7 includes winding roads and a central water feature, introducing complexity to navigation tasks; and Town 10 offers a dense urban environment with numerous intersections and limited maneuvering space.

Real-world scenario construction

As illustrated in Figure 7, these experiments evaluated the autonomous system's ability to detect, negotiate, and navigate around obstacles. The setups ranged from simple configurations with minimal obstacles to intricate scenarios featuring densely packed obstacles, assessing the system's adaptability to dynamically changing and physically constrained environments. The robot is named "Ackerman/Differential ROS car robot" and is controlled by a Jetson Nano running ROS. It includes a depth camera for visual perception, but not for speech interaction. The laser rangefinder (Lidar M10P, Riegl) has a measurement radius of 30 m, a scanning frequency of 12 Hz, and a sampling frequency of 20,000 Hz. The output contains angle and distance information. The robot is driven by brushless motors and uses a 360° scanning range to measure distance. Despite transmission delays, our experiments revealed that these delays had minimal impact on the overall performance metrics, including the collision rate and navigation efficiency.

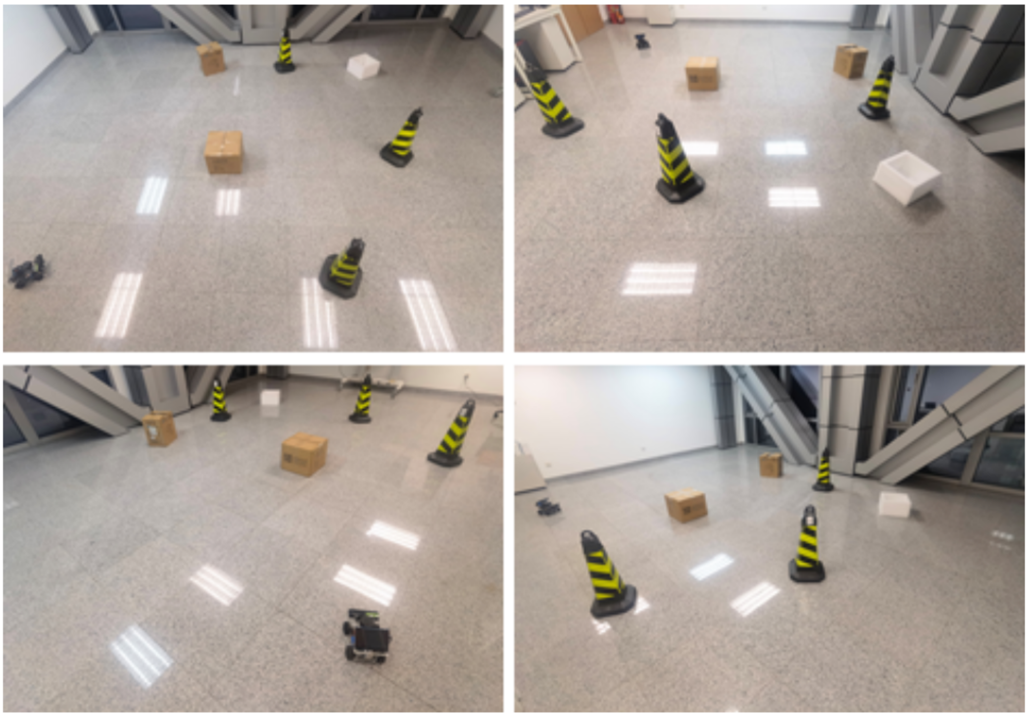


Figure 7. Real-world setups testing obstacle avoidance in autonomous systems.

The main objective of these real-world tests was to validate the robustness and reliability of the navigation algorithms and their associated safety mechanisms. Through these tests, our goal was to ensure that the autonomous systems could effectively detect and avoid immediate physical obstacles while maintaining adherence to safety standards across diverse environmental conditions.

5.2. Design of the Reward Function

Our autonomous system utilized a complex reward function aimed at optimizing navigation efficiency and safety. The reward function is segmented into multiple components that collectively ensure the vehicle adheres to operational standards:

5.2.1. Velocity Compliance Reward (R_v)

This reward is awarded for maintaining a specified target velocity, encouraging efficient transit and fuel economy:

$$R_\text{v} = \begin{cases} 1 & \text{if } v_\text{current} = v_\text{target} \\ \frac{1}{1+\lambda\left|v_\text{current}-v_\text{target}\right|} & \text{otherwise} \end{cases} \tag{7}$$

where v_current represents the vehicle's current velocity, v_target denotes the target velocity, and λ is a parameter that penalizes deviations from this target.

5.2.2. Lane Maintenance Reward (R_l)

This reward incentivizes the vehicle to stay within the designated driving lane:

$$R_\text{l} = \begin{cases} 1 & \text{if } d_\text{offset} = 0 \\ -1 & \text{if } d_\text{offset} > d_\text{max} \\ \frac{d_\text{max}-d_\text{offset}}{d_\text{max}} & \text{otherwise} \end{cases} \tag{8}$$

where d_offset denotes the lateral displacement from the lane center, and d_max represents the threshold beyond which penalties are applied.

5.2.3. Orientation Alignment Reward (R_o)

This component imposes penalties on the vehicle for incorrect heading angles:

$$R_\text{o} = \frac{1}{1+\mu|\theta_\text{current}-\theta_\text{ideal}|} \tag{9}$$

where θ_current represents the vehicle's current orientation, θ_ideal denotes the ideal orientation along the road, and μ is a constant that determines the strictness of the alignment requirement.

5.2.4. Exploration Incentive Reward (R_e)

An innovative component introduced to promote the exploration of less-traveled paths, thereby enhancing the robustness of the navigation strategy:

$$R_\text{e} = \exp(-\nu \cdot n_\text{visits}) \tag{10}$$

where n_visits denotes the count of times a specific path or region has been traversed, and ν is a decay factor that diminishes the reward with repeated visits.

5.2.5. Composite Reward Calculation

The overall reward (R_total) is a composite measure, defined as follows:

$$R_\text{total} = \omega_\text{v} \cdot R_\text{v} + \omega_\text{l} \cdot R_\text{l} + \omega_\text{o} \cdot R_\text{o} + \omega_\text{e} \cdot R_\text{e} \tag{11}$$

where ω_v, ω_l, ω_o, and ω_e are weights that prioritize different aspects of the reward structure according to strategic objectives.

5.3. Evaluation Metrics

Inspired by [43], to rigorously assess the performance of autonomous driving systems in our simulation, a comprehensive set of metrics is utilized, which encompasses the various facets of driving quality, including safety, efficiency, and adherence to rules. The metrics are defined as follows:

1. **Route Completion (RC):** This metric quantifies the percentage of each route completed by the agent without intervention. It is defined as follows:

$$RC = \frac{1}{N}\sum_{i=1}^{N} R_i \times 100\% \tag{12}$$

   where R_i represents the completion rate for the i-th route. A penalty applies if the agent deviates from the designated route, reducing RC proportionally to the off-route distance.

2. **Infraction Score (IS):** Capturing the cumulative effect of driving infractions, this score uses a geometric series, with each infraction type assigned a specific penalty coefficient:

$$IS = \prod_{j=\{\text{Ped, Veh, Stat, Red}\}} (p_j)^{\#\text{infractions}_j} \tag{13}$$

   Coefficients are set as $p_{\text{Ped}} = 0.50$, $p_{\text{Veh}} = 0.60$, $p_{\text{Stat}} = 0.65$, and $p_{\text{Red}} = 0.70$ for infractions involving pedestrians, vehicles, static objects, and red lights, respectively.

3. **Driving Score (DS):** This primary evaluation metric combines route completion with infraction penalties:

$$DS = \frac{1}{N}\sum_{i=1}^{N} R_i \times P_i \tag{14}$$

   where P_i is the penalty multiplier for infractions on the i-th route.

4. **Collision Occurrences (COs):** This metric quantifies the frequency of collisions during autonomous driving, providing a key measure of the safety and reliability of the driving algorithm. A lower CO value indicates better collision avoidance, which is critical for the safe operation of autonomous vehicles. This metric is defined as follows:

$$CO = \frac{\text{Number of Collisions}}{\text{Total Distance Driven}} \times 100 \tag{15}$$

   where the Number of Collisions denotes the total count of collisions encountered by the autonomous vehicle, and the Total Distance Driven signifies the total distance covered by the vehicle during testing. This metric is expressed as a percentage to standardize the measure across various distances driven.

5. **Infractions per Kilometer (IPK):** This metric normalizes the number of infractions by the distance driven, providing a measure of infractions per unit distance:

$$IPK = \frac{\sum_{i=1}^{N} I_i}{\sum_{i=1}^{N} K_i} \tag{16}$$

   where I_i denotes the number of infractions on the i-th route, and K_i represents the distance driven on the i-th route.

6. **Time to Collision (TTC):** This metric estimates the time remaining before a collision occurs, assuming the current velocity and trajectory of the vehicle and any object or vehicle in its path remain unchanged. It critically measures the vehicle's ability to detect and react to potential hazards in its immediate environment:

$$TTC = \min\left(\frac{d}{v_{\text{rel}}}\right) \tag{17}$$

where d represents the distance to the nearest object in the vehicle's path, and v_{rel} is the relative velocity towards the object. A lower TTC value signifies a higher immediate risk, necessitating more urgent responses from the system.

7. **Collision Rate (CR):** This metric quantifies the frequency of collisions during autonomous operation:

$$CR = \frac{\text{Number of Collisions}}{\text{Total Distance Driven}} \tag{18}$$

Expressed as collisions per kilometer, this metric evaluates the efficacy of collision avoidance systems integrated into autonomous driving algorithms.

These metrics collectively provide a robust framework for evaluating autonomous driving systems under varied driving conditions, thus facilitating a detailed analysis of their capability to navigate complex urban environments while adhering to traffic rules and maintaining high safety standards.

5.4. Baseline Setup

- Dreamer [39]: A reinforcement learning agent designed to tackle long-horizon tasks using latent imagination in learned world models. It distinguishes itself by employing deep learning to process high-dimensional sensory inputs and learn intricate behaviors.
- LatentSW-PPO [43]: Wang et al introduced a novel RL framework for autonomous driving that enhances safety and efficiency. This framework integrates a latent dynamic model that captures environmental dynamics from bird's-eye view images, thereby improving learning efficiency and mitigating safety risks through synthetic data generation. Additionally, it incorporates state-wise safety constraints using a barrier function to ensure safety at every state during the learning process.
- Diffuser [35]: Janner et al proposed a novel approach to model-based reinforcement learning that integrates trajectory optimization into the modeling process, addressing the empirical shortcomings of traditional methods. They utilize a diffusion probabilistic model to plan by iteratively denoising trajectories, making sampling and planning nearly identical. In contrast to their proposed model, we further enhanced it with safety considerations.

Additionally, we propose several ablation versions of our method to evaluate the performance of each sub-module.

- Safe Autonomous Driving with Latent End-to-end Navigation (SAD-LEN): This version excludes the latent diffusion component, relying solely on traditional latent state representation, the primary focus of which is on evaluating the impact of the latent state representation on navigation performance without the enhancements provided by diffusion processes.
- Autonomous Driving with End-to-end Navigation and Diffusion (AD-END): This version removes the safety guarantee mechanisms, focusing on the integration of end-to-end navigation with diffusion models. It aims to assess the contribution of safety constraints to overall performance and safety.

6. Results and Analysis

6.1. Evaluating Prediction Performance

The accuracy of future scene generation is indeed crucial, as it directly impacts the safety and appropriateness of the actions taken. As shown in Figure 8, we can see that our model has better prediction and interoperability for surrounding agents. From the attention map, it indicates it can align with our logic.

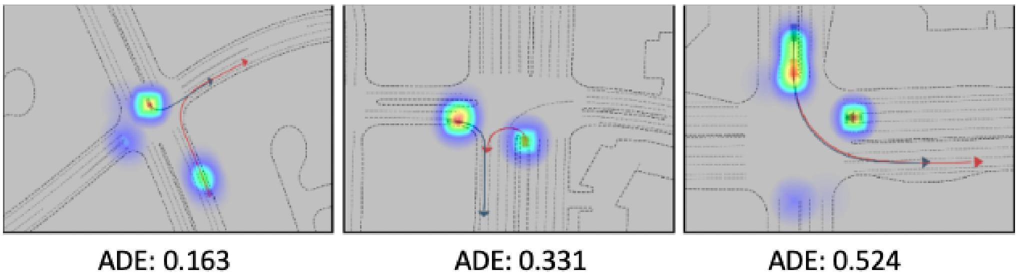


Figure 8. Visualization of the prediction for surrounding agents.

6.2. Evaluating Safety and Efficiency During Exploratory

We first evaluate the performance of the scenarios depicted in Figure 9. The primary evaluation criteria are derived from the metrics mentioned above, with a focus on the Average Displacement Error (ADE), which measures the average distance between the predicted trajectory and the actual trajectory of surrounding agents. Overall, the visualizations demonstrate that the model is capable of generating reasonable and accurate predictions across different scenarios.

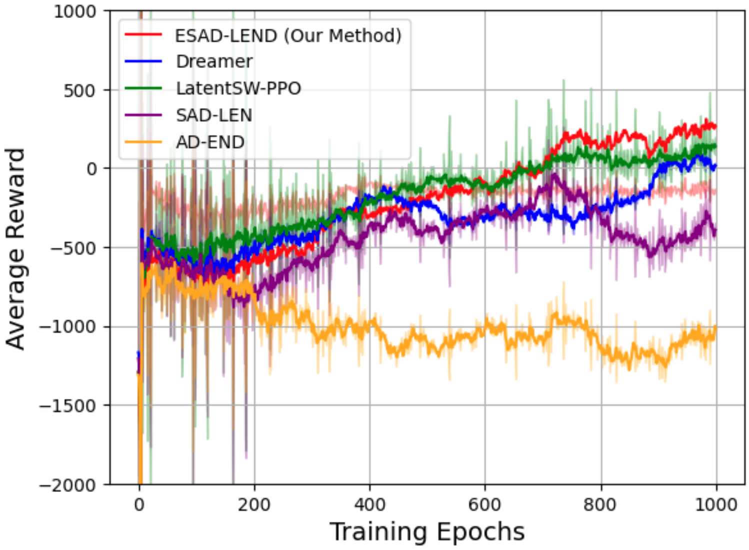


Figure 9. Comparative performance of various reinforcement learning methods over 1000 training epochs. The plot showcases the average reward trajectories for ESAD-LEND (our method), Dreamer, LatentSW-PPO, SAD-LEN, and AD-END.

In our comprehensive evaluation of autonomous driving systems, as shown in Table 1, ESAD-LEND demonstrated superior performance across multiple metrics in a simulated testing environment. ESAD-LEND achieved the highest DS (91.2%) and RC (98.3%), surpassing all comparative methods, including the baseline SAC, which had a DS of 78.2% and RC of 90.1%. Additionally, ESAD-LEND exhibited remarkable compliance and safety, recording the lowest IS (0.5%), indicative of fewer traffic violations and enhanced adherence to safety protocols. Its operational efficiency was also superior, with the lowest scores in CR and TTC, suggesting reduced incidences and smoother operational flow. Furthermore, its ability to handle unexpected obstructions was demonstrated by the lowest CO score (0.5%), highlighting its robustness and adaptability in dynamic environments. Diffuser also showed strong performance, with a DS of 87.6% and RC of 96.4%, and it recorded the lowest IS (0.3%), further emphasizing its reliability and effectiveness. These results establish ESAD-LEND and Diffuser as leading approaches in AI-driven transportation, providing safe, efficient, and compliant autonomous driving experiences.

Table 1. Enhanced driving performance and infraction analysis in the testing environment with variance indicators.

Metric	ESAD-LEND	Dreamer	LatentSW-PPO	SAD-LEN	AD-END	SAC	Diffuser
DS (%)	91.2 ± 1.5	91.7 ± 2.1	86.5 ± 2.4	83.3 ± 2.7	80.4 ± 2.9	78.2 ± 3.1	87.6 ± 2.3
RC (%)	98.3 ± 0.5	97.2 ± 0.7	95.8 ± 0.9	94.1 ± 1.1	92.0 ± 1.3	90.1 ± 1.6	96.4 ± 0.8
IS (%)	0.5 ± 0.1	0.7 ± 0.2	0.9 ± 0.2	1.1 ± 0.3	0.4 ± 0.3	1.6 ± 0.4	0.9 ± 0.3
CO (%)	0.5 ± 0.1	0.8 ± 0.2	1.0 ± 0.2	1.3 ± 0.3	1.6 ± 0.4	1.9 ± 0.5	0.7 ± 0.2
CR (%)	0.4 ± 0.1	0.6 ± 0.2	0.8 ± 0.2	1.0 ± 0.3	1.3 ± 0.4	1.5 ± 0.5	0.6 ± 0.2
TTC (%)	0.4 ± 0.1	0.6 ± 0.2	0.8 ± 0.2	1.0 ± 0.3	1.3 ± 0.4	1.5 ± 0.5	0.6 ± 0.2

6.3. Evaluate Generalization Ability

We conducted two additional experiments, one using a map from CARLA Table 2, and another in a real-world environment. (Table 3).

Table 2. Enhanced driving performance and infraction analysis in the testing environment in CARLA with variance indicators.

Metric	ESAD-LEND	Dreamer	LatentSW-PPO	SAD-LEN	AD-END	SAC	Diffuser
DS (%)	95.3 ± 1.2	87.4 ± 2.6	84.1 ± 3.0	80.5 ± 3.5	78.9 ± 3.8	76.8 ± 4.1	85.6 ± 5.5
RC (%)	99.2 ± 0.3	96.5 ± 1.1	94.3 ± 1.4	92.1 ± 1.7	89.8 ± 2.0	87.6 ± 2.4	96.4 ± 0.9
IS (%)	0.4 ± 0.05	0.7 ± 0.1	1.0 ± 0.15	1.2 ± 0.18	1.5 ± 0.22	1.8 ± 0.25	0.3 ± 0.1
CO (%)	0.2 ± 0.03	0.6 ± 0.09	0.9 ± 0.13	1.2 ± 0.16	1.5 ± 0.20	1.8 ± 0.24	0.5 ± 0.2
CR (%)	0.2 ± 0.04	0.5 ± 0.08	0.7 ± 0.11	0.9 ± 0.13	1.2 ± 0.16	1.4 ± 0.19	0.4 ± 0.1
TTC (%)	0.3 ± 0.05	0.6 ± 0.09	0.9 ± 0.13	1.1 ± 0.16	1.4 ± 0.19	1.6 ± 0.22	0.4 ± 0.2

Table 3. Comparison of path planning algorithms in a real-world environment.

Planning Algorithm	Length of Path (m)	Maximum Curvature	Training Time (min)	Failure Rate (%)
ESAD-LEND	44.2	0.48	89	3
Dreamer	46.7	0.60	160	7
LatentSW-PPO	45.5	0.43	155	4
SAD-LEN	47.9	0.66	170	9
AD-END	46.3	0.58	165	11
SAC	48.1	0.72	160	14
Diffuser	45.0	0.50	140	5

The comparative analysis of path planning algorithms in both simulated and real-world environments highlights the performance and reliability of the ESAD-LEND method. In the CARLA simulated environment, ESAD-LEND demonstrates superior performance across multiple metrics, achieving the highest DS of 95.3% with minimal variance, indicating consistent performance across trials. Additionally, it excels in RC at 99.2%, significantly ahead of other methods such as Dreamer and LatentSW-PPO, which score lower in both categories.

Furthermore, ESAD-LEND maintains the lowest IS and other critical safety metrics, including CO and Collision per Kilometer (CP), indicating fewer rule violations and safer driving behavior compared to the other models. These results underscore the robustness of ESAD-LEND in adhering to safety standards while effectively navigating complex environments.

6.4. Bridging the Gap between Simulation and Real-World

To bridge the gap between simulation and real-world experiments, our research considered a multi-faceted approach that addresses the sim-to-real transfer and quantifies the reality gap. Inspired by the work [44], we utilize the Pearson correlation coefficient (PCC) and the max normalized cross-correlation (MNCC) to evaluate the correlation between simulation results and real-world data, ensuring that our models perform consistently across both domains.

In real-world settings, the ESAD-LEND continues to demonstrate its efficacy by achieving the shortest path length and lowest maximum curvature among all tested algorithms, resulting in more efficient and smoother paths. Despite slightly longer training times compared to other methods, ESAD-LEND records the lowest failure rate, confirming its reliability and practical applicability in real-world scenarios where unpredictable variables can affect outcomes.

This comprehensive evaluation demonstrates that ESAD-LEND excels in controlled simulations and adapts effectively to real-world conditions, surpassing established algorithms in terms of both efficiency and safety.

6.5. Evaluate Robustness

As shown in Figure 10, we further evaluate the robustness of our proposed framework against different types of obstacles. By adjusting the speed of moving obstacles, we simulate various levels of dynamic complexity and observe how well the model anticipates and reacts to changing trajectories and potential hazards. This test demonstrates the model's obstacle avoidance skills and evaluates its potential real-world applicability and robustness under varying speeds and densities of pedestrian traffic. The speeds of the dynamic objects are set at 1 m/s, 2 m/s, and 3 m/s, with fixed start and goal points.

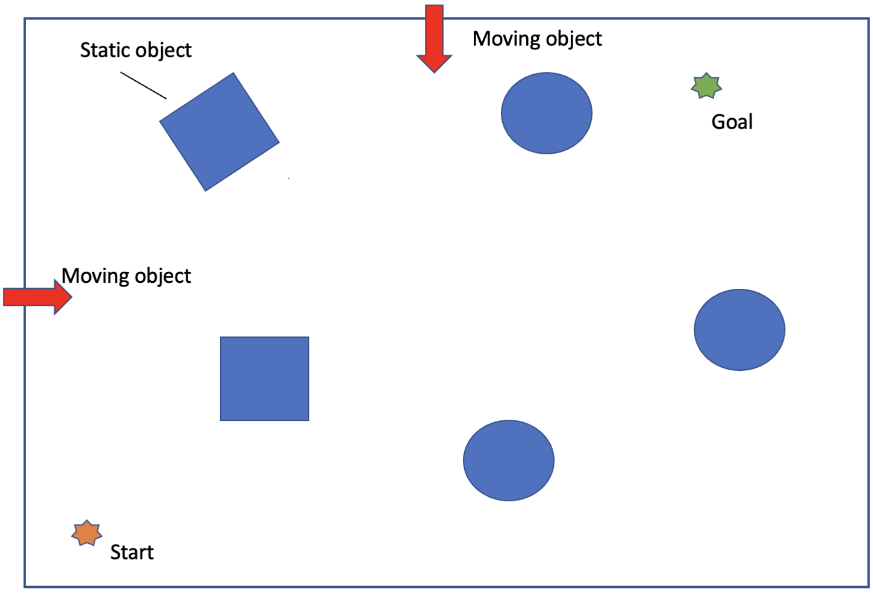


Figure 10. Scenario with static and moving objects to mimic real-world dynamics. Students simulate moving obstacles at speeds of 1 m/s, 2 m/s, and 3 m/s.

The Table 4 provides a detailed comparison of various path planning algorithms under dynamic conditions with varying speeds of moving obstacles. ESAD-LEND consistently outperforms other algorithms across all speed scenarios, demonstrating its superior ability to adapt and maintain high safety scores even with increasing obstacle speed. Notably, ESAD-LEND maintains a low failure rate, with only a slight increase as the obstacle speed escalates from 1 m/s to 3 m/s. This robust performance underscores its effective handling of dynamic challenges in real-world environments.

Table 4. Performance comparison of path planning algorithms under varying speed scenarios.

Speed	1 m/s			2 m/s			3 m/s		
Algorithm	**Fail Rate (%)**	**Avg. Time (s)**	**Safety Score**	**Fail Rate (%)**	**Avg. Time (s)**	**Safety Score**	**Fail Rate (%)**	**Avg. Time (s)**	**Safety Score**
ESAD-LEND	1	120	95	3	130	93	5	140	90
Dreamer	5	140	90	7	150	88	10	170	85
LatentSW-PPO	3	130	93	5	140	90	8	160	87
SAD-LEN	4	135	92	6	145	89	9	165	86
AD-END	7	150	88	10	160	85	14	180	82
SAC	6	145	89	9	155	87	13	175	84
Diffuser	2	125	94	4	135	91	6	150	88

Dreamer demonstrates reasonable performance at lower speeds but exhibits a significant drop in safety scores and an increase in failure rates as the speed increases, indicating limited adaptability to faster-moving obstacles. Similarly, LatentSW-PPO and SAD-LEN perform adequately at lower speeds but struggle to maintain efficiency and safety at higher speeds.

AD-END and SAC, designed for robustness, show the highest failure rates and longest completion times, particularly at higher speeds, suggesting areas for improvement in their algorithms to better handle dynamic and unpredictable environments.

Overall, the analysis highlights that while most algorithms manage slower-moving obstacles adequately, the primary challenge lies in adapting to higher speeds, where reaction times and predictive capabilities are crucial. ESAD-LEND's superior performance underscores its effectiveness in minimizing risks and optimizing path planning across varied dynamic conditions.

7. Conclusions

In this study, we propose a novel end-to-end algorithm for autonomous driving employing safe reinforcement learning. We develop a latent imagination model to forecast future trajectories, enabling the agent to explore within an imagined horizon initially. This method mitigates irreversible damage during training. Additionally, we introduce a VaR-based soft actor-critic to address the constrained optimization problem. Our model-based reinforcement learning approach demonstrates robustness and achieves a balanced trade-off between exploration and exploitation. In experiments, we validate the Carla simulator and real-world environments.

In future research, we plan to utilize other simulators such as Highway [45] to expand testing scenarios. Additionally, we aim to explore multi-agent settings where agents collaborate. We will also consider more complex network structures such as the graph attention algorithm [46,47].

Author Contributions: Conceptualization, H.-F.L. and L.-Y.B.; methodology, D.-T.C.; validation, D.-T.C. and J.-N.H.; investigation, D.-T.C. and W.K.; writing—original draft preparation, H.-F.L. and D.-T.C.; writing—review and editing, D.-T.C. and L.-Y.B.; visualization, D.-T.C.; project administration, H.-F.L.; funding acquisition, H.-F.L.; data curation, Z.-L.F.; formal analysis, J.-N.H. and P.Z.; resources, L.-Y.B.; software, L.-Y.B.; supervision, L.-Y.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 62372148.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: We would like to express our sincere gratitude to the editor and three anonymous reviewers for their valuable comments, which have greatly improved this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
2. Shi, T.; Chen, D.; Chen, K.; Li, Z. Offline Reinforcement Learning for Autonomous Driving with Safety and Exploration Enhancement. *arXiv* **2021**, arXiv:2110.07067.
3. Kuutti, S.J. End-to-End Deep Learning Control for Autonomous Vehicles. Ph.D. Thesis, University of Surrey, Guildford, UK, 2022.
4. Xiao, Y.; Codevilla, F.; Gurram, A.; Urfalioglu, O.; López, A.M. Multimodal end-to-end autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 537–547. [CrossRef]
5. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A.A.; Yogamani, S.; Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4909–4926. [CrossRef]
6. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
7. Kim, B.; Lee, K.H.; Xue, L.; Niu, X. A review of dynamic network models with latent variables. *Stat. Surv.* **2018**, *12*, 105. [CrossRef] [PubMed]
8. Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained policy optimization. In Proceedings of the International Conference on Machine Learning (PMLR), Sydney, NSW, Australia, 6–11 August 2017; pp. 22–31.
9. Alan, A.; Taylor, A.J.; He, C.R.; Ames, A.D.; Orosz, G. Control barrier functions and input-to-state safety with application to automated vehicles. *IEEE Trans. Control. Syst. Technol.* **2023**, *31*, 2744–2759. [CrossRef]
10. Verstraete, D.; Droguett, E.; Modarres, M. A deep adversarial approach based on multi-sensor fusion for semi-supervised remaining useful life prognostics. *Sensors* **2019**, *20*, 176. [CrossRef]
11. Hao, Z.; Zhu, H.; Chen, W.; Cai, R. Latent Causal Dynamics Model for Model-Based Reinforcement Learning. In Proceedings of the International Conference on Neural Information Processing, Changsha, China, 20–23 November 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 219–230.
12. Wei, J.; Dolan, J.M.; Snider, J.M.; Litkouhi, B. A point-based MDP for robust single-lane autonomous driving behavior under uncertainties. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2586–2592.
13. Shalev-Shwartz, S.; Shammah, S.; Shashua, A. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv* **2016**, arXiv:1610.03295.
14. Garcıa, J.; Fernández, F. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* **2015**, *16*, 1437–1480.
15. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning (PMLR), Lille, France, 7–9 July 2015; pp. 1889–1897.
16. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
17. Deisenroth, M.P.; Fox, D.; Rasmussen, C.E. Gaussian processes for data-efficient learning in robotics and control. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *37*, 408–423. [CrossRef]
18. Koller, T.; Berkenkamp, F.; Turchetta, M.; Krause, A. Learning-based model predictive control for safe exploration. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 6059–6066.
19. Zanon, M.; Gros, S. Safe reinforcement learning using robust MPC. *IEEE Trans. Autom. Control* **2020**, *66*, 3638–3652. [CrossRef]
20. 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.
21. Hewing, L.; Wabersich, K.P.; Menner, M.; Zeilinger, M.N. Learning-based model predictive control: Toward safe learning in control. *Annu. Rev. Control Robot. Auton. Syst.* **2020**, *3*, 269–296. [CrossRef]
22. Brüdigam, T.; Olbrich, M.; Wollherr, D.; Leibold, M. Stochastic model predictive control with a safety guarantee for automated driving. *IEEE Trans. Intell. Veh.* **2021**, *8*, 22–36. [CrossRef]
23. Pek, C.; Zahn, P.; Althoff, M. Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1477–1483.
24. Wang, J.; Zhang, Q.; Zhao, D.; Chen, Y. Lane change decision-making through deep reinforcement learning with rule-based constraints. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–6.
25. Sewell, D.K.; Chen, Y. Latent space models for dynamic networks. *J. Am. Stat. Assoc.* **2015**, *110*, 1646–1657. [CrossRef]

26. Sarkar, P.; Moore, A.W. Dynamic social network analysis using latent space models. *ACM SIGKDD Explor. Newsl.* **2005**, *7*, 31–40. [CrossRef]
27. Padakandla, S. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–25. [CrossRef]
28. Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv* **2018**, arXiv:1805.00909.
29. Lee, K.; Seo, Y.; Lee, S.; Lee, H.; Shin, J. Context-aware dynamics model for generalization in model-based reinforcement learning. In Proceedings of the International Conference on Machine Learning (PMLR), Virtual, 13–18 July 2020; pp. 5757–5766.
30. Chen, J.; Li, S.E.; Tomizuka, M. Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 5068–5078. [CrossRef]
31. Li, Y.; Song, J.; Ermon, S. Inferring the latent structure of human decision-making from raw visual inputs. *arXiv* **2017**, arXiv:1703.08840.
32. Wang, H.; Gao, H.; Yuan, S.; Zhao, H.; Wang, K.; Wang, X.; Li, K.; Li, D. Interpretable decision-making for autonomous vehicles at highway on-ramps with latent space reinforcement learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8707–8719. [CrossRef]
33. Gulcehre, C.; Colmenarejo, S.G.; Sygnowski, J.; Paine, T.; Zolna, K.; Chen, Y.; Hoffman, M.; Pascanu, R.; de Freitas, N. Addressing Extrapolation Error in Deep Offline Reinforcement Learning. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 4 May 2021.
34. Venkatraman, S.; Khaitan, S.; Akella, R.T.; Dolan, J.; Schneider, J.; Berseth, G. Reasoning with latent diffusion in offline reinforcement learning. *arXiv* **2023**, arXiv:2309.06599.
35. Janner, M.; Du, Y.; Tenenbaum, J.B.; Levine, S. Planning with Diffusion for Flexible Behavior Synthesis. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022.
36. Zhang, Z.; Li, A.; Lim, A.; Chen, M. Predicting Long-Term Human Behaviors in Discrete Representations via Physics-Guided Diffusion. *arXiv* **2024**, arXiv:2405.19528.
37. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
38. Kumar, A.; Zhou, A.; Tucker, G.; Levine, S. Conservative q-learning for offline reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1179–1191.
39. Hafner, D.; Lillicrap, T.; Ba, J.; Norouzi, M. Dream to control: Learning behaviors by latent imagination. *arXiv* **2019**, arXiv:1912.01603.
40. Yang, Q.; Simão, T.D.; Tindemans, S.H.; Spaan, M.T. WCSAC: Worst-case soft actor critic for safety-constrained reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 10639–10646.
41. Mavrin, B.; Yao, H.; Kong, L.; Wu, K.; Yu, Y. Distributional reinforcement learning for efficient exploration. In Proceedings of the International Conference on Machine Learning (PMLR), Long Beach, CA, USA, 9–15 June 2019; pp. 4424–4434.
42. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning (PMLR), Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
43. Wang, C.; Wang, Y. Safe Autonomous Driving with Latent Dynamics and State-Wise Constraints. *Sensors* **2024**, *24*, 3139. [CrossRef]
44. Daza, I.G.; Izquierdo, R.; Martínez, L.M.; Benderius, O.; Llorca, D.F. Sim-to-real transfer and reality gap modeling in model predictive control for autonomous driving. *Appl. Intell.* **2023**, *53*, 12719–12735. [CrossRef]
45. Leurent, E. An Environment for Autonomous Driving Decision-Making. 2018. Available online: https://github.com/eleurent/highway-env (accessed on 20 April 2024).
46. Shi, T.; Wang, J.; Wu, Y.; Miranda-Moreno, L.; Sun, L. Efficient connected and automated driving system with multi-agent graph reinforcement learning. *arXiv* **2020**, arXiv:2007.02794.
47. Wang, J.; Shi, T.; Wu, Y.; Miranda-Moreno, L.; Sun, L. Multi-agent Graph Reinforcement Learning for Connected Automated Driving. In Proceedings of the International Conference on Machine Learning (ICML) Workshop on AI for Autonomous Driving, Virtual Event, 13–18 July 2020.