

# Cross-Scenario Maneuver Decision with Adaptive Perception for Autonomous Driving

Yuan Fu  
University of Electronic Science and  
Technology of China  
Chengdu, China  
fuyuan@std.uestc.edu.cn

Shuncheng Liu  
University of Electronic Science and  
Technology of China  
Chengdu, China  
liushuncheng@std.uestc.edu.cn

Yuyang Xia  
University of Electronic Science and  
Technology of China  
Chengdu, China  
xiayuyang@std.uestc.edu.cn

Fangda Guo  
Institute of Computing Technology,  
Chinese Academy of Sciences  
Beijing, China  
guofangda@ict.ac.cn

Kai Zheng<sup>\*†</sup>  
University of Electronic Science and  
Technology of China  
Chengdu, China  
zhengkai@uestc.edu.cn

## ABSTRACT

Autonomous driving is a rapidly advancing field that promises to revolutionize the transportation industry through an intelligent perception-and-decision paradigm. Despite decades of research, existing methods are limited in adapting to complex scenarios or expanding to unseen situations, which pose significant challenges to the development of autonomous driving. Inspired by the process of human learning to drive, autonomous vehicles can prioritize developing driving capabilities in basic scenarios and then extending the atomic abilities to more complex scenarios. To this end, we proposed a perception-and-decision framework, called *ATEND*, which consists of an adaptive perception module and a maneuver decision module. Specifically, the perception module based on Variational Autoencoder is proposed to map perceptual data of complex scenarios into basic scenarios. Then the reinforcement learning-based decision module can make high-level decisions in transformed scenarios. Once *ATEND* learns to drive in basic scenarios, it can achieve safe and efficient driving in real scenarios without additional training. Extensive experiments in different traffic scenarios evidence that the proposed framework advances the state of the art in terms of both macroscopic and microscopic effectiveness.

## CCS CONCEPTS

• **Applied computing** → **Transportation**; • **Computing methodologies** → *Perception*.

<sup>\*</sup>Corresponding author: Kai Zheng.

<sup>†</sup>Kai Zheng is with Yangtze Delta Region Institute (Quzhou), School of Computer Science and Engineering, and Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*CIKM '23, October 21–25, 2023, Birmingham, United Kingdom*  
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-0124-5/23/10...\$15.00  
<https://doi.org/10.1145/3583780.3614831>

## KEYWORDS

Autonomous Driving; Environmental Perception; Maneuver Decision

### ACM Reference Format:

Yuan Fu, Shuncheng Liu, Yuyang Xia, Fangda Guo, and Kai Zheng. 2023. Cross-Scenario Maneuver Decision with Adaptive Perception for Autonomous Driving. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3583780.3614831>

## 1 INTRODUCTION

Autonomous driving has been developed with the promise of preventing accidents, reducing emissions, transporting the mobility-impaired, and reducing driving-related stress. With the rapid development of autonomous driving systems [28], the technology stacks converge to a modular perception-decision pipeline. As the eyes and brains of autonomous vehicles, perception and decision modules are designed to comprehend dynamic environments and make tactical lane/velocity changes. Both components can be developed either based on classical non-learning approaches [24, 32] or deep learning methodologies [9, 34], and they attempt to replace the human drivers' observation and arbitration to prevent dangerous driving behaviors (e.g., hard braking and abrupt lane-changing). Although they have achieved safe, efficient, and comfortable autonomous driving in some trial scenarios [39], they still suffer from a long tail of unseen or unfamiliar scenarios, which is one reason why we have yet to see autonomous vehicles deployed at scale.

The *perception stage* enables the autonomous vehicle to sense and interpret its surroundings using a combination of sensing hardware and algorithms [27]. With the development of onboard sensors and computer vision algorithms, we find that the bottleneck of perceiving dynamic scenarios lies in comprehending the perceived results. Some traditional approaches employ spatio-temporal graphs [34, 36] to organize the states of surrounding vehicles and utilize a graph-based neural network to encode the graph data into compact vectors. Despite simplifying the complicated perceived results, they do not adapt to varying lane structures and vehicle counts. Recently, some methods [4, 31] fuse multi-modal



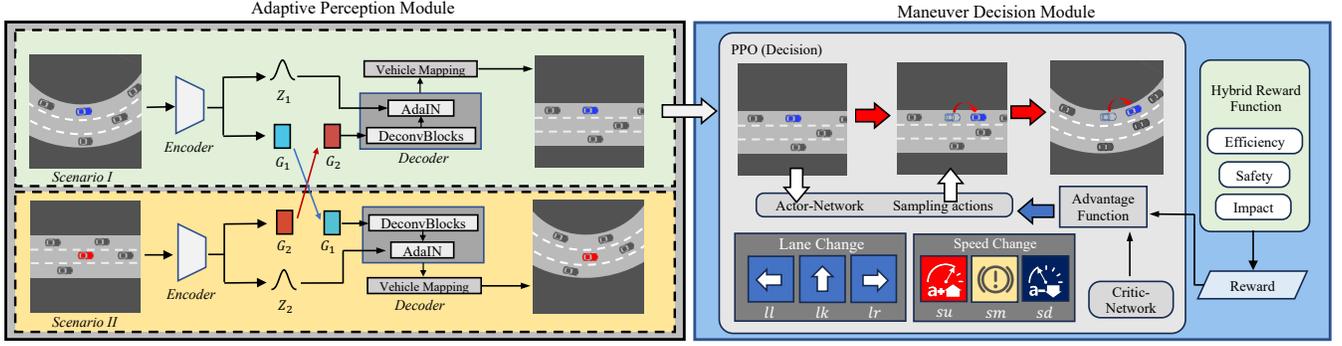


Figure 2: ATEND Framework Overview

**BEV Representation.** Bird’s Eye View (BEV) images capture road geometry and vehicle states centered around the autonomous vehicle. We represent each image as a bitmap with three semantic channels: road, position, and velocity (as shown in Figure 1). In the road and position channels, pixels occupied by lanes and lane markings are set to 255 and 204, respectively, while other pixels are set to 0. In the position channel, pixels occupied by vehicles are set to 255, while other pixels are set to 0. In the velocity channel, pixels occupied by vehicles are set to the velocity of the vehicles, which maps the velocity (from  $v_{min}$  to  $v_{max}$ ) to grayscale values (from 55 to 255). Each channel ( $100 \times 100$ ) covers a detection area of  $50m \times 50m$ , and unoccupied pixels are set to 0 as default.

**Traffic Scenarios** We divide various traffic scenarios into *basic* and *real* scenarios. The basic scenarios have fixed lane counts and simple road topology, and the real scenarios have varying lane counts or complex road topology. As shown in Figure 1, the basic scenario is a straight multi-lane road, while the real scenario is a roundabout section. In this work, the basic scenarios are taken as simple experimental environments, while the real scenarios are viewed as long-tail deployment environments.

**Maneuver.** A maneuver is a pair of *lane change behavior* and *velocity change behavior* performed by a vehicle.  $(A^t.L, A^t.V)$  represents the maneuver of  $A$  at time step  $t$ , where  $L$  is one of three lane change behaviors: *change lane to left* ( $ll$ ), *change lane to right* ( $lr$ ), and *lane keep* ( $lk$ ) (i.e.,  $L \in \{ll, lr, lk\}$ ), and  $V$  is one of three velocity change behaviors: *speed-up* ( $su$ ), *speed-down* ( $sd$ ), and *maintain speed* ( $sm$ ) (i.e.,  $V \in \{su, sd, sm\}$ ). The high-level maneuvers are cross-scenario since controllers [38] can perform them based on environments.

**Objective.** In this work, our objective is that *after the autonomous vehicle learns to drive in basic scenarios, it can achieve safe and efficient driving in real scenarios without additional preparation.*

## 2.2 Framework Overview

Figure 2 presents the architecture of ATEND, which consists of two components: *adaptive perception module* and *maneuver decision module*. After the autonomous vehicle learns to drive in Scenario II (basic straight road), we attempt to adapt it to Scenario I (real curved road). We first use the pre-trained adaptive perception module to encode BEV images from Scenario I and Scenario II into vectors  $Z$  (road distribution) and  $G$  (road geometry). Then we exchange the vectors  $G$  of two scenarios, and the reorganized results are decoded to generate new BEV images, which preserve the road

distribution and vehicle states but swap the road geometry. The transformed BEV images of Scenario I are fed into the maneuver decision module. The proximal policy optimization-based model (PPO) combined with the hybrid reward function can make safe and efficient decisions based on the transformed scenario, and the high-level maneuvers can be acted in Scenario I.

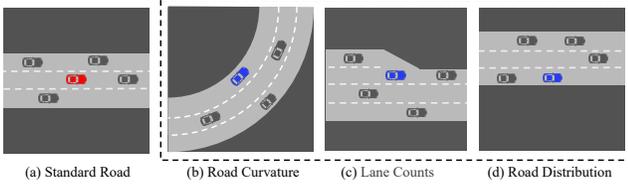
## 3 ADAPTIVE PERCEPTION

To help autonomous vehicles adapt to diverse and complex scenarios, we propose an adaptive perception module, which maps complex environmental attributes into basic space. In this section, we first define the problems, and then we introduce the module.

### 3.1 Problem Definition

Converting real scenarios to basic scenarios can be viewed as a domain transfer task. The perceived real scenario is the source domain  $\mathcal{D}_S$ , and the transformed basic scenario is the target domain  $\mathcal{D}_T$ . Based on the Markov Decision Processes (MDPs), the source domain is defined as  $\mathcal{D}_S = (\mathcal{S}^{(s)}, \mathcal{A}, \mathcal{T}, r, \gamma)$ , and the target domain is denoted as  $\mathcal{D}_T = (\mathcal{S}^{(t)}, \mathcal{A}, \mathcal{T}, r, \gamma)$ . The states  $(\mathcal{S}^{(s)}, \mathcal{S}^{(t)})$  indicate the acquired BEV images, which vary widely in different domains. The action spaces  $\mathcal{A}$ , transitions  $\mathcal{T}$ , and reward functions  $r$  are shared across both domains. The differences between  $\mathcal{S}^{(s)}$  and  $\mathcal{S}^{(t)}$  can be summarized into three categories: 1) the road geometry, 2) the road distribution, and 3) the vehicle information. As shown in Figure 3, the road geometry involves road curvature and lane counts, and the road distribution indicates the road layout in the image determined by the position of the autonomous vehicle. For example, the road distribution in Figure 3(d) is different from that in Figure 3(a) because the autonomous vehicle changes from the middle lane to the right lane. The vehicle information includes the position and velocity of vehicles, which are represented by the position and velocity channels in BEV images. The objective of the adaptive perception module is to *transform the road geometry of the source domain into those of the target domain, while maintaining the road distribution and the vehicle states.*

The objective is motivated by the following analyses: (1) High-level maneuvers of autonomous vehicles are independent of road geometry. We only need to preserve the correct lane distribution and vehicle information in the basic scenario, so that the autonomous vehicle can make appropriate decisions in both scenarios. (2) Maneuver decision module can adapt to dynamic vehicle information


**Figure 3: Examples of Road Variations**

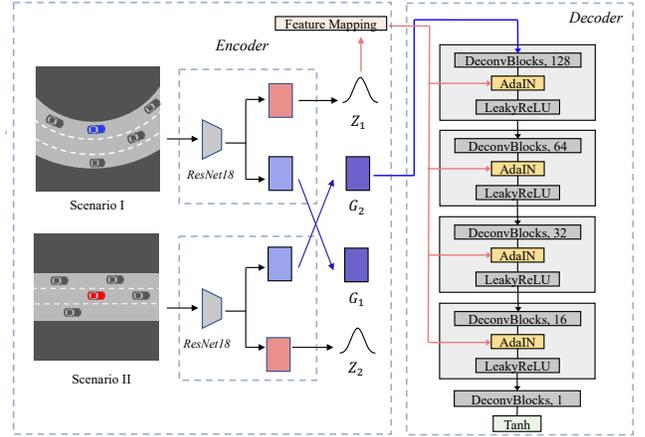
and road distribution in basic scenarios. Since the basic scenarios have simple road geometry, the RL-based maneuver decision module can make optimized decisions based on different vehicle information and road distribution. In order to achieve the objective, we need to first identify the distribution features and geometric features in the BEV images of real scenarios. Then we can replace the complex geometric features with simple features derived from basic scenarios. Finally, we should accurately map the vehicle information to convert the real scenario into a basic one. Next, we elaborate the model that implements the above procedure.

### 3.2 EVA-AIN Model

We propose a model to convert real scenarios to basic scenarios, called EVA-AIN (Encoder-Decoder with Variational Autoencoder and Adaptive Instance normalization). As shown in Figure 4, the VAE-based EVA-AIN model consists of an encoder with ResNet [7] and a decoder with an AdaIN layer [10]. VAE-based models have been proven to be effective in disentangling features for various style transfer tasks [8, 11]. Similar to Cycle-Consistent VAE [11], we employ the swapping strategy in the encoder to disentangle the distribution features and geometric features in BEV images. In addition, we concatenate the deconvolutional block and the AdaIN layer [10] in the decoder, which can reconstruct the BEV images that preset the road geometry of the basic scenario while keeping the road distribution of the real scenario. Finally, we accurately map the vehicle information from real scenarios to the generated BEV images. Next, we introduce the workflow and the components of the model.

**Workflow.** During the training phase, our EVA-AIN model learns to disentangle road distribution and geometry with BEV images, which are collected in some predefined basic scenarios and real scenarios in a simulator [18]. Similar to [11], we use a pair of training BEV images from the same scenario in each iteration. Since their road geometries are very similar, the difference between the swapped latent spaces  $G_1$  and  $G_2$  is small. This training strategy can enhance training stability and efficiency. After pre-training, we can use a pair of BEV images from two scenarios as shown in Figure 4. Then we can swap the latent space  $G_1$  and  $G_2$  that represent the road geometry to transform the road geometry of Scenario I into Scenario II while maintaining the road distribution.

**Encoder.** The encoder can be defined as a mapping function [11] that takes a BEV image as input, denoted as  $\text{Enc}(x) = (f_z(x), f_g(x))$ , where  $f_z(x) = (\mu, \sigma) = Z_x$  and  $f_g(x) = G_x$ . The  $x$  denotes the input of the encoder. The function  $f_g(x)$  is a typical encoder with vector latent space, and  $f_z(x)$  is an encoder that parameterizes the approximate posterior  $q_\phi(z|x)$ . In other tasks,  $f_z(x)$  is often used for style feature mapping. As the autonomous vehicle remains in the center of the image, any positional change is reflected as a road


**Figure 4: EVA-AIN Network Structure**

distribution shift, similar to style features in style transfer tasks. Thus, the road distribution in the BEV image is represented by  $f_z(x)$  in our case. In the meantime, we employ  $f_g(x)$  to represent the road geometry in various scenarios. To enhance representation accuracy and efficiency, we employ a pre-trained ResNet-18 [7] to extract features from the input state for mapping  $Z$  and  $G$ . The entire encoder shares weights except for the final layer, which branches out to provide outputs for the two functions  $f_z(x)$  and  $f_g(x)$ .

**Decoder.** The decoder  $x' = \text{Dec}(z, g)$  is represented by the conditional likelihood  $p_\theta(x|z, g)$  [11]. Here,  $x'$  denotes the reconstructed image generated by the decoder, and the conditional likelihood function  $p_\theta(x|z, g)$  represents the probability of generating the corresponding  $x$  based on the given  $z$  and  $g$ . When we maximize the conditional likelihood, we are essentially optimizing the decoder's parameters  $\theta$  to produce reconstructed samples that are closer to the true samples, i.e., minimizing the squared reconstruction error. In our case, we expect the decoder to obtain the features of road distribution from  $f_z(x)$  and the features of road geometry from  $f_g(x)$ . By using the latent space  $Z$  as input for each AdaIN module [10, 12] and the latent space  $G$  as input for the deconvolution layers, we can generate the BEV images that reflect the road geometry of the basic scenarios while still maintaining the road distribution of the real scenarios.

**Optimization.** The EVA-AIN model needs to minimize the loss function [11] as follows:

$$\mathcal{L}_1 = -\mathbb{E}_{q_\phi(z|x, g)} [\log p_\theta(x|z, g)] + \text{KL}(q_\phi(z|x, g) \| p(z)) \quad (1)$$

This loss function consists of two components: reconstruction error and KL divergence. The reconstruction error measures the difference between the original data and the reconstructed data, which is obtained by encoding input data  $x$  into the latent space  $z$  using the encoder, represented as  $q_\phi(z|x, g)$  and decoding it back through the decoder, represented as  $p_\theta(x|z, g)$ . On the other hand, KL divergence measures the difference between the distribution of the latent space  $z$  obtained by the encoder and the prior latent space distribution  $p(z)$ . Minimizing KL divergence ensures that the learned latent space distribution is as close to the prior distribution as possible, avoiding overfitting and improving generalization capabilities [11, 15].

**Vehicle Mapping.** EVA-AIN adapts effectively to variations in road geometry and distribution, ensuring the consistency of road distribution in the transformed BEV image with the real scenario. Consequently, we use the locations of the road and the autonomous vehicle as a reference for vehicle position mapping [33]. By calculating their relative distances to the lane centerline and the autonomous vehicle, we can accurately transfer vehicle positions.

## 4 MANEUVER DECISION

Leveraging the adaptive perception module, we can first map the states of real scenarios into basic scenarios, and then an RL-based maneuver decision module only need to be trained on basic scenarios to adapt to real scenarios. Therefore, our objective is to employ a deep reinforcement learning-based model with a hybrid reward function to solve the single-scenario MDP, focusing on learning the optimal decision in basic scenarios.

### 4.1 Environment Modeling

Based on the problem defined in Section 3.1, we model the maneuver decision task under the Markov decision process (MDP). An MDP can be defined as:  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma)$ , which includes the state spaces  $\mathcal{S}$ , the action spaces  $\mathcal{A}$ , the transitions  $\mathcal{T}$ , the reward functions  $r$ , and the discount factor  $\gamma$ . These components collectively describe the entire interaction process between an agent and the environment, as we proceed to detail the main tuples.

**State.** The state  $s^t$  represents a BEV perception of the agent  $A$ 's surroundings within a 50-meter driving range at time  $t$ , including the channels of road, position, and velocity, as defined in Section 2. The adaptive perception module simplifies the state captured from the real scenarios to the style of the basic scenarios, allowing the autonomous vehicle to adapt and make proper maneuver decisions.

**Action.** According to the definition in Section 2, The maneuver of the autonomous vehicle involves performing high-level actions to change the desired lane or speed. We define the action  $a^t$  of vehicle  $A$  at time step  $t$  as follows:

$$a^t = (A^t.L, A^t.V) \quad (2)$$

where  $L \in \{ll, lr, lk\}$  and  $V \in \{su, sd, sm\}$  represent the discrete high-level actions for lateral lane change and speed change, respectively. During the process of acceleration or deceleration, the absolute value of acceleration is maintained at  $2 m/s^2$ .

**State Transition.** After the autonomous vehicle executes an action  $a^t$  at the state  $S^t$ , we update the current state  $S^{t+1}$  at time step  $t+1$  by capturing the road, current positions of vehicles, and vehicle speed information within the autonomous vehicle's range, and reflecting these updates in the BEV image.

**Reward.** After executing an action at a state  $S^t$ , the autonomous vehicle will obtain a reward as feedback, denoted as  $r^t$ . We construct a hybrid reward function to determine  $r^t$ , taking into account three aspects: 1) safety, 2) efficiency, and 3) impact, which are detailed in Section 4.2.

**Optimization.** Based on the above MDP, our goal is to find a policy  $\pi$  that guides the autonomous vehicle in selecting the best action for each visual state  $S^t$ . In this work, we utilize Proximal Policy Optimization (PPO) [30] as the policy gradient algorithm in the maneuver decision module. This module includes an action policy

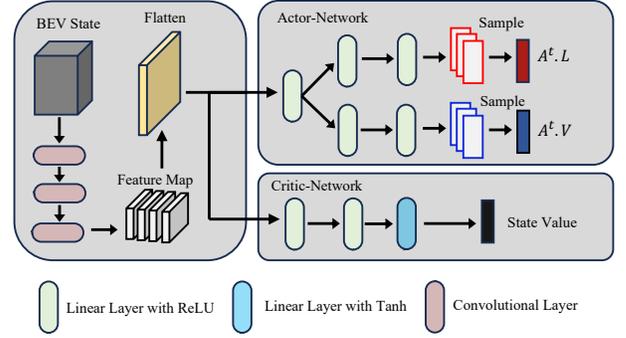


Figure 5: Network Structure of Maneuver Decision

$\pi(a|s)$  and a state-value function  $\mathcal{V}(s)$ . At each time step  $t$ , the policy  $\pi(a|s)$  generates a probability distribution over actions  $a^t$  based on the current state  $S^t$ , guiding the agent to select an action accordingly. Meanwhile, the state-value function  $\mathcal{V}(s)$  estimates the expected return or the cumulative discounted reward from the current state  $S$ . Unlike other reinforcement learning algorithms such as DDPG [20], PPO does not directly compute the state-action value function  $Q^\pi(s, a)$  but rather estimates the advantage function  $D(t)$  using the GAE [29] method. Thus, data sampled from  $T$  timesteps is used for updating the loss function, which incorporates the advantage function as follows:

$$D(t) = \delta^t + \gamma \lambda D(t+1) \quad (3)$$

where  $\gamma$  is a discount factor and  $\delta^t$  is defined as follows:

$$\delta^t = r^t + \gamma \mathcal{V}(s^{t+1}) - \mathcal{V}(s^t) \quad (4)$$

Moreover, traditional policy gradient methods tend to substantially alter the policy at each update. To mitigate this, PPO introduces a clipped objective function as follows:

$$\max \mathcal{L}_{(\pi', \pi)} = \mathbb{E}_t [\min (\rho(t) D(t), \text{clip}(\rho(t), 1 - \epsilon, 1 + \epsilon) D(t))] \quad (5)$$

where  $\rho(t)$  represents the ratio between the current policy  $\pi'$  and the old policy  $\pi$  at time  $t$ . Clip is a function that ensures  $\rho(t)$  is within the range of  $[1 - \epsilon, 1 + \epsilon]$ . Therefore, by optimizing this objective, PPO ensures stable policy updates while still improving the agent's policy towards better actions.

**Network Structure.** As shown in Figure 5, PPO employs an actor-critic structure in which the actor network is responsible for selecting the optimal action given the current environment state. Typically, the critic network represents the state value function, which combined with the feedback reward, is used to calculate the advantage function in order to direct the policy (i.e. action selection) update. In addition, a convolutional feature extractor is utilized to encode the state  $S^t$  at time step  $t$ . This extractor consists of three convolutional layers that convert a 3-channel image into a 64-channel feature map  $M^t$  that, after flattened, functions as the input for our actor-critic network. Consisting of three linear layers, the actor network branches into two outputs, with each output generating three categorical probability distributions. Sampling these probability distributions determines the optimal action for the current state. The two branches output high-level maneuver decisions for lane-changing  $X_{\text{out}}^t.L$  and speed-changing  $X_{\text{out}}^t.V$  respectively,

i.e.,  $L \in \{ll, lr, lk\}$  and  $V \in \{su, sd, sm\}$ . The computations are as follows:

$$X_{\text{out}}^t \cdot L = \text{ReLU}(\phi_3(\text{ReLU}(\phi_2(\text{ReLU}(\phi_1 M^t + b_1)) + b_2)) + b_3) \quad (6)$$

$$X_{\text{out}}^t \cdot V = \text{ReLU}(\phi_4(\text{ReLU}(\phi_2(\text{ReLU}(\phi_1 M^t + b_1)) + b_2)) + b_4) \quad (7)$$

where  $\phi_1, \phi_2, \phi_3, \phi_4$  are the linear transformations, and  $b_1, b_2, b_3, b_4$  are their biases.

The Critic-Network receives the same inputs as the Actor-Network for the calculation of state value functions, i.e. predicting an achievable reward under the given state. The computations are as follows:

$$X_{\text{value}}^t = \text{Tanh}(\phi_7(\text{ReLU}(\phi_6(\text{ReLU}(\phi_5 M^t + b_5)) + b_6)) + b_7) \quad (8)$$

where  $\phi_5, \phi_6, \phi_7$  are the linear transformations, and  $b_5, b_6, b_7$  are their biases.

## 4.2 Hybrid Reward Function

The hybrid reward function guides the autonomous vehicle towards learning the optimal policy by prioritizing safety, efficiency, and minimal negative impacts on surrounding conventional vehicles. Therefore, we construct a hybrid reward function considering three aspects: 1) safety, 2) efficiency, and 3) impact, as follows:

$$r^t = w_1 R_{\text{safety}}^t + w_2 R_{\text{efficiency}}^t + w_3 R_{\text{impact}}^t \quad (9)$$

where  $w_1, w_2$ , and  $w_3$  are three adjustable coefficients to balance the importance of safety, efficiency, and impact, respectively.

**Safety.** Time to collision (TTC), a widely used safety indicator, represents the time remaining before a collision occurs if two vehicles maintain their current velocities [6]. In accordance with the standard definition [39], the TTC of the autonomous vehicle  $A$  with its front conventional vehicle  $C$ , after the autonomous vehicle performs an action, is defined as follows:

$$TTC^t = \frac{d(C^t, A^t)}{-v(C^t, A^t)} \quad (10)$$

where  $C^t$  represents the front conventional vehicle of the autonomous vehicle at time  $t$ ,  $d$  and  $v$  refer to the relative distance and speed between the two vehicles in the lane direction, respectively. Besides, when  $v(C^t, A^t) < 0$ , i.e.,  $C^t \cdot v \geq A^t \cdot v$ , the calculated  $TTC^t$  is considered invalid. Additionally, if the autonomous vehicle causes a collision, we assign a negative reward to indicate its compromised safety. Therefore, the safety reward value  $R_{\text{safety}}^t$  is defined in the range of  $[-10, 0]$  as follows:

$$R_{\text{safety}}^t = \begin{cases} -10 & \text{collision} \\ \max\left(-10, \log\left(\frac{TTC^t}{4}\right)\right) & 0 < TTC^t < 4 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

In this reward function, a collision refers to either a vehicular crash or road boundary infringement. Often, the autonomous vehicle instigates a collision when  $TTC^t$  is less than 4 [32, 39], prompting us to exponentially indicate its deteriorating safety.

**Efficiency.** The longitudinal velocity of the autonomous vehicle is a direct measure of its driving efficiency. Therefore, we define the efficiency reward as follows:

$$R_{\text{efficiency}}^t = \frac{A^t \cdot v - v_{\min}}{v_{\max} - v_{\min}} \quad (12)$$

where  $A^t \cdot v$  denotes the speed of the agent along the road direction, with the remaining symbols retaining their meanings as defined in Section 2.

**Impact.** The actions of the autonomous vehicle, including deceleration or lane changing, may impact the efficiency of nearby conventional vehicles [22, 36]. To account for this, we incorporate an impact reward that reflects how much the autonomous vehicle influences the deceleration of other vehicles. This is assessed by tracking the deceleration of the conventional vehicle behind the autonomous vehicle following its action. This impact reward value,  $R_{\text{impact}}^t \in [-1, 0]$ , is defined as follows:

$$R_{\text{impact}}^t = \begin{cases} \frac{C^{t+1} \cdot v - C^t \cdot v}{2a' * \Delta t} & C^{t+1} \cdot v - C^t \cdot v > v_{\text{impact}} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where  $v_{\text{impact}}$  is a velocity threshold employed to ascertain the impact of the autonomous vehicle on the trailing conventional vehicle  $C$ . Furthermore,  $2a' * \Delta t$  denotes the maximum velocity change occurring between two consecutive time steps.

## 5 EXPERIMENTS

### 5.1 Experimental Settings

**Simulator Setting.** As it requires interaction between autonomous and conventional vehicles, most of the experiments are conducted in a simulated environment called Highway-env [18]. Highway-env is a 2D open-source and gym-like [3] autonomous driving simulation environment. To gain more control and information during the simulation process, we utilize Highway-env and Gym-related APIs along with Python to conveniently establish interaction between the simulator and the model. In Highway-env, the lane width is 5 meters for each scenario. There are many highway-env-controlled conventional vehicles and one *ATEND*-controlled autonomous vehicle traveling on the road. The road density is established at 150 vehicles per kilometer, which is a suitable setting for autonomous driving research [25, 34]. The autonomous vehicle is initially placed at the starting point of the road on a randomly selected lane. We set the traffic restrictions as  $V_{\min} = 0$  km/h,  $V_{\max} = 72$  km/h = 20 m/s, and  $a' = 5$  m/s<sup>2</sup>, following the settings in the previous works [22, 39]. We define an episode as the autonomous vehicle driving from the origin to the destination of the road or driving from the origin to the location of a collision. Each episode is initialized randomly to ensure diversity. Meanwhile, we refer to the total number of interactions between the autonomous vehicle and the environment in the simulator as total timesteps. In each timestep, the agent selects and executes an action based on the observed information and the reinforcement learning policy. We train for 4,000 episodes in each experiment and test within 500 episodes.

**Implementation Details.** We implement the adaptive perception and maneuver decision modules in *ATEND* as follows:

(1) Adaptive perception module. As described in Section 2, this module utilizes a sensor with a  $50m \times 50m$  detection area and processes three semantic channels: road, position, and velocity. Regarding the network structure of the EVA-AIN model introduced in section 3.2, we set the encoder's output latent space dimensions to 64. To ensure decision-making precision and generated image

**Table 1: Macroscopic Effectiveness: End-to-End Performance of Baselines and *ATEND***

Methods	Macroscopic Metrics											
	Highway			Roundabout			Bottleneck			Tollgate		
	Rate NC-C (%)	Rate NC-L (%)	AvgT -A (s)									
IDM	90.20	100.00	221.42	85.40	96.20	26.76	83.80	84.20	79.13	84.60	86.40	76.67
ACC-LC	92.40	<b>100.00</b>	215.12	89.60	<b>97.80</b>	24.22	82.40	86.60	76.80	83.80	88.20	72.15
<i>ATEND</i> -STG-w/o-EVA-AIN	<b>97.80</b>	96.20	<b>190.90</b>	-	-	-	86.60	87.20	64.48	87.20	88.60	62.09
<i>ATEND</i> -BEV-w/o-EVA-AIN	96.80	97.40	194.29	-	-	-	87.20	91.40	64.25	90.80	94.40	58.78
<b><i>ATEND</i></b>	96.60	98.40	192.56	<b>95.40</b>	96.80	<b>19.49</b>	<b>92.80</b>	<b>95.20</b>	<b>61.06</b>	<b>95.60</b>	<b>96.80</b>	<b>55.25</b>

quality, we apply a thresholding operation to the output of EVA-AIN. pixel values within the range of 0.1 and 1 are mapped to 1, and all other values are mapped to 0. In addition, we train the EVA-AIN model by using Adam optimizer [14] for 100 epochs with a learning rate of 0.0003 and a batch size of 64 by default.

(2) Maneuver decision module. For the network structure of PPO, we set the corresponding dimensions as  $D_{\phi_1} = 128$ ,  $D_{\phi_2} = 64$ ,  $D_{\phi_3} = 3$ ,  $D_{\phi_4} = 3$ ,  $D_{\phi_5} = 128$ ,  $D_{\phi_6} = 64$ ,  $D_{\phi_7} = 1$ . In addition, the tuneable coefficients for the hybrid reward function are set as follows:  $w_1 = 0.90$ ,  $w_2 = 0.80$ , and  $w_3 = 0.20$ , following the settings in the previous works [22, 26, 34]. We train the maneuver decision module using Adam optimizer [14] for 4,000 episodes with a scheduled learning rate of 0.001. Our experimental results are reported based on the above settings unless expressly specified.

**Traffic Scenarios.** We design four driving scenarios as follows:

(1) Highway: The autonomous vehicle negotiates a three-lane highway that is populated by other vehicles. The distance from the start to the end in this scenario is 3 kilometers.

(2) Roundabout: The autonomous vehicle navigates a 50-meter radius roundabout with flowing traffic. Assuming that all vehicles travel in an anticlockwise direction, the start and end points are chosen within the roundabout, with a total distance of 300 meters.

(3) Bottleneck: In this scenario, the lane count decreases from five to three within the bottleneck area. The autonomous vehicle travels a total distance of 800 meters. We use blue markers to denote unreachable road obstacles in sections where the road changes.

(4) Tollgate: The lane count expands from three to seven forming a tollgate area. The total distance is 800 meters.

**Compared Methods.** Since existing learning-based methods cannot accomplish our multi-scenario driving tasks, we employ the rule-based approaches and the variants of *ATEND* as the baselines:

(1) IDM [32]. The traditional Intelligent Driver Model (IDM) is a rule-based method that incorporates an anticipated velocity and following distance when in a free state.

(2) ACC-LC [18, 24, 35]. The Adaptive Cruise Control (ACC) with a lane-changing policy from Highway-env serves as a traditional rule-based decision-making system.

(3) *ATEND*-STG-w/o-EVA-AIN. We remove the adaptive perception module and adjust the model's input to a traditional Spatial-Temporal Graph (STG) [36], capturing the relative distances and speeds of six surrounding vehicles.

(4) *ATEND*-BEV-w/o-EVA-AIN. We remove the adaptive perception module but maintain our BEV input format.

## 5.2 End-to-End Evaluation

We evaluate the end-to-end performance of *ATEND* by comparing it against several baselines (IDM, ACC-LC, *ATEND*-STG-w/o-EVA-AIN, and *ATEND*-BEV-w/o-EVA-AIN). *ATEND* and its variant baselines are trained in highway scenarios, followed by conducting 500 test episodes across all constructed scenarios. EVA-AIN is trained using data sampled from the experimental scenarios. We measure the effectiveness from both macroscopic and microscopic perspectives.

**Macroscopic Effectiveness.** We design three macroscopic metrics to test the driving performance of the autonomous vehicle for each test episode as follows:

(1) No-Collision with Conventional Vehicles Episode Rate (RateNC-C): This metric quantifies the percentage of episodes in which the autonomous vehicle successfully avoids collisions specifically with conventional vehicles. A higher RateNC-C value implies safer interactions with conventional traffic.

(2) No-Collision with Lane Boundaries Episode Rate (RateNC-L): This is the proportion of episodes in which the autonomous vehicle successfully avoids collisions with the boundaries of the lane in the current scenario. A higher RateNC-L signifies improved scenario adaptation by the autonomous vehicle.

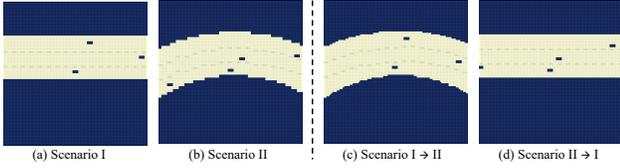
(3) Average Driving Time of the Autonomous Vehicle (AvgT-A): We record the end-to-end driving time for the autonomous vehicle in each scenario from the starting point to the end in no-collision test episodes. A smaller AvgT-A indicates greater driving efficiency for the autonomous vehicle.

We report RateNC-C, RateNC-L, and AvgT-A in Table 1. As shown, rule-based methods such as IDM and ACC-LC struggle to execute efficient maneuvers, resulting in an increase in AvgT-A. *ATEND*-STG-w/o-EVA-AIN and *ATEND*-BEV-w/o-EVA-AIN tend to fail in the Roundabout scenario where road curvature changes and their driving safety metrics RateNC-L and RateNC-C are lower in Bottleneck and Tollgate scenarios where the lane count varies. In addition, *ATEND* achieves a higher RateNC-L and RateNC-C and a lower AvgT-A across the majority of training scenarios. These results demonstrate that only using STG and BEV for perception is not sufficient to adapt well to the scenario changes. However, *ATEND* can not only adapt to varying scenarios but also makes safe and efficient maneuvering decisions.

**Microscopic Effectiveness.** We record three microscopic metrics based on our reward function in Section 4.2, to evaluate the driving performance of the autonomous vehicle at each time step as follows:

**Table 2: Microscopic Effectiveness: End-to-End Performance of Baselines and *ATEND***

Methods	Microscopic Metrics											
	Highway			Roundabout			Bottleneck			Tollgate		
	Min TTC-A (s)	Avg Vel-A (m/s)	Avg Dec-C (m/s)	Min TTC-A (s)	Avg Vel-A (m/s)	Avg Dec-C (m/s)	Min TTC-A (s)	Avg Vel-A (m/s)	Avg Dec-C (m/s)	Min TTC-A (s)	Avg Vel-A (m/s)	Avg Dec-C (m/s)
IDM	3.52	13.55	0.28	3.36	11.21	0.29	3.31	10.11	0.30	3.46	10.43	0.27
ACC-LC	3.61	13.95	0.26	3.56	12.39	0.27	3.48	10.42	0.28	3.54	11.09	0.26
<i>ATEND</i> -STG-w/o-EVA-AIN	3.74	<b>15.72</b>	0.21	-	-	-	3.61	12.41	0.23	3.61	12.88	0.20
<i>ATEND</i> -BEV-w/o-EVA-AIN	3.65	15.44	<b>0.19</b>	-	-	-	3.54	12.45	0.25	<b>3.68</b>	13.61	0.22
<b><i>ATEND</i></b>	<b>3.76</b>	15.58	0.20	<b>3.85</b>	<b>15.39</b>	<b>0.18</b>	<b>3.65</b>	<b>13.10</b>	<b>0.21</b>	3.67	<b>14.48</b>	<b>0.18</b>

**Figure 6: Curvature Change in Scenario Translation Tests**

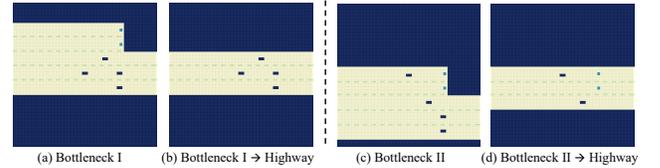
- (1) Minimum Time to Collision of the Autonomous Vehicle (MinTTC-A): This metric corresponds to the minimum value of TTC for the autonomous vehicle. A larger AvgTTC-A value signifies a safer performance by the autonomous vehicle.
- (2) Average Velocity of the Autonomous Vehicle (AvgVel-A): This is the mean velocity of the autonomous vehicle in no-collision test episodes. A larger AvgVel-A suggests a faster autonomous vehicle, thereby indicating better efficiency.
- (3) Average Deceleration of the Conventional Vehicle Behind the Autonomous Vehicle (AvgDec-C): This records the average deceleration of the conventional vehicle trailing the autonomous vehicle across all test episodes. A smaller AvgDec-C signifies that the autonomous vehicle imposes less disruption on the conventional vehicles following it, indicating more harmonious interaction with conventional traffic.

We report MinTTC-A, AvgVel-A, and AvgDec-C in Table 2. In the majority of scenarios, *ATEND* has the longer MinTTC-A, the higher AvgVel-A, and the lower AvgDec-C, proving that *ATEND* enables the autonomous vehicle to execute safe cross-scenario maneuvers with high velocity and minimal impact on surrounding vehicles.

### 5.3 Evaluation of Adaptive Perception

We evaluate EVA-AIN’s translation ability, i.e. how well it maps BEV images from real scenarios into a basic and interpretable space. During scenario transformation, we evaluate two subtasks of the perception module: curvature change and lane count change. Additionally, we employ t-SNE plots to evaluate the disentangling capability of the encoder within EVA-AIN.

**Curvature Change.** We train EVA-AIN in Highway and Roundabout scenarios, utilizing observations collected by an agent operating under a random policy in these two scenarios. As depicted in Figure 6, the pre-trained Adaptive Perception Module enables the autonomous vehicle to support flexible mapping between the perceived information from Scenario I (Highway) and Scenario II (Roundabout). For instance, when the RL-based decision module of the autonomous vehicle is trained in Scenario I (Highway), it can

**Figure 7: Lane Count Change in Scenario Translation Tests**

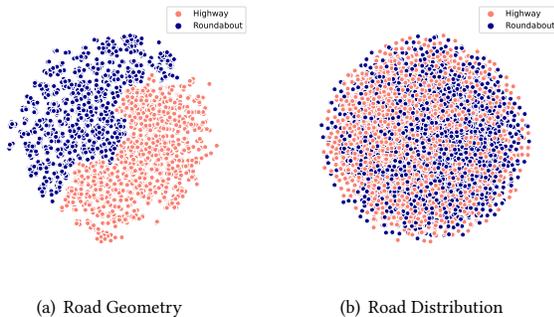
use the Adaptive Perception Module to translate states encountered in Scenario II into interpretable Scenario I’s style, and then process these transformed states to efficiently execute high-level maneuvers. From the generated results, it’s evident that the distribution of lane positions and the relative locations of surrounding vehicles in the image are maintained during the scenario translation, ensuring that decisions based on the transformed scenario remain valid and applicable to the real scenario.

**Lane Count Change.** We trained EVA-AIN in Highway scenarios with three and five lanes respectively, using the same sampling method. This enables the autonomous vehicle to adapt to variations in the number of lanes, as depicted by the Bottleneck scenario in Figure 7. The blue markers, which represent unreachable road obstacles, are incorporated into the transformation results during the vehicle mapping phase. They serve as references in the autonomous decision-making process. In the Bottleneck I scenario depicted in Figure 7(a), the pre-trained Adaptive Perception Module enables an autonomous vehicle trained in a Highway scenario to adapt to lane variations by translating real scenario states into Highway-style states when not in lane reduction areas. In scenarios like Bottleneck II in Figure 7(c), the autonomous vehicle (situated in the image’s center) retains its road-edge position in the transformed state. The accurate translation of lane position distributions ensures that decisions made by the autonomous vehicle, based on the transformed state, prevent potentially dangerous actions such as collisions with road boundaries or obstacles.

**Evaluation of Disentangling Ability.** Just like [11], we also use t-SNE plots to evaluate the disentangling capability of the EVA-AIN encoder. We randomly sampled 800 BEV images from Highway and Roundabout scenarios, disentangling them into two latent spaces representing road geometry and road distribution using a pre-trained EVA-AIN, as described in Section 3.2. As shown in Figure 8, we used t-SNE plots for visualization. The points are color-coded to represent specific scenario types. Since the primary difference between the two scenarios is road geometry, we can observe a good cluster formation according to the latent space of road

**Table 3: Generalization Performance of *ATEND* and *ATEND-BEV-w/o-EVA-AIN* Variants Across Various Scenarios**

Methods	Training Scenarios	Macroscopic Metrics											
		Highway			Roundabout			Bottleneck			Tollgate		
		Rate NC-C (%)	Rate NC-L (%)	AvgT -A (s)	Rate NC-C (%)	Rate NC-L (%)	AvgT -A (s)	Rate NC-C (%)	Rate NC-L (%)	AvgT -A (s)	Rate NC-C (%)	Rate NC-L (%)	AvgT -A (s)
<i>ATEND-BEV-w/o-EVA-AIN-1</i>	Highway	<b>96.80</b>	97.40	194.29	-	-	-	87.20	91.40	64.25	90.80	94.40	58.78
<i>ATEND-BEV-w/o-EVA-AIN-2</i>	Roundabout	-	-	-	94.80	<b>97.80</b>	21.35	-	-	-	-	-	-
<i>ATEND-BEV-w/o-EVA-AIN-3</i>	Bottleneck	90.20	94.80	195.83	-	-	-	92.60	<b>96.40</b>	<b>59.49</b>	91.40	94.60	59.85
<i>ATEND-BEV-w/o-EVA-AIN-4</i>	Tollgate	91.80	95.60	198.37	-	-	-	90.40	94.80	63.45	94.80	<b>97.20</b>	56.34
<i>ATEND</i>	Highway	96.60	<b>98.40</b>	<b>192.56</b>	<b>95.40</b>	96.80	<b>19.49</b>	<b>92.80</b>	95.20	61.06	<b>95.60</b>	96.80	<b>55.25</b>

**Figure 8: Scenario Disentanglement Tests**

geometry in Figure 8(a). Figure 8(b) shows that the latent space of road distribution is mixed between the two scenarios, proving it doesn't contain Road Geometry information that could separate scenario types. These results demonstrate that EVA-AIN has a strong disentangling ability for the two features of road geometry and road distribution.

#### 5.4 Generalization Effect of Maneuver Decision

We evaluate *ATEND*'s generalization capability by comparing it to four *ATEND-BEV-w/o-EVA-AIN* variants, each with its decision module trained in one of the four distinct scenarios: highway, roundabout, bottleneck, and tollgate. In this evaluation, all comparative methods utilize the BEV state format. We measure the effectiveness from a macroscopic perspective. We report RateNC-C, RateNC-L, and AvgT-A in Table 3. The *ATEND-BEV-w/o-EVA-AIN-1* variant trained in the Highway scenario tends to fail in the Roundabout scenario due to varying road curvature. Similarly, the *ATEND-BEV-w/o-EVA-AIN-2* variant trained in the Roundabout scenario is prone to collisions in scenarios featuring straight roads such as the Highway, Bottleneck, and Tollgate scenarios. In the Bottleneck and Tollgate scenarios, *ATEND-BEV-w/o-EVA-AIN-3* and *ATEND-BEV-w/o-EVA-AIN-4*, both have lower RateNC-C, RateNC-L, and AvgT-A in three scenarios outside of their training scenarios, compared to *ATEND*. This demonstrates that they are largely limited to their original training scenarios. Moreover, *ATEND*'s performance in all test scenarios closely matches that of variants trained specifically for those scenarios. As shown, these results demonstrate *ATEND*'s adaptability across various scenarios. As well, the decisions made by *ATEND* based on the transformed basic-style states remain valid in real scenarios.

## 6 RELATED WORK

Perception-and-decision frameworks aim to equip autonomous vehicles with the ability to understand dynamic environments and make lane and/or velocity changes. Next, we will introduce some representative methods and analyze their shortcomings.

**Perception.** Traditional methods use spatiotemporal graphs to simplify complex perceptual results [36], but they struggle with varying lane structures and vehicle counts. Vision-based methods [1, 2, 21] use forward-facing camera images to sense the environment, but they face challenges with changing weather, lights, and road variations. Recently, some works [4, 31] encode bird's-eye view images into latent space to capture key environmental attributes. But these methods may not cover all traffic conditions, and autonomous vehicles struggle to comprehend complex and dynamic images.

**Decision.** Some traditional decision-making methods, e.g., IDM [32], ACC [24, 35], and Krauss [17], employ rule-matching algorithms to maintain vehicle spacing. However, as autonomous driving scenarios become more complex, rule-based approaches often struggle to adapt to environmental changes. Considering the mechanism when a driver perceives the surrounding traffic and decides on a maneuver decision suits well within the realm of reinforcement learning [16]. Numerous studies utilize reinforcement learning to make maneuver decisions, e.g., AD-DDPG [39], MCTS-DRL [9], and EA-DQN [19]. However, these methods are trained in limited scenarios, which makes them difficult to adapt to unfamiliar environments.

## 7 CONCLUSION

In this work, we propose a novel perception-and-decision framework (*ATEND*) that enhances the ability of autonomous vehicles to make suitable maneuvers in various scenarios. *ATEND* uses an adaptive perception module to comprehend complex scenarios and maps perceptions into basic scenarios. A reinforcement learning-based decision module then makes high-level decisions by these transformed scenarios. After *ATEND* learns in basic scenarios, it can be applied in real scenarios without additional training. Our experiments demonstrate *ATEND*'s superiority in terms of both macroscopic and microscopic effectiveness.

## ACKNOWLEDGMENTS

This work is partially supported by NSFC (No. 61972069, 61836007, 61832017, 62272086), Shenzhen Municipal Science and Technology R&D Funding Basic Research Program (JCYJ20210324133607021), Municipal Government of Quzhou under Grant No. 2022D037, and Key Laboratory of Data Intelligence and Cognitive Computing, Longhua District, Shenzhen.

## REFERENCES

- [1] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).
- [2] Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. 2017. Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv preprint arXiv:1704.07911* (2017).
- [3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
- [4] Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. 2019. Model-free deep reinforcement learning for urban autonomous driving. In *2019 IEEE intelligent transportation systems conference (ITSC)*. IEEE, 2765–2771.
- [5] Jakob Erdmann. 2015. SUMO's lane-changing model. In *Modeling Mobility with Open Data: 2nd SUMO Conference 2014 Berlin, Germany, May 15-16, 2014*. Springer, 105–123.
- [6] Leonard Evans. 1991. *Traffic safety and the driver*. Science Serving Society.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [8] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*.
- [9] Carl-Johan Hoel, Katherine Driggs-Campbell, Krister Wolff, Leo Laine, and Mykel J Kochenderfer. 2019. Combining planning and deep reinforcement learning in tactical decision making for autonomous driving. *IEEE transactions on intelligent vehicles* 5, 2 (2019), 294–305.
- [10] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*. 1501–1510.
- [11] Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VS Rao Veeravasaru. 2018. Disentangling factors of variation with cycle-consistent variational autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 805–820.
- [12] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4401–4410.
- [13] Nasser Kehtarnavaz, N Groszold, Kell Miller, and P Lascoe. 1998. A transportable neural-network approach to autonomous vehicle following. *IEEE Transactions on Vehicular Technology* 47, 2 (1998), 694–702.
- [14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [15] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [16] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* 23, 6 (2021), 4909–4926.
- [17] Stefan Krauß, Peter Wagner, and Christian Gawron. 1997. Metastable states in a microscopic model of traffic flow. *Physical Review E* 55, 5 (1997), 5597.
- [18] Edouard Leurent. 2018. An Environment for Autonomous Driving Decision-Making. <https://github.com/eleurent/highway-env>.
- [19] Edouard Leurent and Jean Mercat. 2019. Social attention for autonomous decision-making in dense traffic. *arXiv preprint arXiv:1911.12250* (2019).
- [20] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).
- [21] Fei Liu, Zihao Lu, and Xianke Lin. 2022. Vision-Based Environmental Perception for Autonomous Driving. *arXiv preprint arXiv:2212.11453* (2022).
- [22] Shuncheng Liu, Han Su, Yan Zhao, Kai Zeng, and Kai Zheng. 2021. Lane change scheduling for autonomous vehicle: A prediction-and-search framework. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3343–3353.
- [23] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. 2022. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. *arXiv preprint arXiv:2205.13542* (2022).
- [24] Vicente Milanés and Steven E Shladover. 2014. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. *Transportation Research Part C: Emerging Technologies* 48 (2014), 285–300.
- [25] Kai Nagel, Dietrich E Wolf, Peter Wagner, and Patrice Simon. 1998. Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E* 58, 2 (1998), 1425.
- [26] Xiaobo Qu, Yang Yu, Mofan Zhou, Chin-Teng Lin, and Xiangyu Wang. 2020. Jointly dampening traffic oscillations and improving energy consumption with electric, connected and automated vehicles: a reinforcement learning based approach. *Applied Energy* 257 (2020), 114030.
- [27] Francisca Rosique, Pedro J Navarro, Carlos Fernández, and Antonio Padilla. 2019. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors* 19, 3 (2019), 648.
- [28] David Schrank, Tim Lomax, Bill Eisele, et al. 2011. 2011 urban mobility report. (2011).
- [29] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* (2015).
- [30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- [31] Siyu Teng, Long Chen, Yunfeng Ai, Yuanye Zhou, Zhe Xuanyuan, and Xuemin Hu. 2022. Hierarchical Interpretable Imitation Learning for End-to-End Autonomous Driving. *IEEE Transactions on Intelligent Vehicles* (2022).
- [32] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical review E* 62, 2 (2000), 1805.
- [33] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. 2010. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE International Conference on Robotics and Automation*. IEEE, 987–993.
- [34] Yuyang Xia, Shuncheng Liu, Xu Chen, Zhi Xu, Kai Zheng, and Han Su. 2022. RISE: A Velocity Control Framework with Minimal Impacts based on Reinforcement Learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2210–2219.
- [35] Lin Xiao, Meng Wang, and Bart Van Arem. 2017. Realistic car-following models for microscopic simulation of adaptive and cooperative adaptive cruise control vehicles. *Transportation Research Record* 2623, 1 (2017), 1–9.
- [36] Zhi Xu, Shuncheng Liu, Ziniu Wu, Xu Chen, Kai Zeng, Kai Zheng, and Han Su. 2021. PATROL: A Velocity Control Framework for Autonomous Vehicle via Spatial-Temporal Reinforcement Learning. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2271–2280.
- [37] Fei Ye, Pin Wang, Ching-Yao Chan, and Jiucui Zhang. 2021. Meta reinforcement learning-based lane change strategy for autonomous vehicles. In *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 223–230.
- [38] Pan Zhao, Jiajia Chen, Yan Song, Xiang Tao, Tiejuan Xu, and Tao Mei. 2012. Design of a control system for an autonomous vehicle based on adaptive-pid. *International Journal of Advanced Robotic Systems* 9, 2 (2012), 44.
- [39] Meixin Zhu, Yin Hai Wang, Ziyuan Pu, Jingyun Hu, Xuesong Wang, and Ruimin Ke. 2020. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *Transportation Research Part C: Emerging Technologies* 117 (2020), 102662.