

## Article

# Driver Intent-Based Intersection Autonomous Driving Collision Avoidance Reinforcement Learning Algorithm

Ting Chen <sup>1</sup>, Youjing Chen <sup>1</sup>, Hao Li <sup>2</sup>, Tao Gao <sup>1,\*</sup>, Huizhao Tu <sup>2</sup> and Siyu Li <sup>1</sup><sup>1</sup> School of Information Engineering, Chang'an University, Xi'an 710064, China<sup>2</sup> Key Laboratory of Road and Traffic Engineering of the Ministry of Education, College of Transportation Engineering, Tongji University, Shanghai 201804, China

\* Correspondence: gaotao@chd.edu.cn

**Abstract:** With the rapid development of artificial intelligent technology, the deep learning method is widely applied to predict human driving intentions due to its relative accuracy of prediction, which is one of critical links for security guarantee in the distributed, mixed driving scenario. In order to sense the intention of human-driven vehicles and reduce the self-driving collision avoidance rate, an improved intention prediction method for human-driving vehicles based on unsupervised, deep inverse reinforcement learning is proposed. Firstly, a contrast discriminator module was proposed to extract richer features. Then, the residual module was created to overcome the drawbacks of gradient disappearance and network degradation with the increase in network layers. Furthermore, the dropout layer was generated to prevent the over-fitting phenomenon in the whole training process of the GRU network, so as to improve the generalization ability of the network model. Finally, abundant experiments were conducted on datasets to evaluate our proposed method. The pass rate of self-driving vehicles with conservative driver probabilities of  $p = 0.25$ ,  $p = 0.4$ , and  $p = 0.6$  improved by a maximum of 8%, 10%, and 3%, compared with the classical method LSTM and VAE + RNN. It indicates that the prediction results of our proposed method fit more with the basic structure of the given traffic scenario in a long-term prediction range, which verifies the effectiveness of our proposed method.

**Keywords:** self-driving vehicles; latent states; variational autoencoder; deep reinforcement learning



**Citation:** Chen, T.; Chen, Y.; Li, H.; Gao, T.; Tu, H.; Li, S. Driver Intent-Based Intersection Autonomous Driving Collision Avoidance Reinforcement Learning Algorithm. *Sensors* **2022**, *22*, 9943. <https://doi.org/10.3390/s22249943>

Academic Editor: Felipe Jiménez

Received: 12 November 2022

Accepted: 14 December 2022

Published: 16 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the development of artificial intelligence and electric vehicles, in the process of converting from a manual driving transportation system to an autonomous driving transportation system, there will inevitably be a stage of mixed driving with different levels of intelligent carriers. The essence of mixed traffic flow is the coexistence of various driving behaviors, without standards to speak of, but the lack of uniformity will lead to difficulties in decision-making. Since human behavior is the most difficult to predict, autonomous vehicles usually adopt conservative driving strategies in mixed traffic flow scenarios, resulting in them constantly being overtaken by manual driving vehicles in the congested traffic flow, and the efficiency of passage is difficult to guarantee [1,2]. How to perceive the intentions of manual driving vehicles more intelligently and further improve the transportation efficiency of autonomous driving vehicles under the condition of ensuring safety has become a challenging task.

Taking the uncontrolled T-intersection scenario as an example, in the absence of critical guidance facilities such as traffic lights, self-driving vehicles must interact with other vehicles in the main lane if they want to merge safely and efficiently from the T-intersection to the main lane [3]. Since each traffic participant has its own driving strategy and driving style, autonomous vehicles need to perceive the hidden information of the surrounding vehicles and plan their own reasonable behavior trajectory accordingly [4,5].

If the latent traits of a human driver are known, that is, whether the driver is aggressive or conservative, the self-driving vehicles adopt different driving strategies correspondingly, so as to obtain a better balance of safety and efficiency. Therefore, it is crucial to accurately judge the driving traits of human-driven vehicles.

The latent traits of human-driven cars are divided into intent estimation and trait estimation [6]. Intent estimation typically uses methods such as probabilistic graphical models and non-parametric belief trackers to predict the future actions of other drivers, thereby providing information for the next trajectory planning for self-driving cars. Song W et al. [7] proposed the use of the continuous hidden Markov model to predict both the high-level motion intentions (e.g., turn right, turn left) and the low-level interaction intentions (e.g., the yield status of related vehicles). Dong C et al. [8] utilized the approach based on the probabilistic graphical model (PGM) to efficiently estimate the intent of self-driving cars and interact with them in ramp merging scenarios, even without communication between vehicles. Bai H et al. [9] proposed an online planning method to estimate latent pedestrian intentions using a partially observable Markov decision process (POMDP) for self-driving vehicles to make the systematical and robust decisions in the presence of many pedestrians.

Trait estimation infers the driving characteristics of drivers, such as driving style, driving preference, fatigue status, and degree of distraction, etc. Supervised and unsupervised learning are usually utilized to classify driving characteristics. Morton et al. [10] learned the latent traits of driver characteristics and input the traits and current environmental states into the policy network to produce multi-modal behaviors. However, the input of the strategy network represents the short-term state of the current vehicle, which does not adequately represent the long-term nature of these strategies. Ma et al. [11] utilized supervised learning to classify the traits of human-driven vehicles for autonomous vehicle navigation at intersections, but trait labels are expensive to obtain and do not typically exist in most real driving datasets. Second, the driving policies are trained with ground truth trait labels rather than predictive features. When feature classifiers and policy networks are combined, generating errors in testing and cascading leads to severe performance degradation. Another class of characteristic driving feature learning methods is variational auto-encoding (VAE) and its variants [12,13]. Moreover, conditional VAE (CVAE) is widely used in trajectory predictions of the pedestrian and vehicle trajectory prediction because discrete potential states represent different behavioral patterns, such as braking and turning. Salzmann T et al. [14] proposed a generative multi-intelligent trails prediction method that generated a probability distribution for the agent's motion planning and decision-making. Ivanovic B et al. [15] proposed CVAE to predict human behavior, which generates multi-modal probability distributions on future human trails based on past human-robot interactions and the future actions of candidate robots. Feng X et al. [16] proposed a model that estimates potential driver characteristics and generates a CVAE for multi-modal trail prediction. These behavior patterns change frequently, and the driving characteristics of each driver are persistent. Bowman et al. [17] introduced a recurrent neural network-based VAE model to simulate the latent properties of sentences and explicitly modeled the overall properties of sentences. Liu et al. [18] was inspired by learning the traits of drivers from trails, encoding the trails of drivers as driving features of drivers using a proposed RNN-based VAE. However, the VAE network has a limited ability to characterize the approximate posterior distribution, resulting in low quality of the generative latent variables. These drawbacks largely limit the ability of the latent variables in VAE to express serial information, which is unique to the vehicle trails.

In recent years, contrast learning has been widely applied to learn feature information from continuous data such as video and pedestrian trajectories [19]. Wang X et al. [20] proposed a Siamese triplet network with rank loss function to train the visual representation method of the convolutional neural network (CNN). Liu Y et al. [21] introduced a social contrastive loss that regularizes the extracted motion representation by discerning the ground truth positive events from synthetic negative ones. Zhe Xie et al. [22] introduced

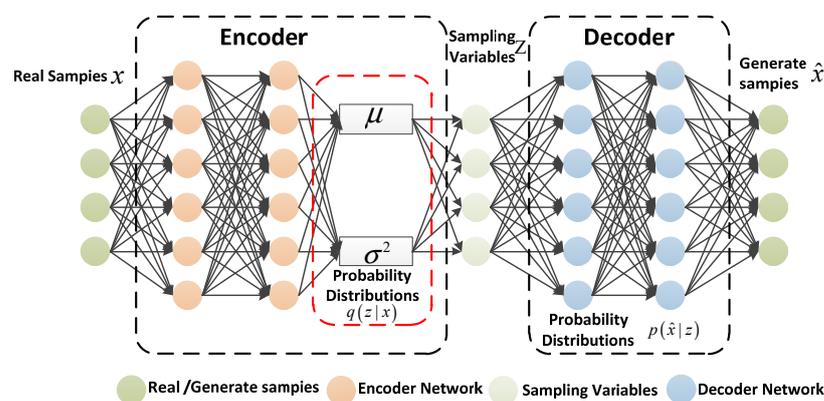
contrast learning into the VAE model and utilized contrast loss to improve the ability of the VAE model to represent features and learn the unique features of different users.

Inspired by contrast learning, this paper improves the VAE + RNN model and introduces contrast loss and the residual network to form the contrastive-ResNet-VAE model (C-ResNet-VAE) so that autonomous vehicles can better avoid people when driving through the uncontrolled intersections. The optimized contrast loss not only enhances the model's ability to separate different features but also improves the model's ability to learn the potential features of different drivers from the trajectory. Our main contributions are as follows:

- (1) In order to improve the ability to learn the potential features of different drivers from the trajectory, contrast learning was proposed into the model, which used the minimization of contrast loss to learn the exclusive features of different drivers in the driver trajectory and enhanced the ability of the model to separate different features.
- (2) We introduced residual modules in the GRU model to capture detailed feature information with strong representational power. These stacked residual units greatly improved the training efficiency, ensuring that the network in the latter layer captured more feature information than the previous layer, reducing information loss. Moreover, the dropout layer was introduced to prevent the over-fitting phenomenon in the whole training process of the gated recurrent unit (GRU) network to improve the generalization ability of the network model.

## 2. Preliminaries

Kingma et al. [12] proposed VAE as a deep generative model in 2013. The VAE model contains two parts: the encoder and decoder. The encoder makes variational inferences with the input and generates an approximate posterior probability distribution of hidden variables. The role of the decoder is to recover the hidden variables to an approximate probability distribution of the input. The overall framework of the VAE model is shown in Figure 1.



**Figure 1.** The overall framework of the VAE model.

$x$  is the real sample,  $z$  is the hidden variable, and  $\hat{x}$  denotes the output of the decoder. Moreover,  $\varphi$  represents the parameters in the encoder, and  $\theta$  represents the parameters in the decoder. Furthermore,  $\mu$  and  $\sigma^2$  represent the mean and variance of the approximate posterior probability distribution of  $z$ . Specifically,  $x$  is an observable random vector of the high-dimensional space, and  $z$  is an unobservable random vector of relatively low-dimensional space. The high-dimensional represents the observable  $x$ -space in the VAE model. The low-dimensional representation decoder reduces the dimensionality of the hidden variable  $z$ . The VAE model sample generation is divided into two processes: the encoder infers the approximate distribution process  $q_{\varphi}(z|x)$  of the hidden variables, and then the decoder restores the hidden variables  $z$  to a process  $P_{\theta}(x|z)P_{\theta}(z)$  similar to the

probability distribution of the input sample.  $q_\varphi(z|x)$  denotes the approximate posterior probability distribution of  $z$ .

Due to the fact that the encoder is unable to obtain the prior probability distribution of the hidden variable  $z$ , the VAE model introduces a learning model  $q_\varphi(z|x)$  instead of the true posterior distribution of the hidden variable  $z$ ; it assumes  $q_\varphi(z|x)$  obeys the ordinary normal distribution. Meanwhile, for calculation convenience, assume that the implicit variable prior distribution  $P_\theta(z)$  follows the standard normal distribution. The optimization goal of the encoder is to get  $q_\varphi(z|x)$  as close as possible to  $p_\theta(x|z)$ .  $p_\theta(x|z)$  denotes the approximate posterior probability distribution of  $x$ .

The VAE model adopts the Kullback–Leibler (KL) divergence [23] to evaluate the similarities between them. Thus, the encoder optimization objective is expressed as:

$$\operatorname{argmin} D_{KL}(q_\varphi(z|x) \parallel p_\theta(x|z)) = \log(P_\theta(X)) - L(\theta, \varphi; X) \quad (1)$$

where  $L(\theta, \varphi; X)$  is the variational lower bound function of the VAE model,  $\log(P_\theta(X))$  is the constant of the encoder, and  $P_\theta(z)$  denotes the hidden variable prior probability distribution.

The VAE model adopts the encoder to learn the posterior distributed parameter mean  $\mu$  and variance  $\sigma^2$  of the latent variables from the input sample, and then performs sampling to obtain the latent variables from the distribution. Since the sampling operations are irreducible, the reparameterization technique was proposed in literature [24]. Specifically, the process of sampling from a normal distribution  $z \sim N(\mu, \sigma^2)$  is replaced with  $\varepsilon$  acquisition from a standard normal distribution, and the parameter transformation  $z = \mu + \varepsilon \times \sigma$  is utilized to obtain the latent variables.  $\varepsilon$  denotes standard normal distribution. After reparameterization transformation, the sampling process is accessible, and the model is able to be trained.

### 3. Proposed Methods

We mostly researched two-way two lanes at an uncontrollable T-Intersection, shown in Figure 2. The vehicle in the lower lane turned left, whereas, the vehicle in the higher lane turned right. An autonomous vehicle turned to the higher lane in a safe way to turn right. More specifically, the blue vehicle was conservative; the red vehicle was aggressive; and the yellow vehicle was autonomous. The conservative vehicle gave way to the autonomous vehicle, but the aggressive vehicle ignored the autonomous vehicle and continued forward.

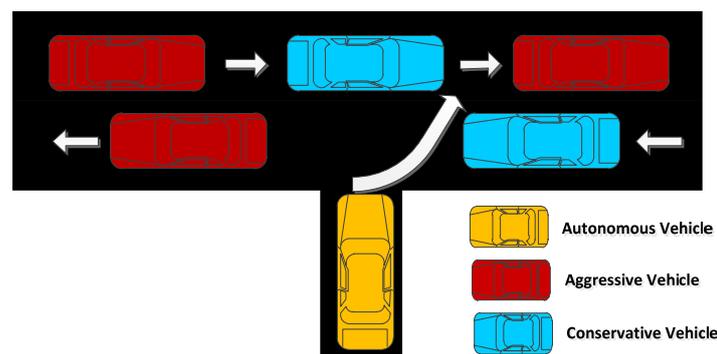
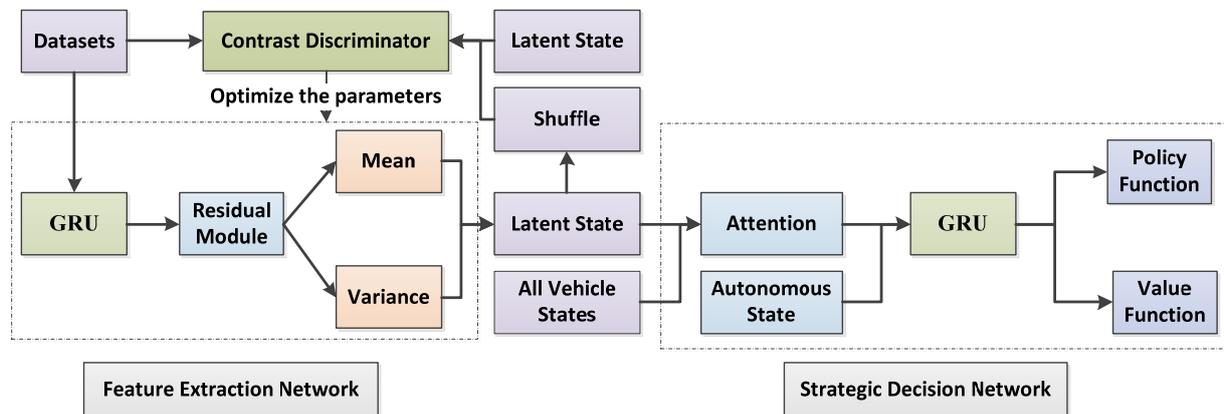


Figure 2. An uncontrollable T-intersection.

Liu et al. adopted VAE + RNN to infer drivers' potential features from the trail of vehicles. However, the VAE network is restricted to deducing the abilities of potential features. We introduced contrastive learning and residual modules based on the VAE + RNN network. Meanwhile, we leveraged C-ResNet-VAE + RNN networks to extract the potential states of various drivers from original driving trails and clusters in an unsupervised way. Figure 3 shows the whole framework of the network. Both datasets and unsorted potential states were simultaneously inputted to the contrastive learning classifier, and contrastive

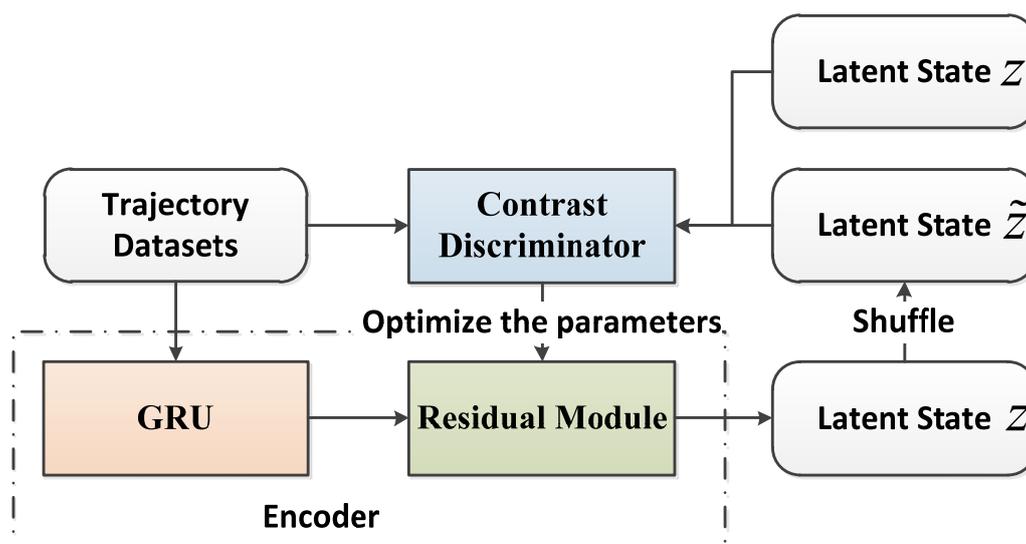
losses were calculated. The contrastive learning classifier optimized encoded parameters in a back propagation way. Then, potential features with the learned features of drivers and all the states of the vehicles to a navigation strategy were submitted. The strategy network contained the GRU network with an attentive module, trained by model-free reinforcement learning. According to inferred features, autonomous vehicles adjusted strategies when interacting with a variety of drivers to perform efficiently.



**Figure 3.** The whole framework of the C-ResNet-VAE + RNN network.

### 3.1. The Network of Potential Features with an Unsupervised Cluster Module

In order to obtain the potential features of every driver from the trails of drivers, having the C-ResNet-VAE + GRU network extract the styles of driving was proposed. The C-ResNet-VAE network is composed of an encoder, contrastive learning classification, and decoder. The encoder squeezes the collected trail  $x$  and forms the distribution of the potential variable  $z$ . The decoder rebuilds the trails from potential features. The potential variable  $z$  is unsorted to get the potential states  $\tilde{z}$ . We inputted both the positive and negative samples,  $(x, z)$  and  $(x, \tilde{z})$ , respectively, to the classifier. Moreover, we calculated contrastive losses and optimized encoded parameters in a back propagation way. The potential features with an unsupervised cluster module are shown in Figure 4.



**Figure 4.** The potential features with an unsupervised cluster module.

### 3.1.1. Contrast Learning

The data of drivers' trails contains generous similar samples, so the VAE module poorly analyzes the drivers' particular features. If it fails to distinguish the different features of every driver, it will influence the results of autonomous driving in a decision network, and the decoder will rebuild the original inputs from the VAE module. If the exclusive target of the module is to reconstruct sequences, something necessary and remarkable, such as drivers' personal information, will be neglected. Here, we aimed to use contrastive learning to train the VAE module, which enhances the representational ability of potential variables.

We compared the contrastive losses, which the potential variable learns in different drivers by using the contrastive classifier  $G_\omega$ , and gained the effective and crucial features of  $x_u$ . We defined a set of coordinates  $(x_u, z_u)$  as positively matching; that is to say, the trail  $x_u$  generated the corresponding potential encoded variable  $z_u$ .

The formulation of contrastive losses is as follows:

$$L_{\omega, \phi}(x_u, z_u) = - \sum_{t=1}^{T_u} [E_{z_u \sim Q_\phi(z_u|x_u)} \log(\sigma(G_\omega(x_u, z_u)^{(t)}))] + E_{\tilde{z}_u \sim Q_\phi(\tilde{z}_u|x_{u'})} \log(\sigma(1 - G_\omega(x_{u'}, \tilde{z}_u)^{(t)})) \quad (2)$$

where  $u$  defines the set of all driver trajectories,  $E$  denotes the evidence lower bound,  $t$  denotes the timestep, and  $T_u$  denotes the number of items in the driver trajectories.

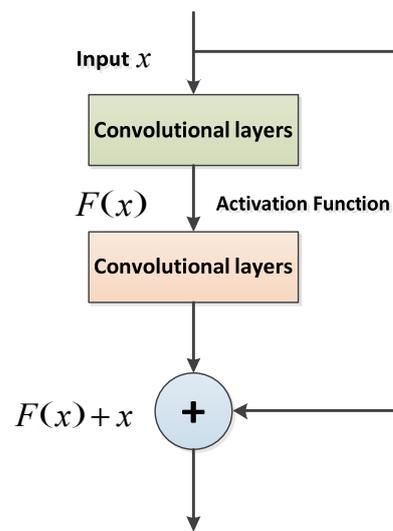
When the contrastive loss is minimum, i.e.,  $L_{\omega, \phi}(x_u, z_u)$ , the classifier  $G_\omega$  distinguishes positive matching with negative matching efficiently. The potential variables that decoders infer will explicitly collect more significant individual information.

### 3.1.2. Feature Extraction Based on Residual

We improved the network based on VAE + GRU and presented a feature extraction network based on residual structure to capture detailed feature information with strong representation ability. Generally speaking, researchers mostly increase the number of network layers to improve the richness of feature information; with more layers or a wider network, the abstract level of feature information will gradually increase. From the initial acquired edge, information gradually becomes more representational semantic information. However, the number of network layers are not deepened infinitely. In the process of deepening the network layers, the model will have a foremost outcome. If the number of network layers continues to increase, the loss will increase accordingly. He et al. [25] presented the ResNet model with the basic idea that residual mapping is easy to optimize, so the ResNet model skips over the convolutional layers and forms the residual unit by using rapid connections. These stacked residual units greatly improve the training efficiency, ensuring that the next layer obtains more feature information than the previous layer and solves the degradation problem caused by the deepening of the network in a large part. The residual unit consists of two main branches. The first branch is identity mapping and the other is residual learning. If the input value of the residual unit is  $x$ , the feature mark obtained by residual learning is  $F(x)$ , and the output value is  $H(x)$ , then this unit is expressed as:

$$H(x) = x + F(x) \quad (3)$$

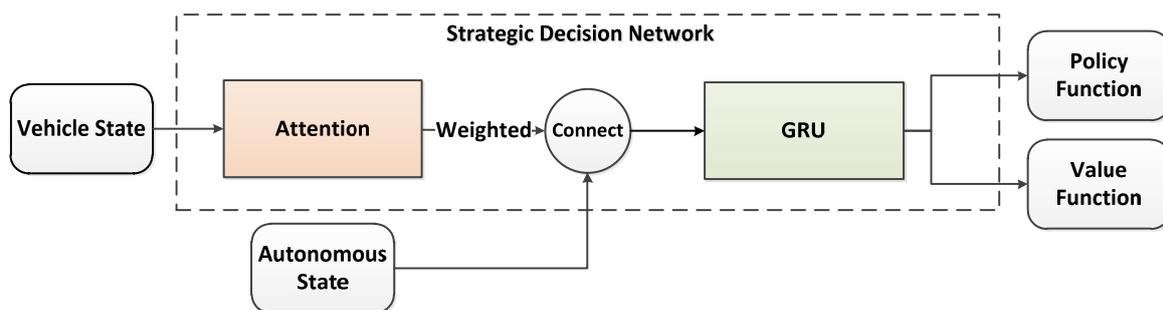
The structure of residual unit is shown in Figure 5.



**Figure 5.** The structure of the residual unit.

### 3.2. Intensive Learning Decision-Making Module Network

As shown in Figure 6, the decision-making module is a GRU network with an attention module. The vehicle status includes the potential characteristics of the driver, all observable vehicle site coordinates, and the site and speed of the autonomous vehicle. Here is how it works. First of all, the vehicle status is inputted to the attention module, which distributes the attention weight to every surrounding vehicle. Then, the weight characteristics counted by each vehicle and the site and speed of the autonomous vehicle are fed into the GRU network. Finally, the hiding states in the GRU network are put into a full connected layer to obtain the value function and the policy function. In this paper, we used a policy gradient method of model-free intensive learning, which we called the proximal policy optimization (PPO) algorithm, to learn the value function and policy function [26] and utilized the approach in this literature [27] to achieve the PPO algorithm.



**Figure 6.** Strategy decision-making module.

#### Proximal Policy Optimization (PPO) Algorithm

This method alternates between the sample data by interacting with the environment and using random gradient rise to optimize the “agent” objective function. Assume  $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ , where  $a_t$  is the current action, and  $s_t$  is the current state.

The objective function of the PPO algorithm is:

$$L(\theta) = \hat{E}t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \hat{A}_t\right] = \hat{E}[r_t(\theta) \hat{A}_t] \quad (4)$$

The  $\hat{A}_t$  is an estimated value of the preponderance function of time step( $t$ ) of  $(s_t, a_t)$ , and  $\hat{E}t[ \dots ]$  represents the expected value for a batch of samples. The policy gradient algorithm is more efficient in continuous action spaces [28].

#### 4. Experiments and Results

Our simulation environment was at the uncontrolled T-intersections, and we assumed that there were  $n$  cars moving toward opposite directions in a two-way street and that all of the vehicles that were controlled by the Intelligent Driver Model (IDM) [29] never took turns or changed lanes. The drivers with a conservative driving style varied their front gaps from the preceding vehicles between 0.5 m and 0.7 m and had the desired speed of 2.4 m/s. The drivers with an aggressive driving style varied their front gaps between 0.3 m and 0.5 m and had the desired speed of 3 m/s. The ego car started at the bottom of the T-intersection and then took a right turn to merge into the upper lane without colliding with other cars. If the ego car encountered the other cars, the conservative drivers would yield to the ego car, while the aggressive drivers would ignore and collide with the ego car. The ego car with a fixed right-turn path was controlled by a longitudinal proportional–derivative (PD) controller, whose desired speed was set by the policy network.

Let the state of the ego vehicle successfully making a full right turn be  $S_{success}$  and the vehicle successfully making a full right-turn have a small reward on the speed, where  $r_{speed}(s) = 0.05 \times \|v_{auto}\|_2$ ;  $v_{auto}$  means the speed of self-driving vehicles. Meanwhile, let the state of the ego vehicle colliding with other vehicles be  $S_{fail}$  and let the vehicle have a constant penalty on the speed, where  $r_{step} = -0.0013$ . Otherwise, we set the length of the cars as 5 cm and the width of the cars as 2 cm. This is to encourage the ego car to reach the goal of making a full right turn as soon as possible. The reward function is defined as:

$$r(s, a) = \begin{cases} 2.5, & s \in S_{success} \\ -2, & s \in S_{fail} \\ r_{speed}(s) + r_{step}, & others \end{cases} \quad (5)$$

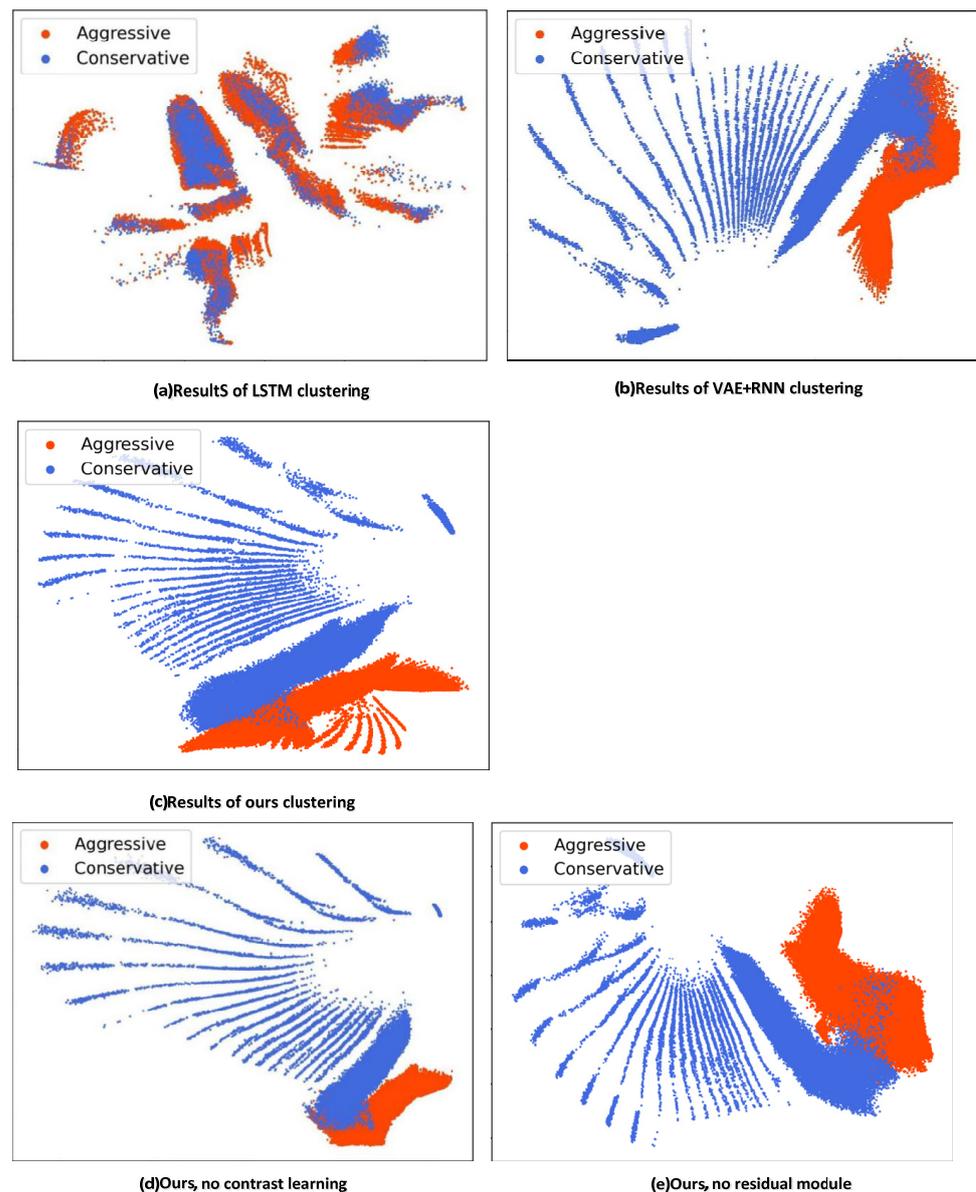
##### 4.1. Datasets

The dataset used in this paper is a randomly generated trajectory dataset in Python produced by Liu et al. Because using Python to generate the dataset not only allows one to set the type of dataset needed, such as the number of entries allocated to radical and conservative trajectories in the program, it also saves the cost of obtaining these datasets. It contains approximately 700,000 driving trajectories from two types of drivers. We set the train/test split ratio is 2:1. We trained the policy model with 466,667 random trajectory data and tested with the other 233,333 trajectory data. We set the decaying learning rate to  $5 \times 10^{-4}$  and the weight of the KL divergence loss to  $\beta 5 \times 10^{-8}$ .

##### 4.2. Unsupervised Clustering Representations of Latent Driver Traits

In this paper we proposed a network of C-ResNet-VAE + GRU and compared it with the VAE + GRU network [18] and GRU network [11]. Both the study [18] and we utilized a GRU as the encoder and GRU the as decoder, while [10] utilized GRU as the encoder and the multilayer perceptron (MLP) as the decoder. In addition, in order to verify the effectiveness of the residual module and reinforcement learning, we trained the ablation experiments without the residual module and comparative loss.

We trained two methods for 500 epochs and then utilized a set of test trajectories as inputs to act as encoders to measure if the latent trait effect was good or bad. The unsupervised classifier results of both methods are shown in Figure 7.



**Figure 7.** Comparison of unsupervised clustering result. The  $x$  and  $y$ -axes are the horizontal and longitudinal displacements in meters.

The red areas represent aggressive drivers, the blue areas represent conservative drivers, and the middle, which is not fully separated out, usually contains very short trajectories and trajectories with vague front clearance. The lines closer to the boundary represent vehicle trajectories where the car has moved out of the range and we cannot see. As shown in Figure 7a–c, the unsupervised classifier we proposed successfully classified most of the result differences in driving styles. From Figure 7b, it can be seen that there are still a large number of blue areas in the red region that have not been successfully separated out, and from Figure 7c, it can be seen that there are significantly fewer blue areas in the red region. It can be seen that the method in this paper can better separate the shorter trajectories and the fuzzy front gap trajectories and is more capable of separating the two different characteristics of conservative and aggressive driver styles. It has a better effect, compared with the methods [10,18] proposed and has the ability to separate the traits of aggressive and conservative driving styles. Study [10] utilized GRU as the encoder to obtain the two latent vectors. We assumed that the vehicles only considered current states

and actions and the encoder only considered the short-term information of the vehicle acceleration, such as the latent traits of drivers.

Therefore, trajectories with different potential characteristics were mostly gathered together and not classified successfully. Study [18] utilized the VAE + RNN network, where VAE had limited approximate posterior distribution trait abilities. Hence, there were some poor sample qualities generated. The C-ResNet-VAE network we proposed improved the approximate posterior distribution trait abilities of the VAE network, obtained richer information from vehicle trajectory, was simpler and better suited for the unsupervised classifier, as well as distinguished the difference between different trajectories, so it had a better performance.

From the results in Figure 7c–e, it can be seen that there are significantly more blue blocks in the red area without the addition of the contrast discriminator module and the residual module, and there are still many driving trajectories with different potential features fused together. With the addition of the contrast discriminator module and the residual module, the separation ability is enhanced, and the clustering effect is better. Thus, the effectiveness of the method is verified.

#### 4.3. Decision Results of Self-Driving Strategies

We used two baselines as comparative experiments:

- (1) The supervised learning with labels proposed by Ma X et al. [11], which trained a supervised trait predictor and a reinforcement learning policy with truth trait labels separately and combined them at test time.
- (2) The strategy of [10,18], and our model all utilize unsupervised methods to infer the potential state of drivers to make reinforcement learning decisions.

In addition, we used a reinforcement learning policy directly trained with truth labels as a baseline. We ran experiments with different proportions of two types of drivers and tested four models with 500 random cases. The percentage of auto-vehicles successfully taking a right turn to merge into the upper lane, colliding with surrounding vehicles and completing overtime, were calculated, respectively, where overtime refers to a situation where the car failed to make a right turn within the allotted time and did not crash. The results are shown in Tables 1–3.

**Table 1.** Objective evaluation decision results of self-driving strategies with aggressive drivers  $p = 0.25$ .

Models	Success (%)	Timeout (%)	Collision (%)
True Labels	85	1	14
GNN	66	18	16
LSTM	70	16	14
VAE + RNN	73	15	12
Ours	78	10	12

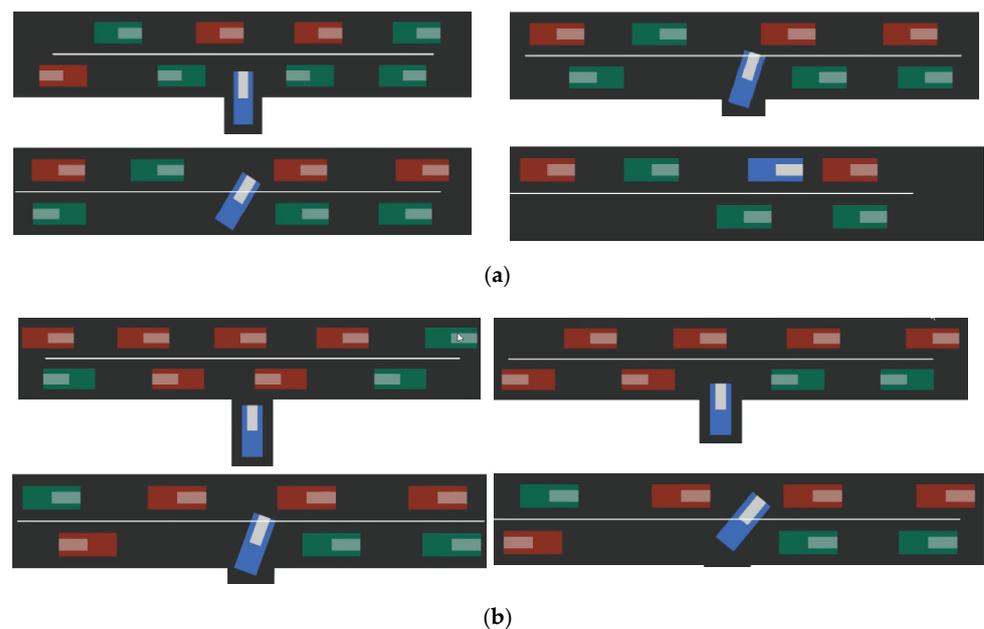
**Table 2.** Objective evaluation decision results of self-driving strategies with aggressive drivers  $p = 0.4$ .

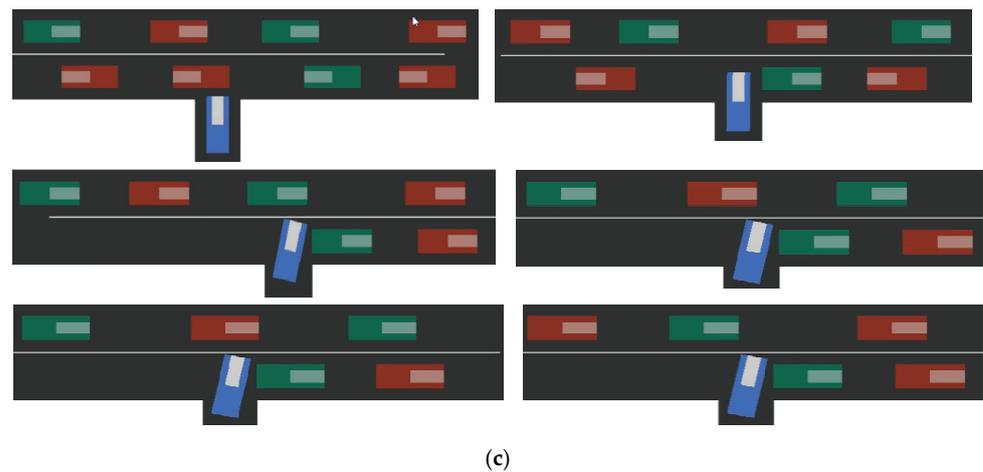
Models	Success (%)	Timeout (%)	Collision (%)
True Labels	91	4	5
GNN	73	22	5
LSTM	74	20	6
VAE + RNN	80	12	8
Ours	84	11	5

**Table 3.** Objective evaluation decision results of self-driving strategies with aggressive drivers  $p = 0.6$ .

Models	Success (%)	Timeout (%)	Collision (%)
True Labels	97	1	2
GNN	87	10	3
LSTM	95	3	2
VAE + RNN	96	2	2
Ours	98	0	2

$p$  is the probability for each surrounding driver to be conservative. The task difficulty decreased as  $p$  increased. From the experimental results, the collision rate of the proposed potential state feature extraction method was lower than the other three methods, and the successful completion of the task accounted for a higher proportion, which is closer to decision-making under the real label training. The main reason was that our method effectively extracted the trait differences of the surrounding drivers, which made better use of the reinforcement learning for the decision-making of the ego vehicle. Our policy was able to utilize the existing trait representation and focused more on the decision-making of the ego vehicle, which led to better navigation performance. The model in [10] had a low success rate when  $p$  value became smaller. The reason is that the latent representation did not distinguish between different traits and only provided very limited useful information to learners. For Ma et al. [11], both strategies had good performance when tested separately. However, when the two modules were combined together, intermediate and cascading errors significantly lowered the success rates. Since the policy was trained with true traits, it failed easily whenever the trait classifier made a small mistake. The model in Liu S et al. [18] had limited representation ability, low separation ability for some fuzzy trajectories (very short trajectories and the trajectories with fuzzy front gaps), and could not learn unique traits of different drivers well. The simulation process of the self-driving car successfully making a full right-turn on the upper lane at the uncontrolled T-intersection is shown in Figure 8. The cars in the two lanes went in opposite directions; the conservative cars are in green, and the aggressive cars are in red.

**Figure 8.** Cont.



**Figure 8.** Process of the self-driving car making a full right-turn on the upper lane. (a) self-driving car successfully makes a full right-turn on the upper lane at time  $t$ . (b) Self-driving car collide at time  $t + 1$ . (c) Self-driving car overtime at time  $t + n$ .

At time  $t$ , it can be seen from Figure 8a that when the self-driving car met an aggressive red vehicle, it gave way to it. When it met a green conservative vehicle, it passed it, and finally, the self-driving car successfully merged into the car lane and completed the right turn.

At time  $t + 1$ , it can be seen from Figure 8b that the autonomous vehicle successfully identified the red aggressive vehicle when passing the drop-off lane and passed the green conservative vehicle when it encountered it. The unsuccessful recognition of the red aggressive vehicle at the entry lane resulted in a collision.

At time  $t + n$ , it can be seen from Figure 8c that the autonomous vehicle recognized the red aggressive vehicle when passing the drop-off lane and passed the green conservative vehicle when it encountered it; however, failed to make a judgment when it was about to enter the drop-off lane and failed to take action, resulting in a timeout. Timeout  $n$  was set to 50 s.

## 5. Conclusions

In this paper, the C-ResNet-VAE network was proposed to improve the existing deficiency of the VAE + RNN model and learn the potential characteristics of drivers from vehicle trajectories. Then, the potential characteristics and vehicle status were used to learn the trajectory prediction of autonomous vehicles at uncontrolled T-intersections. The introduction of contrast loss better learned the exclusive characteristics of the drivers from the vehicle trajectory and ensured the personalized and distinctive characteristics of the drivers; the residual network was added to the latent variable of feature extraction to improve the ability of feature extraction and prevent the gradient from disappearing. Experiments showed that the proposed method better separated the potential characteristics of drivers with different styles, received more exclusive characteristics of drivers from the trajectory, and improved the collision probability at uncontrolled intersections. However, the fuzzy driving trajectory was not successfully distinguished, and lane change and turning were not studied, which is the direction of our future research.

**Author Contributions:** Conceptualization, T.G.; Formal analysis, S.L.; Writing—original draft, T.C. and Y.C.; Writing—review & editing, H.L.; Project administration, H.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key R&D Program of China (2019YFE0108300), the National Natural Science Foundation of China (62001058, 52172379), the Fundamental Research Funds for the Central Universities (300102241201, 300102242901, 300102242806), and the Swedish Innovation Agency VINNOVA (2019-03418).

**Conflicts of Interest:** The authors declare no conflict of interest

## References

1. Qian, B.; Zhou, H.; Lyu, F.; Li, J.; Ma, T.; Hou, F. Toward collision-free and efficient coordination for automated vehicles at unsignalized intersection. *IEEE Internet Things J.* **2019**, *6*, 10408–10420. [CrossRef]
2. Sunberg, Z.N.; Ho, C.J.; Kochenderfer, M.J. The value of inferring the internal state of traffic participants for autonomous freeway driving. In Proceedings of the 2017 American control conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 3004–3010.
3. Zhan, W.; Sun, L.; Wang, D.; Shi, H.; Clausse, A.; Naumann, M.; Tomizuka, M. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv* **2019**, arXiv:1910.03088.
4. Liu, S.; Chang, P.; Liang, W.; Chakraborty, N.; Driggs-Campbell, K. Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 3517–3524.
5. Sadigh, D.; Landolfi, N.; Sastry, S.S.; Seshia, S.A.; Dragan, A.D. Planning for cars that coordinate with people: Leveraging effects on human actions for planning and active information gathering over human internal state. *Auton. Robot.* **2018**, *7*, 1405–1426. [CrossRef]
6. Brown, K.; Driggs-Campbell, K.; Kochenderfer, M.J. A taxonomy and review of algorithms for modeling and predicting human driver behavior. *arXiv* **2020**, arXiv:2006.08832.
7. Song, W.; Xiong, G.; Chen, H. Intention-aware autonomous driving decision-making in an uncontrolled intersection. *Math. Probl. Eng.* **2016**, *2016*, 1025349. [CrossRef]
8. Dong, C.; Dolan, J.M.; Litkouhi, B. Intention estimation for ramp merging control in autonomous driving. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1584–1589.
9. Bai, H.; Cai, S.; Ye, N.; Hsu, D.; Lee, W.S. Intention-aware online POMDP planning for autonomous driving in a crowd. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 454–460.
10. Morton, J.; Kochenderfer, M.J. Simultaneous policy learning and latent state inference for imitating driver behavior. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
11. Ma, X.; Li, J.; Kochenderfer, M.J.; Isele, D.; Fujimura, K. Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 6064–6071.
12. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
13. Sohn, K.; Lee, H.; Yan, X. Learning structured output representation using deep conditional generative models. *Adv. Neural Inf. Process. Syst.* **2015**, *28*.
14. Salzmann, T.; Ivanovic, B.; Chakravarty, P.; Pavone, M. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 683–700.
15. Ivanovic, B.; Leung, K.; Schmerling, E.; Pavone, M. Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach. *IEEE Robot. Autom. Lett.* **2020**, *2*, 295–302. [CrossRef]
16. Feng, X.; Cen, Z.; Hu, J.; Zhang, J. Vehicle trajectory prediction using intention-based conditional variational autoencoder. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3514–3519.
17. Bowman, S.R.; Vilnis, L.; Vinyals, O.; Dai, A.M.; Jozefowicz, R.; Bengio, S. Generating sentences from a continuous space. *arXiv* **2015**, arXiv:1511.06349.
18. Liu, S.; Chang, P.; Chen, H.; Chakraborty, N.; Driggs-Campbell, K. Learning to Navigate Intersections with Unsupervised Driver Trait Inference. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 3576–3582. Available online: <https://drive.google.com/drive/folders/1gG5Ykf9c0irOnXctPo4--255d0SM6pnd?usp=sharing> (accessed on 14 September 2021).
19. Misra, I.; Zitnick, C.L.; Hebert, M. Shuffle and learn: Unsupervised learning using temporal order verification. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 527–544.
20. Wang, X.; Gupta, A. Unsupervised learning of visual representations using videos. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 2794–2802.
21. Liu, Y.; Yan, Q.; Alahi, A. Social nce: Contrastive learning of socially-aware motion representations. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 15118–15129.
22. Xie, Z.; Liu, C.; Zhang, Y.; Lu, H.; Wang, D.; Ding, Y. Adversarial and contrastive variational autoencoder for sequential recommendation. In Proceedings of the Web Conference, Ljubljana, Slovenia, 19–23 April 2021; pp. 449–459.
23. Miao, Y.; Lei, Y.; Blunsom, P. Neural variational inference for text processing. *Comput. Sci.* **2015**, *48*, 1791–1799.
24. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *1*, 79–86. [CrossRef]

25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
26. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
27. Kostrikov, I. Pytorch Implementations of Reinforcement Learning Algorithms. 2018. Available online: <https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail> (accessed on 14 September 2021).
28. Schmerling, E.; Leung, K.; Vollprecht, W.; Pavone, M. Multimodal probabilistic model-based planning for human-robot interaction. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 3399–3406.
29. Kesting, A.; Treiber, M.; Helbing, D. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philos. Trans. R. Soc. A* **2010**, *1928*, 4585–4605. [[CrossRef](#)] [[PubMed](#)]