



Article

An Improved Multimodal Trajectory Prediction Method Based on Deep Inverse Reinforcement Learning

Ting Chen ¹, Changxin Guo ¹, Hao Li ², Tao Gao ^{1,*}, Lei Chen ³, Huizhao Tu ² and Jiangtian Yang ¹¹ School of Information Engineering, Chang'an University, Xi'an 710064, China² Key Laboratory of Road and Traffic Engineering of the Ministry of Education, College of Transportation Engineering, Tongji University, Shanghai 201804, China³ RISE Research Institutes of Sweden AB, 41756 Gothenburg, Sweden

* Correspondence: gaotao@chd.edu.cn

Abstract: With the rapid development of artificial intelligence technology, the deep learning method has been introduced for vehicle trajectory prediction in the internet of vehicles, since it provides relative accurate prediction results, which is one of the critical links to guarantee security in the distributed mixed-driving scenario. In order to further enhance prediction accuracy by making full utilization of complex traffic scenes, an improved multimodal trajectory prediction method based on deep inverse reinforcement learning is proposed. Firstly, a fused dilated convolution module for better extracting raster features is introduced into the existing multimodal trajectory prediction network backbone. Then, a reward update policy with inferred goals is improved by learning the state rewards of goals and paths separately instead of original complex rewards, which can reduce the requirement for predefined goal states. Furthermore, a correction factor is introduced in the existing trajectory generator module, which can better generate diverse trajectories by penalizing trajectories with little difference. Abundant experiments on the current popular public dataset indicate that the prediction results of our proposed method are a better fit with the basic structure of the given traffic scenario in a long-term prediction range, which verifies the effectiveness of our proposed method.

Keywords: multimodal trajectory prediction; rasterization; dilated convolution; maximum entropy inverse reinforcement learning (MaxEnt RL)



Citation: Chen, T.; Guo, C.; Li, H.; Gao, T.; Chen, L.; Tu, H.; Yang, J. An Improved Multimodal Trajectory Prediction Method Based on Deep Inverse Reinforcement Learning. *Electronics* **2022**, *11*, 4097. <https://doi.org/10.3390/electronics11244097>

Academic Editors: José D. Martín-Guerrero and Antoni Morell

Received: 28 October 2022

Accepted: 6 December 2022

Published: 8 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

Autonomous driving is an advanced stage of the vigorous development of intelligent assisted driving. As one of the popular applications of artificial intelligence, its related technologies currently have become the focus and hotspot in the field of intelligent transportation systems [1,2]. Due to the imbalanced region development and differences of user acceptance, the “autonomous + manual” mixed-driving scenario will most likely exist in the smart roads of developed cities for a long time in the future [3], which has also aroused more scholars' attention to the safety of mixed-driving scenarios [4]. As a key component of the safety planning and navigation of self-driving vehicles (SDV), accurate vehicle trajectory prediction is especially important to ensure driving safety in the internet of vehicles. Vehicle trajectory prediction refers to that SDV perceives the distributed surrounding traffic environment information through various sensing devices, and it predicts its future trajectory according to the sensed information, such as scene structure, traffic participant movements, and interaction among traffic participants [5]. Affected by many factors such as the variability of the scene structure, the diversity of traffic participants and the complexity of traffic participants interaction, the predicted trajectory is most likely multimodal, and it is always with multiple reasonable trajectories. It is really a challenging task to accurately perform multimodal trajectory prediction, which fully reveals the reasonable future behavior space for the target SDV in the mixed-driving scenario [6,7]. The existing trajectory prediction methods can be roughly divided into the following four categories:

The first category of the trajectory prediction methods is based on probability statistics. It assumes a certain correlation between the historical trajectory data and predicted trajectory data, and it constructs a mathematical model to predict the future trajectory through parameter estimation and curve fitting, including the Kalman filter-based trajectory prediction methods [8–11], the hidden Markov trajectory prediction method combined with wavelet analysis [12], the Gaussian mixture model-based trajectory prediction method [13], the Bayesian network model-based trajectory prediction method [14] and more. Although these above methods have achieved the desired prediction results, they are excessively dependent on the quality of the original data and have problems such as poor robustness and low accuracy.

The second category of the trajectory prediction methods is based on traditional neural network, which always utilizes a neural network model to learn data features of the historical trajectories and then make predictions [15–17]. Due to BP problems such as slow convergence and local minimization trap, this category of methods is more suitable for those simple prediction tasks with a relatively small data size. Moreover, in order to ensure the prediction accuracy, the training data should have strong correlation, and the appropriate structure of neural network should be selected carefully for overcoming the over-fitting or non-convergence.

The third category of trajectory prediction methods is based on deep learning, which evolved from the trajectory prediction methods based on neural network and can learn more accurate data features of historical trajectories better, including unimodal prediction [18–20] and multimodal prediction [21–31]. Unimodal prediction only outputs one most likely trajectory, but it does not explore most of the other possible trajectory space, which may usually lead to the unreliable prediction result. Multimodal prediction can fully represent the possible behavior space of the target SDV, which is more suitable for making trajectory prediction in a complex mixed-driving scenario. Several multimodal prediction methods mainly utilize generative adversarial networks (GANs) [21] or variational autoencoders (VAEs) [22,23] to generate multiple hypotheses from potential random variables by sampling; however, they treat all predictions with equal probability and do not assign a reasonable probability to each prediction. There is no doubt that some trajectories are easier to occur than other trajectories. For example, the off-road event always happens with the extremely lowest probability. To address the above critical problem, a multimodal trajectory prediction (MTP) [24] is presented to predict multiple possible trajectories of SDV along with different estimated probabilities. Since MTP can easily suffer from “mode collapse” with a single mode output, MultiPath [25] is proposed to represent the pattern using the fixed anchor obtained from the training set, and the residuals associated with the anchor come out of its regression head. As with MTP, MultiPath uses a CNN with the identical input; however, it only requires a forward inference to obtain multimodal future distributions. By contrast, CoverNet [26] has the advantage of framing the trajectory prediction problem as a classification of a set of different trajectories rather than regression, which achieve performance improvements on MultiPath. The third category of trajectory prediction methods can obtain better prediction results with higher precision as well as adapt to more complex tasks with large amounts of data. However, there are still some problems such as slow training speed, large memory consumption, difficult selection of model parameters, poor interpretability, etc.

The fourth category of trajectory prediction methods is based on a hybrid model, which can further improve the accuracy of output by integrating advantages of different existing prediction methods. Multi-head attention with joint agent-map representation (MHA-JAM) [27] is proposed to address the multimodal nature of the future trajectory by applying multiple attention, considering the joint representation of static scenes and surrounding traffic participants; each attention head can generate a different future trajectory. Different from the previous rasterization method, the cxx [28] suggests integrating a lane representation and lane attention module into a widely used encoder–decoder framework, which can sample the coordinates of each surrounding lane in real time for the neural

network to extract lane information and act as dynamic intent to produce diverse predictions without modal crashes. A graph-structured model called Trajectron [29] is presented, which can predict a number of potential future trajectories of many traffic participants synchronously in a highly dynamic and multi-mode scene. However, it only reasons about relatively simple vehicle models and past trajectory data without adding the available environmental information. Therefore, a modular, graph-structured recurrent model called Trajectron++ [30] is improved to produce dynamically feasible trajectory forecasts from heterogeneous input data, which are incorporated by distinct semantic types of multiple interacting participants. Socially Consistent and Understandable graph attention network (SCOUT) [31] is a flexible and versatile high-level representation of the scene, which is used to simulate interactions and predict the social congruent trajectories of vehicles and vulnerable road users in mixed traffic conditions. The deep inverse reinforcement learning is used for trajectory prediction [32], which uses a neural network to integrate motion and environment to update the reward function.

The above-mentioned methods have achieved better performance, and especially some methods based on hybrid models with deep learning have made great progress. However, the traffic scenes' context always contains rich feature information including a series of past states from a single SDV to its all surrounding vehicles, and high-definition map information, but the existing methods still have the weaknesses of inadequate feature extraction and loss of contextual feature information as well as failure to fully and effectively utilize the feature information of the traffic scene. Therefore, there is still the possibility to further improve the accuracy and robust for trajectory prediction. In this paper, an improved multimodal trajectory prediction method based on deep inverse reinforcement learning has been proposed, and the main contributions are as follows:

- (1) It is very necessary to accurately extract map features and historical sequence information, since it always directly affects the downstream feature analysis and trajectory prediction. In this paper, a fused dilated convolution module is introduced into the existing multimodal trajectory prediction network, which can make a streamlined improvement by expanding the perceptual field without ignoring local information of the traffic scene and retaining the same or even higher generalizability compared with the original network.
- (2) Since the inverse reinforcement learning policy extracts the reward function from expert presentation data, which can effectively solve the problem of the complexity and difficulty of setting the reward function manually, in this paper, an improved MaxEnt RL policy with inferred goals is applied into the existing multimodal trajectory prediction network, which can alleviate the need for a predefined goal state and induce distribution on possible goals.
- (3) It is very crucial to design a reasonable and effective sampling function that not only affects the optimization process of the neural network but also determines the effective utilization of feature information extracted from the dataset. In this paper, a proposed correction factor is added into the existing multimodal trajectory prediction network, which can encourage the generation of diverse trajectories by penalizing pairwise distances with small differences, and this is more in line with the multimodal characteristics of future trajectories.

In conclusion, our proposed improved method can more accurately predict the future behavioral intention and trajectory distribution of surrounding vehicles by considering the multimodal characteristics of vehicle behavior and trajectory and generating multimodal trajectory predictions that conform to the scene structure. Especially, the premise of this paper's research is that high-definition semantic map information as well as surrounding vehicle motion information are available. The trajectory prediction is performed within a scene information focusing on a range of about 50 m. In some complex mixed traffic scenarios where road testing facilities are not perfect or in-vehicle devices are scarce, the method may ignore some scene element information and make unreliable predictions. This paper is organized as follows: the background of trajectory prediction methods and

our contributions are presented in Section 1; the preliminary work associated with the multimodal trajectory prediction method is described in Section 2; Section 3 gives a detailed description of our proposed method; Section 4 verifies the effectiveness of the proposed method by abundant experiment analyses; and our conclusion and the future work are presented in Section 5.

2. Preliminaries

In this section, several related works associated with the multimodal trajectory prediction are briefly reviewed as follows.

2.1. Rasterization for the Traffic Scene

Rasterization for the traffic scene can rasterize high-definition maps and surroundings as well as the estimated state of each traffic participant in the SDV's vicinity, thus providing complete context and the information necessary for the next two-stage backbone. The rasterization for the traffic scene depicted in Figure 1 was first proposed by [33] and then widely used in trajectory prediction as the input of the training networks [24–27]. The input dynamic context relates to the state estimates include the following historical sequence information that describes each of the traffic participants: position, velocity, acceleration, heading, heading change rate, etc. The input static context relates to the mapping data of an area from the high-definition map where the SDV is operating, comprising road and crosswalk polygons, as well as lane directions and boundaries. Moreover, raster images with motion information are often used for visualization.

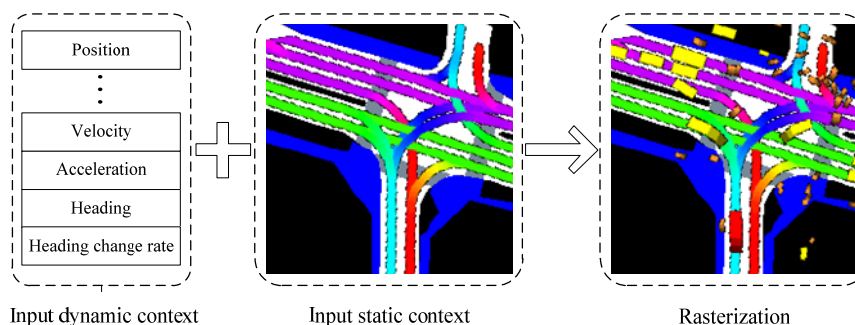


Figure 1. Rasterization for the traffic scene.

2.2. Reward Initialization Based on Two-Stage Backbone

In order to effectively integrate dynamic and static context, the two-stage backbone is usually utilized to approximate the potential reward function [24,26,32]. It can learn a mapping from local features of the traffic rasterized scene for rewarding on a 2D grid. Taking advantage of the equivariance of convolutional layers, the reward model can be transferred to new scenes, which are configured with different scene elements.

Reward initialization based on a two-stage backbone is shown in Figure 2, where the static context is the input to the first stage, and the generated feature maps are concatenated with the motion features extracted from dynamic context. The proximate reward is the output of the second stage. The first stage of the CNN usually consists of an ImageNet pretraining block of ResNet-34 [34] as an extractor utilized to extract scene features, and it manipulates the aerial view I of the static environment around the SDV:

$$\phi_I = CNN_{\text{first}}(I) \tag{1}$$

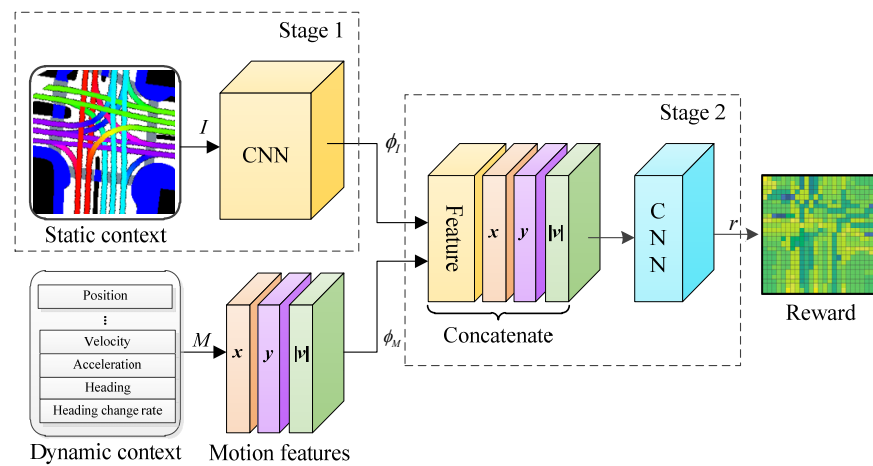


Figure 2. Reward initialization based on two-stage backbone.

The scene feature ϕ_I have the same dimensions as the 2-D grid, matching to state space S . Additionally, the rewards are not only depended on the static context about scene features, but also on the dynamic context about motion features of SDV. The motion features ϕ_M can be obtained by combining the scene features, SDV’s motion information and grid cells position [32]:

$$\phi_M = [|v|, x, y] \tag{2}$$

where $|v|$ means the speed of SDV. x and y are the position of each grid cell in the SDV reference, with the point of origin at the SDV’s current position and the positive direction of the x -axis denoting the SDV’s current motion direction.

The second stage of the CNN usually uses a full convolutional block, which can map the scene features and motion features to reward:

$$r = CNN_{second}(\phi_I, \phi_M) \tag{3}$$

Although the two-stage backbone has achieved better results in recent studies, simple convolutional pooling adopted in CNN_{first} may have the weakness of incomplete feature extraction, tending to reduce feature extraction accuracy of the local information and then ignore the effect of edge information on interactions between the target SDV and its surrounding vehicles, which may further lead to subsequent unreliable predictions.

2.3. Reward Updating Based on MaxEnt RL Policy

Reward updating based on the MaxEnt RL policy is shown in Figure 3, which can deduce and update the reward learned by a two-stage backbone and display different plausible paths generated on the 2D grid.

The MaxEnt RL policy consists of two main algorithms [32]. One algorithm is the approximate value iteration, which solves the current reward function and targets the state received from the upper MaxEnt RL policy. The other algorithm is policy propagation, which involves repeatedly calculating the state visitation frequencies (SVF) at each step of the policy generation process. For the convergent reward model r , it can be sampled from it to give predictions from the initial state to the target path on the 2D grid. Since the policy is stochastic, there are multiple trajectories to reach the goal state.

Given the state sequences $s^{(i)}$, which sampled from the MaxEnt RL policy and represents the i th sample plans:

$$s^{(i)} = [s_1^{(i)}, s_2^{(i)}, \dots, s_N^{(i)}] \tag{4}$$

The sampled plans trained to reach different plausible goal states S , and the state space S is given by the following:

$$S = [s^{(1)}, s^{(2)}, \dots, s^{(i)}] \tag{5}$$

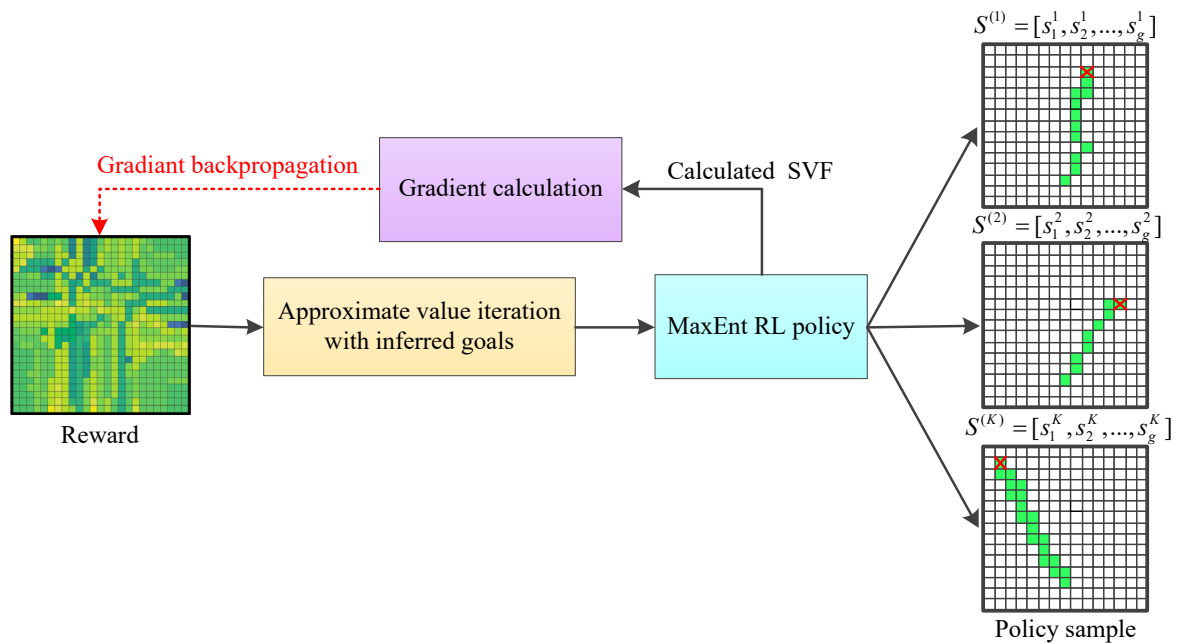


Figure 3. Reward updating based on MaxEnt RL policy.

The generation of existing MaxEnt RL policy relies on a predefined goal state, but these goals are often required to be inferred and are unknown in most occasions.

2.4. Trajectory Generation Based on Attention Mechanism

The structure of the trajectory generator based on an attention mechanism is a soft attention-guided recurrent neural network encoder–decoder [35]. There are four sub-components, including a motion encoder, plan encoder, attention-based decoder and sampling and clustering, as shown in Figure 4.

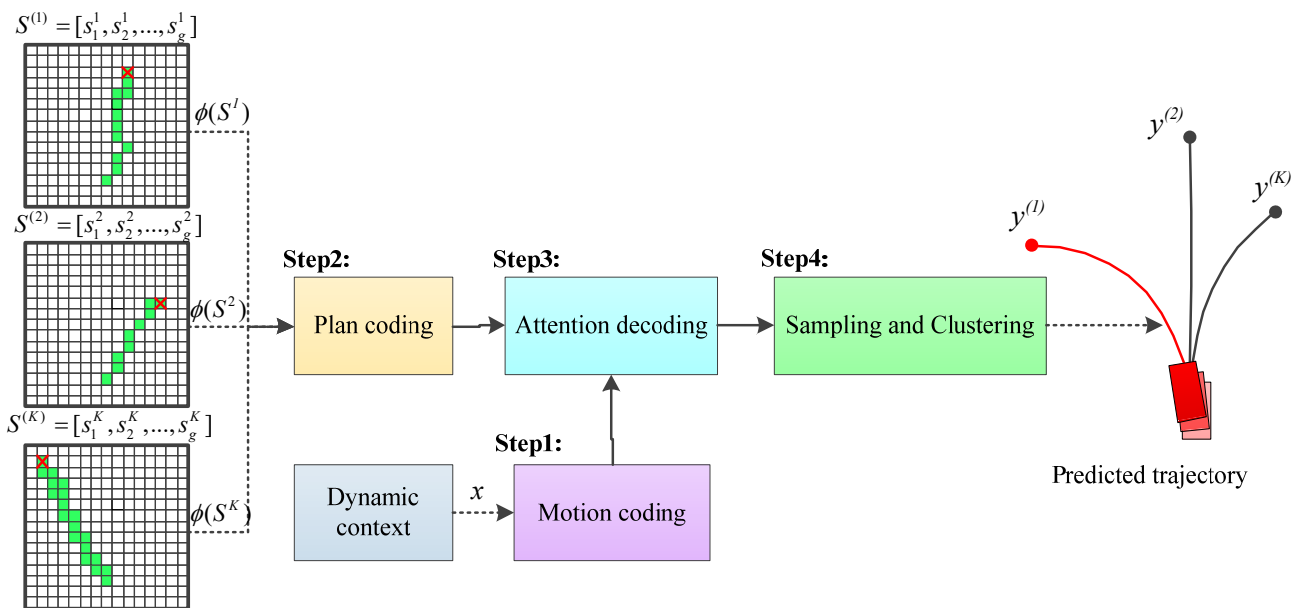


Figure 4. Attention-based trajectory generator.

For the first step, the motion encoder can utilize a gate recurrent unit (GRU) encoder to encode the track history x , which is a snippet of the SDV's nearest track history with time T_h .

$$x = [x_{-T_h}, \dots, x_t, \dots, x_1, x_0] \quad (6)$$

where x represents the dynamics of the SDV's motion, x_t maps the original position, velocity, acceleration and yaw-rate of SDV, t denotes the forecast time, and the prediction moment $t = 0$.

For the second step, the plan encoder can utilize a bidirectional GRU (Bi-GRU) encoder to aggregate status codes on the whole plan, which contains part of the scene feature, surrounding SDV states and the sampling state sequences. The outputs ϕ_1 of CNN_{first} from two-stage backbone is used as the state encodings. For surrounding SDV state, their grid positions are populated with each of the SDV's motion information. For each state $s_n^{(i)}$ in a sampling plan $s^{(i)}$, context of the scene, SDV states and position coordinates at mesh elements are embedded to $s_n^{(i)}$, the outputs are concatenated to obtain state encoding $\phi_s = [s_n^{(i)}]$.

For the third step, the output trajectories $y^{(i)}$ are generated by a GRU decoder based on the soft attention mechanism. The decoder can focus on a specific states of the sampled plan $s^{(i)}$ when generating trajectories by plan. Therefore, the decoder can focus only the earlier states of the sampling plan as it generates a slow moving SDV's future trajectory. It can also focus on the later states as well as generate a future trajectory with a fast moving SDV.

After a series of operations of motion encoding, plan encoding and attention-based decoding, the sampled trajectories conditioned on the sampled plans are generated at the fourth step. However, since sampling by itself is inefficient, it may generate several identical or very similar sampled state sequences or trajectories. In order to obtain a concise trajectory distribution, the K-means algorithm is usually used to further cluster the sampled trajectories. The existing clustering algorithm [24,36] often iterates from randomly selecting k centroids repeatedly until convergence, and the results may have a great randomness. Each calculation will always lead to a different result because the initial randomly selected central masses are different. Moreover, when the amount of data is too large, there is the problem of relatively low time efficiency.

3. Proposed Method

To address the above problems, an improved multimodal trajectory prediction method based on deep inverse reinforcement learning is proposed by introducing a fused dilated convolution module into the convolutional reward policy component, improving the reward update module by learning the state rewards of goals and paths separately as well as adding correction factors for the function of the clustering algorithm in the sampling and clustering module of the attention-based trajectory generator. Our proposed method can expand the perceptual field without losing local information and further generate multiple predictions which are closer to the real trajectory.

3.1. Architecture

The architecture of our proposed multimodal trajectory prediction method based on deep inverse reinforcement learning is given in Figure 5. There are four components, including the rasterization for the traffic scene, the reward initialization based on a two-stage backbone, the reward updating based on MaxEnt RL policy and the trajectory generation based on an attention mechanism.

The rasterization for the traffic scene is illustrated in Figure 5a, which consists of the scene dataset as the input of the overall model framework, the rasterization module and the status input module. The rasterization module extracts the static scene layout information of the dataset as raster images, and the state input module extracts the motion features of the dynamic scene in the form of vectors, both of which are used as inputs for the next part.

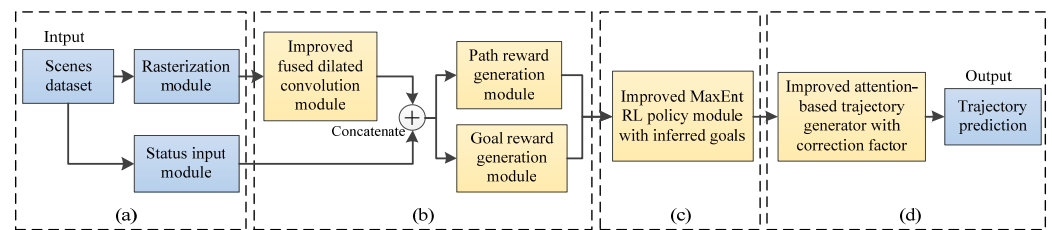


Figure 5. Architecture of the improved multimodal trajectory prediction method. (a) Rasterization for the traffic scene; (b) Reward initialization based on a two-stage backbone; (c) Reward updating based on the MaxEnt RL policy; (d) Trajectory generation based on the attention mechanism.

Figure 5b shows the reward initialization based on a two-stage backbone, in which the first stage uses the improved fused dilated convolution module to extract context features from raster images and concatenates with state information as input to the second stage. The second stage uses a path reward module and a goal reward module instead of a single reward module.

The reward updating based on the MaxEnt RL policy given in Figure 5c first learns the reward for the path and goal states conditioned on the historical trajectory of the surrounding vehicles as well as a policy unconstrained by the predefined goal state. Then, paths to different reasonable goals are generated in a 2D grid according to the policy sampling. We use each state sequence sampled from the policy as a plan and input it to the next section.

The trajectory generation based on an attention mechanism is shown in Figure 5d, which adopts a recurrent neural network encoding–decoding method to first encode the motion features and scene information, and the continuous-valued trajectory conditioned on the sampling plan is output by the trajectory generator improved by adding correction factors.

3.2. Improved Fused Dilated Convolution Module

Since using raster images as the input to the neural network requires a relatively large perceptual field to aggregate contextual information, the ordinary convolution block reduces the accuracy of local information and ignores the effect of edge information on the interaction of the SDV with surrounding vehicles, which may lead to unreliable predictions. To overcome the incomplete information extraction of raster images by CNN_{first} in Section 2.2, an improved dilated convolution block is introduced into the original fully convolutional reward policy module, and we replace the existing CNN_{first} block. Dilated convolution [37] is a special convolution structure that can achieve an expanded field of perception by adding the number of voids between the elements of the convolution kernel. As in Figure 6, a 5×5 image is convolved twice with 3×3 ordinary convolution, and a 1×1 feature map is obtained after convolution. As in Figure 7, a 3×3 dilated convolution with the expansion factor of 2 convolves the original 5×5 image once a 1×1 feature map is also obtained. It can be seen that the dilated convolution achieves a large perceptual field using a small number of parameters.

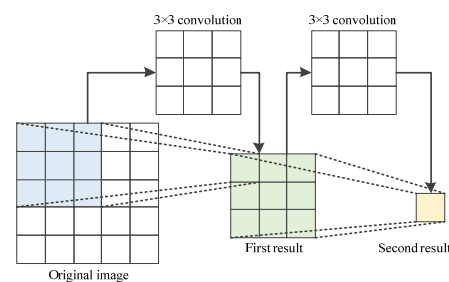


Figure 6. Process of 3×3 ordinary convolution twice.

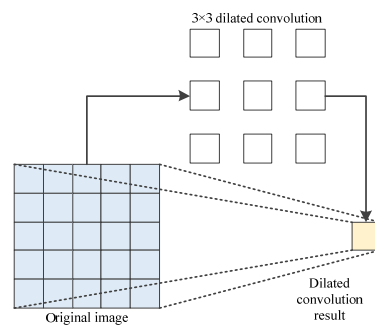


Figure 7. Process of 3×3 dilated convolution with the expansion factor of 2.

For the better deployment of deep neural networks to self-driving cars, it is necessary to reduce the number of model parameters and decrease the complexity of the network. Inspired by the design ideas of lightweight neural networks such as MobileNet [38] and SqueezeNet [39], 1×1 pointwise convolution has the property of equalizing the receptive field, and it is commonly used for the operation of neural network dimensional variation, with the characteristics of less parameters and less computation. We combine the dilated convolution with 1×1 pointwise convolution to generate an improved dilated convolution lightweight module. It is worth noting that too much use of dilated convolution to lighten the neural network may bring a large loss and lose some feature information. Therefore, an improved fused dilated convolution structure can be obtained by fusing the proposed dilated convolution block with the original residual convolution block, as shown in Figure 8. It may obtain the lightweight effect without increasing the loss of accuracy as well as achieving better trade-off between speed and accuracy.

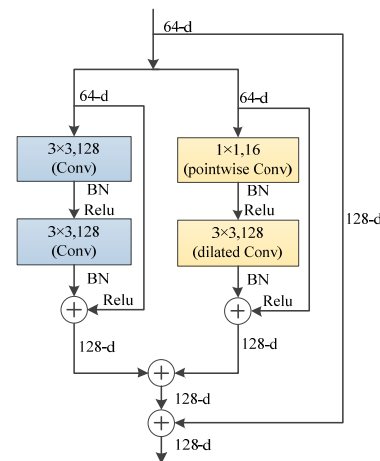


Figure 8. Improved fused dilated convolution module.

3.3. Improved MaxEnt RL Policy Module with Inferred Goals

We would like to relax the requirement of pre-defined goals in the MaxEnt RL in order to make the policy generalizable to various scenarios. The MaxEnt RL policy should explore potential goal states instead of terminating at the pre-defined goals. Therefore, we propose sample paths that end at different goals in the scenario by learning path and goal state rewards.

We divide the second stage full convolution module in Section 2.2 into two parts, learning the goal reward and the path reward, respectively. Specifically, the goal reward module and the path reward module have the same architecture consisting of three 1×1 convolutional layers. As shown in Figure 9, there are two layers with depth 32 and the one layer with depth 1 to output a single corresponding reward. Then, the log-sigmoid activation is utilized for the output, limiting the reward values between $-\infty$ and 0.

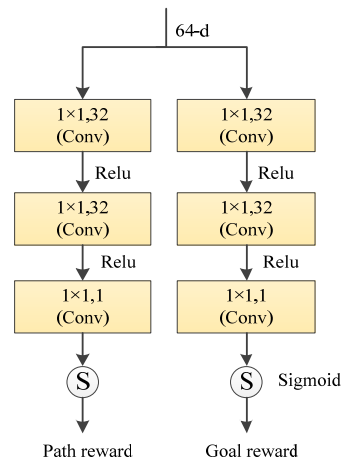


Figure 9. Path and goal reward module.

Due to the large amount of model data, the execution time may be very high. With the emergence of parallel and distributed simulation ways, it uses the considerable memory and high processing power of multiple execution units to effectively handle large-scale simulation [40]. Inspired by this, we use the stochastic parallel gradient descent algorithm to calculate the back propagation gradient by computing the state visitation frequency. The core idea of the parallel gradient descent algorithm is to use multiple processors to compute the gradient using their own data separately, select the negative gradient direction of the objective function as the search direction for each iteration step, and finally implement parallel computation of gradient descent by aggregation or other means to accelerate the model training process.

3.4. Improved Attention-Based Trajectory Generator with Correction Factor

Most of the existing methods [24,36] often utilize the ordinary Euclidean distance formula to accomplish a clustering task, but it cannot work well to address the problem described in Section 2.4. To alleviate the redundancy for the sampling process and avoid being prone to produce similar or duplicate trajectories, we apply the diversity sampling technique Dlow [41] to our trajectory sampling, which can improve the diversity of trajectory sampling and avoid similar samples due to random sampling. Our proposed loss function with the addition of correction factors is given in the following equation:

$$\begin{aligned}
 Loss = & \min_{k \in \{1, \dots, K\}} \frac{1}{T_f} \sum_{t=1}^{T_f} \|y_t^{GT} - \hat{y}_t^{(k)}\|_2 \\
 & + \frac{1}{K(K-1)} \sum_{k_1=1}^K \sum_{k_1 \neq k_2}^K \exp\left(-\frac{\|\hat{y}^{(k_1)} - \hat{y}^{(k_2)}\|_2}{\sigma_d}\right)
 \end{aligned} \tag{7}$$

where y^{GT} is the ground true future trajectory of the target SDV and σ_d is a scaling factor. The first term is the minimum mean displacement error to let the sampled trajectories $\hat{y}^{(k)}$ be close to trajectories y^{GT} . We add a second term to penalize trajectories with semblable trajectories to encourage the generation of diverse sampling trajectories. The prediction trajectory with smaller similarity takes a larger proportion in the whole loss function; we obtained diversity trajectory by training the smaller loss function.

To speed up the convergence of the trajectory generator, we pretrain the module with minimizing the average displacement error between the ground real and forecast trajectories. Specifically, we train the trajectory generator using Adam [42] with a learning rate of 0.0001.

4. Experimental Analysis

The experimental hardware configuration and other environment settings are shown in Table 1.

Table 1. Experimental hardware and environment settings.

Parameter Name	Parameter Content
Computer model	Lenovo 30BBA8M0CW desktop computer
Operating system	Windows 10 Professional 64-bit
CPU	Intel Xeon Gold5118 @ 2.30 GHz (X2)
GPU	Nvidia Quadro RTX 5000 (16 GB)
Python	3.8.8
Pytorch	1.10.0
CUDA	10.2
cuDNN	7.0

4.1. Dataset

In order to verify the effectiveness of our improved method, the relevant metrics are evaluated on the public dataset. Table 2 shows the common datasets for trajectory prediction, and we finally select the current mainstream public dataset nuScenes [43] released in 2019. The nuScenes dataset contains 1000 complex urban traffic scenes captured by in-vehicle cameras and LiDAR sensors passing through Boston and Singapore, which are given in Figure 10. Each scene was recorded in approximately 20 s, has 40 keyframes annotated at 2 Hz, and contains up to 23 semantic objects as well as 11 instances. In this paper, the official split benchmark of the nuScenes prediction challenge is used to train and evaluate our proposed method, with 32,186 prediction instances in the training set, 8560 instances in the validation set, and 9041 instances in the test set.

Table 2. Trajectory prediction related datasets.

Dataset	Release Time	Data Scale	Perspective	Scenes	Remark
NGSIM	2006.12	4 scenes, 9206 vehicles, 5071 km driving distance, 174 h total recording time	top view	highway, city road	It used to be the most popular dataset in this field, but the research found that there are problems such as insufficient precision and coordinate drift, and it is rarely used at present.
ETH	2008.6	3 video clips, 1804 images	vehicle view	city road	A classic pedestrian dataset, suitable for computer vision tasks and social behavior modeling.
Stanford Drone	2016.8	8 scenes, 19,000 targets, 185,000 tagged target interaction messages	top view	campus	Video taken by drones across the Stanford campus, including pedestrians, vehicles, bicycles and other traffic participants.
HighD	2018.10	6 scenes, 110,000 vehicles, 45,000 km driving distance, 447 h total recording time	top view	highway	A large dataset of natural vehicle trajectories on German highways, suitable for driver model parameterization, autonomous driving, and traffic pattern analysis.
nuScenes	2019.3	1000 scenes, 1.4 million camera images, 390,000 LiDAR scans, 1.4 million radar scans	vehicle view	city road	The first publicly available full-sensor dataset, large enough for research on sensor suites. It is widely used in various fields of autonomous driving.

4.2. Metrics

Similar to previous multimodal trajectory prediction methods [24–33], we evaluate our method by using a series of common evaluation metrics from the nuScenes Official Challenge: E_{ave} means the minimum average displacement error, E_{final} represents the minimum final displacement error, R_{miss} is the miss rate, and R_{off} is the off-road rate.

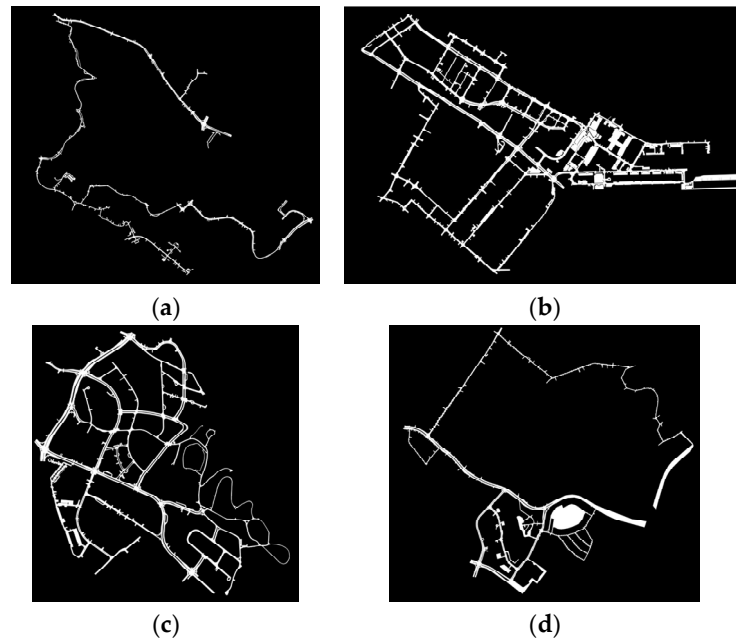


Figure 10. High-definition semantic maps of the nuScenes dataset. (a) Singapore Queenstown; (b) Boston Seaport; (c) Singapore One North; (d) Singapore Holland Village.

E_{ave} and E_{final} for a sample of K trajectory predictions for the target SDV are respectively given by:

$$E_{ave(K)} = \frac{1}{T} \min_{k=1}^K \sum_{t=1}^T \|\hat{y}_t^{(k)} - y_t^{GT}\|_2 \quad (8)$$

$$E_{final(K)} = \min_{k=1}^K \|\hat{y}_t^{(k)} - y_t^{GT}\|_2 \quad (9)$$

where E_{ave} is defined as the Euclidean distance (i.e., 2-Norm) between the true and predicted trajectories.

For a given undetected distance d and the predicted K most likely future trajectories, the missing detection determination formula is presented in Equation (10):

$$R_{miss(K,d)} = \begin{cases} 1 & \min_{k=1}^K \left(\max_{t=1}^{T_f} \|\hat{y}_t^{(k)} - y_t^{GT}\| \right) \geq d \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Furthermore, R_{off} is used to measure the proportion of predicted tracks falling outside the drivable area of the map [44].

4.3. Experimental Comparison and Analysis

We compare our proposed method with several baseline methods that represent the latest techniques in multimodal trajectory prediction. As shown in Table 3, we list the results considering each method to generate the K most probable trajectories.

Specifically, we evaluated $E_{ave(K)}$ and $E_{final(K)}$ at $K = \{5,10\}$, $R_{miss(K,d)}$ at $K = \{5,10\}$ $d = 2$, and R_{off} . We compared the proposed model with other existing methods, and according to the different assessment methods, there may be large differences in the ranking method.

Table 3. Comparative analysis on nuScenes dataset over a prediction horizon of 6-s.

Method	$E_{ave(5)}$	$E_{ave(10)}$	$E_{final(5)}$	$E_{final(10)}$	$R_{miss(5,2)}$	$R_{miss(10,2)}$	R_{off}
Physics oracle [26]	3.69	3.69	9.06	9.06	0.91	0.91	0.12
CoverNet [26]	2.62	1.92	-	-	0.76	0.64	0.13
MTP [24]	2.44	1.57	4.83	3.54	0.70	0.55	0.11
M-SCOUT [31]	1.92	1.92	-	-	0.78	0.78	0.10
Trajectron++ [30]	1.88	1.51	-	-	0.70	0.57	0.25
Multipath [25]	1.78	1.55	3.62	2.93	0.75	0.74	0.36
MHA-JAM [27]	1.81	1.24	3.72	2.21	0.59	0.45	0.07
cxx [28]	1.63	1.29	-	-	0.69	0.60	0.08
Ours	1.49	1.13	3.06	2.06	0.64	0.45	0.03

Compared with the previous physical oracle model, the classical MTP method has been significantly improved in various indicators. With the popularity of vehicle trajectory prediction research, more and more scholars participate in it. Our proposed method performs better than other methods in six of the seven reported metrics. MHA-JAM achieves the best performance result in the metric of $R_{miss(5,2)}$, which shows the effectiveness of the multi-head attention with joint agent-map representation method. The metric $R_{miss(10,2)}$ has the similar effect as the MHA-JAM method, which illustrates that our method can generate diverse trajectories. The lower R_{miss} indicates that our predicted trajectory is unlikely to diverge from the true trajectory within the $d = 2$ m threshold range. The metrics of $E_{final(5)}$ and $E_{final(10)}$ are also reported, respectively, which reveal that our method is superior to other baseline methods by indicating a better prediction of the goal path of the predicted trajectory. In addition, our method possesses a significantly lower value of R_{off} , which affects predictions outside the drivable area. Therefore, our method can generate better trajectory predictions that are more consistent with the basic structure of the traffic scenario.

Several visualization examples of our proposed method from the nuScenes dataset are shown in Figure 11, in which Figure 11a is the raster image input. In order to distinguish the lane direction, lanes are given polygons of different colors, with red rectangles representing the target vehicle and yellow rectangles representing the surrounding vehicles. Since we are interested in the vehicle's historical trajectory, the boundary frames captured in continuous time steps are rasterized at the map vector layer. Each historical character polygon is rasterized using the same color as the current polygon, but the brightness level is reduced to create a render effect. Figure 11b,c are path state visitation frequencies images and target state visitation frequencies images under the maximum entropy policy, respectively. The higher the brightness, the greater the visitation frequency. The aggregated trajectories finally generated by the trajectory generator are shown in Figure 11d, with the red trajectory representing prediction trajectories and the black trajectory representing the real future trajectory. The prediction does not reflect the collision. Singapore, where the dataset is collected, drives on the left, while Boston, another collection place, drives on the right as in China. Therefore, we infer that the first scene in the visual vehicle trajectory prediction is collected by Boston, and the other four scenes are collected by Singapore. It can be found that for different scenario configurations, the MaxEnt policy explores plausible paths and goal states in a 2D grid. The predicted trajectories closely correspond to the state of policy exploration and can generate a set of predictions compatible with various scenario configurations.

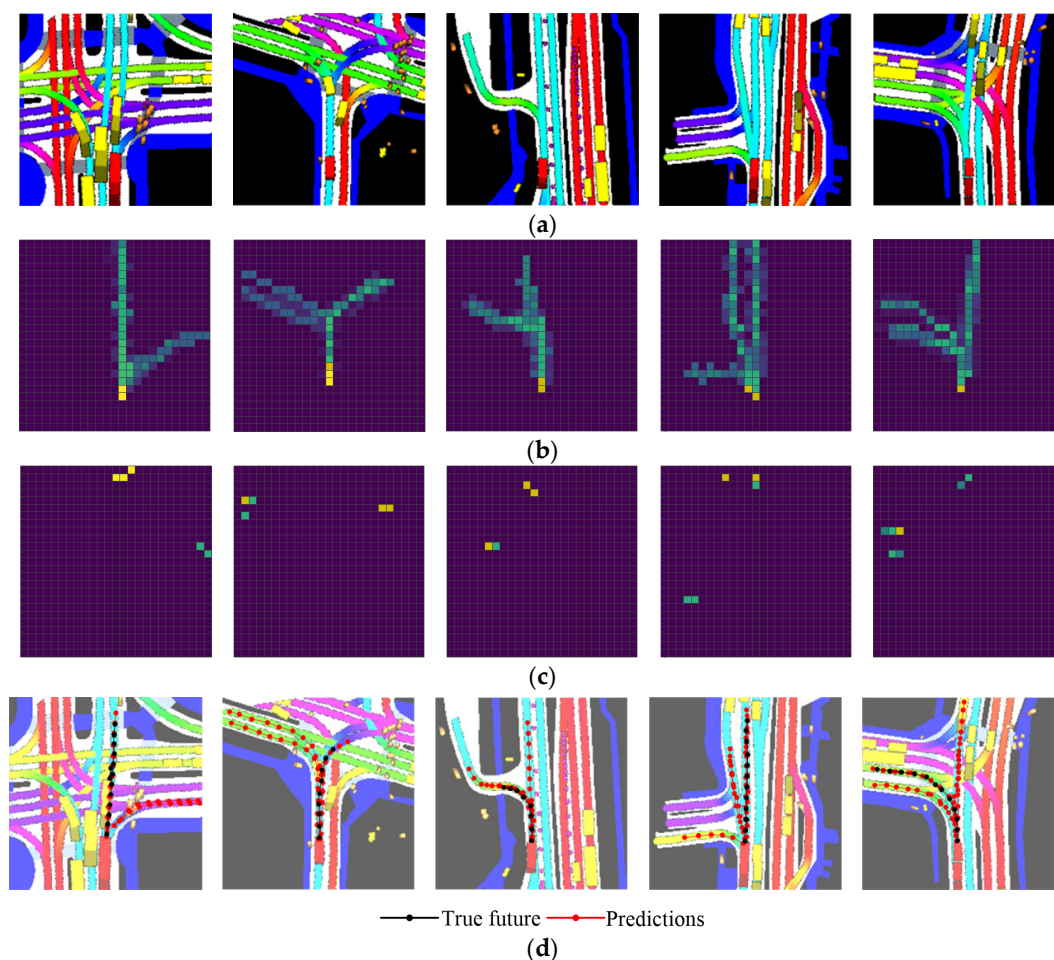


Figure 11. Visualization examples from nuScenes. (a) Raster image input; (b) Path SVF images; (c) Goal SVF images; (d) Predicted trajectories.

5. Conclusions

In this paper, we propose an improved multimodal trajectory prediction method based on deep inverse reinforcement learning. The validation results on the publicly available nuScenes dataset show that our method can fully consider the scene contextual features and generate a variety of trajectory predictions that can match the basic structure of the scene better. Compared with the existing baseline methods, our method can obtain good results for several metrics. In particular, the R_{off} metric is significantly better than other methods, which emphasizes that the predicted future trajectories more conform to the scene structure.

Although our method utilizes scene rasterization as the input scene information of the neural network for feature extraction and achieves better results in trajectory prediction, scene rasterization may suffer from inefficient coding, long training time, and a loss of connection information due to occlusion. Exploring the use of graph neural networks to represent traffic scene information with fewer parameters to achieve advanced performance will be part of our future work.

Author Contributions: Funding acquisition, T.G.; investigation, T.C., T.G., L.C. and J.Y.; methodology, T.C. and T.G.; project administration, T.C.; resources, C.G.; software, C.G.; supervision, H.T.; writing—original draft, C.G.; writing—review and editing, T.C., H.L. and L.C. 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. Wang, J.; Huang, H.; Zhi, P.; Sheng, Z.; Zhou, G. Review of development and key technologies in automatic driving. *Appl. Electron. Tech.* **2019**, *6*, 28–36. [[CrossRef](#)]
2. Meng, X.; Zhang, C.; Su, C. Review of key technologies of autonomous vehicle systems. *Auto Time* **2019**, *17*, 4–5.
3. Pei, Y.; Chi, B.; Lv, J.; Yue, Z. An overview of traffic management in “automatic + manual” driving environment. *J. Transp. Inf. Saf.* **2021**, *5*, 1–11.
4. Gehlot, A.; Singh, R.; Kuchhal, P.; Kumar, A.; Singh, A.; Alsubhi, K.; Ibrahim, M.; Villar, S.G.; Brenosa, J. WPAN and IoT Enabled Automation to Authenticate Ignition of Vehicle in Perspective of Smart Cities. *Sensors* **2021**, *21*, 7031. [[CrossRef](#)] [[PubMed](#)]
5. Wang, K.; Wang, Y.; Deng, X.; Huang, Q.; Liao, K. A review on the study of impact of uncertainty on vehicle trajectory prediction. *Automob. Technol.* **2022**, *7*, 1–14. [[CrossRef](#)]
6. Liu, W.; Hu, K.; Li, Y.; Liu, Z. A review of prediction methods for moving target trajectories. *Chin. J. Intell. Sci. Technol.* **2021**, *2*, 149–160.
7. Leon, F.; Gavrilescu, M. A review of tracking and trajectory prediction methods for autonomous driving. *Mathematics* **2021**, *6*, 660. [[CrossRef](#)]
8. Prevost, C.G.; Desbiens, A.; Gagnon, E. Extended Kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 1805–1810.
9. Barth, A.; Franke, U. Where will the oncoming vehicle be the next second? In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 1068–1073.
10. Qiao, S.; Han, N.; Zhu, X.; Shu, H.; Zheng, J.; Yuan, C. A dynamic trajectory prediction algorithm based on Kalman filter. *Acta Electron. Sin.* **2018**, *02*, 418–423.
11. Vashishtha, D.; Panda, M. Maximum likelihood multiple model filtering for path prediction in intelligent transportation systems. *Procedia Comput. Sci.* **2018**, *143*, 635–644. [[CrossRef](#)]
12. Zhang, X.; Liu, G.; Hu, C.; Ma, X. Wavelet analysis based hidden Markov model for large ship trajectory prediction. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 2913–2918.
13. Lim, Q.; Johari, K.; Tan, U.X. Gaussian process auto regression for vehicle center coordinates trajectory prediction. In Proceedings of the TENCON 2019-2019 IEEE Region 10 Conference (TENCON), Kochi, India, 17–20 October 2019; pp. 25–30.
14. Wang, L.; Song, T. Ship collision trajectory planning and prediction for inland waterway. *J. Hubei Univ. Technol.* **2019**, *2*, 64–68.
15. Yang, C. Research on the trajectory prediction method based on BP neural network. *Pract. Electron.* **2014**, *20*, 22. [[CrossRef](#)]
16. Yang, B.; He, Z. Hypersonic vehicle track prediction based on GRNN. *Comput. Appl. Softw.* **2015**, *7*, 239–243.
17. Gao, T.; Xu, L.; Jin, L.; Ge, B. Vessel trajectory prediction considering difference between heading and data changes. *J. Transp. Syst. Eng. Inf. Technol.* **2021**, *1*, 90–94. [[CrossRef](#)]
18. Djuric, N.; Radosavljevic, V.; Cui, H.; Nguyen, T.; Chou, F.C.; Lin, T.H.; SINGH, N.; Schneider, J. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 2–5 May 2020; pp. 2095–2104.
19. Li, X.; Ying, X.; Chuah, M.C. Grip: Graph-based interaction-aware trajectory prediction. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3960–3966.
20. Luo, W.; Yang, B.; Urtasun, R. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3569–3577.
21. Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; Alahi, A. Social gan: Socially acceptable trajectories with generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 2255–2264.
22. Lee, N.; Choi, W.; Vernaza, P.; Choy, C.B.; Torr, P.H.; Chandraker, M. Desire: Distant future prediction in dynamic scenes with interacting agents. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2018; pp. 336–345.
23. Tang, C.; Salakhutdinov, R.R. Multiple futures prediction. *Adv. Neural Inf. Processing Syst.* **2019**, *32*, 1–11.
24. Cui, H.; Radosavljevic, V.; Chou, F.C.; Lin, T.H.; Nguyen, T.; Huang, T.K.; Schneider, J.; Djuric, N. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2090–2096.
25. Chai, Y.; Sapp, B.; Bansal, M.; Anguelov, D. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv* **2019**, arXiv:1910.05449.

26. Phan-Minh, T.; Grigore, E.C.; Boulton, F.A.; Beijbom, O.; Wolff, E.M. Covernet: Multimodal behavior prediction using trajectory sets. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 14074–14083.
27. Messaoud, K.; Deo, N.; Trivedi, M.M.; Nashashibi, F. Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 165–170.
28. Luo, C.; Sun, L.; Dabiri, D.; Yuille, A. Probabilistic multi-modal trajectory prediction with lane attention for autonomous vehicles. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 2370–2376.
29. Ivanovic, B.; Pavone, M. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2375–2384.
30. 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 (ECCV), Online, 23–28 August 2020; pp. 683–700.
31. Carrasco, S.; Llorca, D.F.; Sotelo, M.A. SCOUT: Socially-consistent and undersTandable graph attention network for trajectory prediction of vehicles and VRUs. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 1501–1508.
32. Zhang, Y.; Wang, W.; Bonatti, R.; Maturana, D.; Scherer, S. Integrating kinematics and environment context into deep inverse reinforcement learning for predicting off-road vehicle trajectories. *arXiv* **2018**, arXiv:1810.07225.
33. Djuric, N.; Radosavljevic, V.; Cui, H.; Nguyen, T.; Chou, F.C.; Lin, T.H.; Schneider, J. Short-term motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv* **2018**, arXiv:1808.05819.
34. 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, 26 June–1 July 2016; pp. 770–778.
35. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
36. Ahmed, M.; Seraj, R.; Islam, S.M.S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics* **2020**, *8*, 1295. [[CrossRef](#)]
37. Liu, J.; Li, C.; Liang, F.; Lin, C.; Sun, M.; Yan, J.; Ouyang, W.; Xu, D. Inception convolution with efficient dilation search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 11486–11495.
38. Sinha, D.; El-Sharkawy, M. Thin mobilenet: An enhanced mobilenet architecture. In Proceedings of the 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 10–12 October 2019; pp. 280–285.
39. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
40. Ibrahim, M.; Iqbal, M.A.; Aleem, M.; Islam, M.A.; Vo, N.S. MAHA: Migration-based adaptive heuristic algorithm for large-scale network simulations. *Clust. Comput.* **2020**, *2*, 1251–1266. [[CrossRef](#)]
41. Yuan, Y.; Kitani, K. Dlow: Diversifying latent flows for diverse human motion prediction. In Proceedings of the European Conference on Computer Vision (ECCV), Online, 23–28 August 2020; pp. 346–364.
42. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
43. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 11621–11631.
44. Greer, R.; Deo, N.; Trivedi, M. Trajectory prediction in autonomous driving with a lane heading auxiliary loss. *IEEE Robot. Autom. Lett.* **2021**, *3*, 4907–4914. [[CrossRef](#)]