A Survey on Trajectory-Prediction Methods for Autonomous Driving

Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen, *Senior Member, IEEE*

***Abstract*—In order to drive safely in a dynamic environment, autonomous vehicles should be able to predict the future states of traffic participants nearby, especially surrounding vehicles, similar to the capability of predictive driving of human drivers. That is why researchers are devoted to the field of trajectory prediction and propose different methods. This paper is to provide a comprehensive and comparative review of trajectory-prediction methods proposed over the last two decades for autonomous driving. It starts with the problem formulation and algorithm classification. Then, the popular methods based on physics, classic machine learning, deep learning, and reinforcement learning are elaborately introduced and analyzed. Finally, this paper evaluates the performance of each kind of method and outlines potential research directions to guide readers.**

***Index Terms*—Autonomous driving, trajectory prediction, machine learning, deep learning, reinforcement learning.**

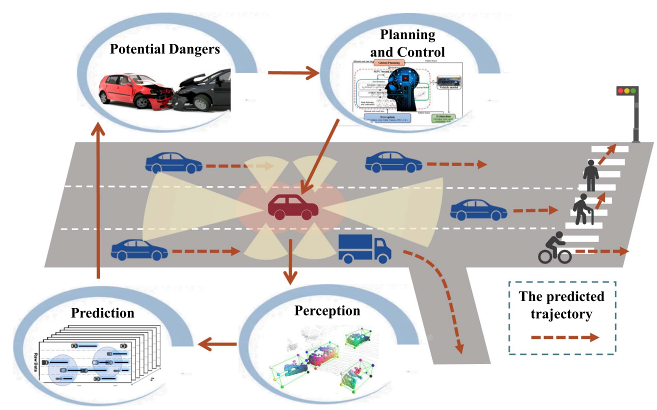

Fig. 1. The impact of trajectory prediction.

I. Introduction

Autonomous driving is attracting more and more attention from both academia and industrial sectors [1], because of its promising merits to solve many long-term transportation challenges related to safety, congestion, energy-saving, and so on [2], [3]. In recent years, we have witnessed the rapid development of perception, planning, and control systems for autonomous vehicles (AVs). However, mass production of AVs will become true only if the safety of autonomous driving is verified. To further improve the safety, one of the most key technologies is AVs should be able to predict the future states of the surrounding environment in real time like human drivers.

When a human drives a vehicle, he/she usually observes the surrounding traffic participants and predicts their future states before initiating a new driving maneuver, e.g., acceleration or lane change. Future states of traffic participants can be represented by future trajectories, utilized to detect potential dangers in advance and used in designing decision-making or planning algorithm, as shown in Fig. 1. However, due to diverse maneuvers of traffic participants, the complex interactions between traffic participants and environments, the uncertainty of sensory information, the computation burdens and computing time requirements of AVs, how to accurately predict future trajectories of traffic participants is drawing much attention and becoming one of the key points to improve the safety of autonomous driving.

Many researchers are devoted to the field of trajectory prediction and propose a number of useful methods. Several review papers have discussed a part of trajectory-prediction techniques. Lefèvre *et al.* [4] present a survey on existing methods of motion prediction and risk assessment for AVs before 2014. Most of these methods are classical but out of date. Mohammad *et al.* [5] review behavior-prediction methods at intersections based on drivers' maneuvers. A review of deep learning-based approaches focusing on vehicle behavior analysis is presented in 2019 by Mozaffari *et al.* [6], which describes different criteria to classify only a part of popular methods based on input and output information, and it does not involve some latest published methods. Two recent publications [7], [8] similarly focus on trajectory prediction for AVs, but Ref. [7] provides a review about tracking and trajectory prediction, which only contains methods using deep learning and methods using stochastic techniques, and Ref. [8] only presents deep learning methods. Other surveys [9], [10] use vision information to detect anomaly behavior and Ref. [11], [12] survey human motion prediction, which is obviously different from the topic of this study.

Thus, this survey comprehensively reviews trajectory-prediction methods for AVs proposed over the last two decades. We select heuristic and state-of-the-art trajectory prediction methods for a period of time to compare and summarize. Note

Manuscript received 24 January 2022; revised 3 April 2022; accepted 11 April 2022. Date of publication 13 April 2022; date of current version 24 October 2022. This work was supported in part by the National Key R&D Program of China under Grant 2020AAA0108100, and in part by the Science and Technology Commission of Shanghai Municipality under Grant 21ZR1465900. *(Corresponding author: Hong Chen.)*

The authors are with the School of Automotive Studies, Tongji University, Shanghai 201804, China (e-mail: huangyanjun404@gmail.com; 2031575@tongji.edu.cn; yangziru1029@163.com; zhou_ze_wei@163.com; zhanglin_jlu@foxmail.com; chenhong2019@tongji.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2022.3167103.

Digital Object Identifier 10.1109/TIV.2022.3167103

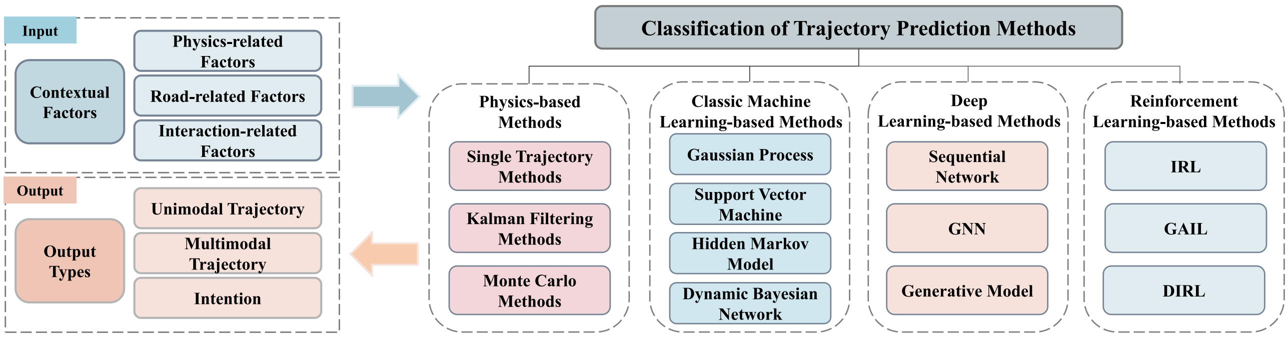

Fig. 2. The taxonomy of trajectory-prediction models for AVs.

that the historical trajectory information used in prediction methods can be obtained from the perception system [13] and vehicle to everything V2X [14], and vision-based methods are not the focus of this review. Since traffic participants, e.g., surrounding vehicles, directly impact the ego vehicle, this paper mainly focuses on trajectory-prediction methods for vehicles. As shown in Fig. 2, this paper will review physics-based methods, classic machine learning-based methods, deep learning-based methods, and reinforcement learning-based methods, respectively. The main contributions of this work can be summarized as follows:

1) The popular trajectory prediction methods for AVs based on physics, classic machine learning, deep learning, and reinforcement learning are elaborately reviewed.
2) The metrics and datasets for evaluating the performance of methods are detailed summarized.
3) The pros and cons of each method are discussed, and potential research directions are outlined.

The rest of this paper is arranged as follows: In Section II, the problem of trajectory prediction is described and methods used are classified according to different criteria. Section III, IV, V, and VI review physics-based methods, classic machine learning-based methods, deep learning-based methods, and reinforcement learning-based methods, respectively. Section VII summarizes the datasets and metrics for trajectory prediction and compares some methods based on the NGSIM dataset. Section VIII summarizes the pros and cons of each method and puts forward some possible future directions. The key conclusions are presented in Section IX.

II. Problem Formulation and Classification of Trajectory-Prediction Methods

In this section, the problem of trajectory prediction is described and the existing methods are classified based on different criteria.

A. Problem Formulation of Trajectory Prediction

Trajectory-prediction problems can be expressed as using past states of traffic participants in a given scene to estimate their future states. The historical states of traffic participants, e.g. vehicles, observed by the AVs or road side units is

$$\boldsymbol{X} = \left\{\boldsymbol{p}^1, \boldsymbol{p}^2, \ldots, \boldsymbol{p}^{t_h}\right\}, \tag{1}$$

where $\boldsymbol{p}^t (t \in 1, 2, \ldots, t_h)$ represents the states when the number of time steps is t; t_h represents the length of historical trajectory and $\boldsymbol{p}^{t_h}$ denotes the states of traffic vehicles at the current time. Regarding most of trajectory-prediction methods, $\boldsymbol{p}^t$ only contains the coordinate information of the vehicles, defined as:

$$\boldsymbol{p}^t = \left\{x_0^t, y_0^t, x_1^t, y_1^t, \cdots x_n^t, y_n^t\right\}, \tag{2}$$

where n represents all traffic vehicles detected by the ego vehicle; (x_j^t, y_j^t) refers to coordinates of vehicle j at time step t. $\boldsymbol{X}$ is the input of the prediction model, and vehicle trajectory with a time step length t_f is predicted. For other methods, $\boldsymbol{p}^t$ may also contain information such as velocity, acceleration, orientation, etc. The output of the model is defined as:

$$\boldsymbol{Y} = \left\{\boldsymbol{p}^{t_h+1}, \boldsymbol{p}^{t_h+2}, \ldots, \boldsymbol{p}^{t_h+t_f}\right\}. \tag{3}$$

Regard the trajectory prediction model as the function $\mathscr{F}$. Some methods can directly output future trajectories, that is $Y = \mathscr{F}(X)$. Others generate intermediate results M, from which Y is generated: $M = \mathscr{F}_1(X)$, $Y = \mathscr{F}_2(M)$. Note that, M can be maneuvers generated by some maneuver-based methods, or reward functions generated by reinforcement learning-based methods, etc.

B. Classification of Trajectory-Prediction Methods

The classification of trajectory-prediction methods for AVs, input, and outputs are shown in Fig. 2. Besides Fig. 3 shows the input and output factors of trajectory prediction.

1) Prediction Methods: According to different modeling approaches, prediction methods over the last two decades can be divided into four parts: physics-based methods, classic machine learning-based methods, deep learning-based methods, and reinforcement learning-based methods, as shown in 2.

2) Contextual Factors: Since trajectory-prediction methods usually need to model future states based on their historical trajectories under current environment, some factors should be considered. This study divides these factors into three categories: physics-related factors, road-related factors, and interaction-related factors.

1) *Physics-related factors* refer to dynamics and kinematics factors of vehicles.
2) *Road-related factors* include the modeling of the map information and corresponding traffic rules.

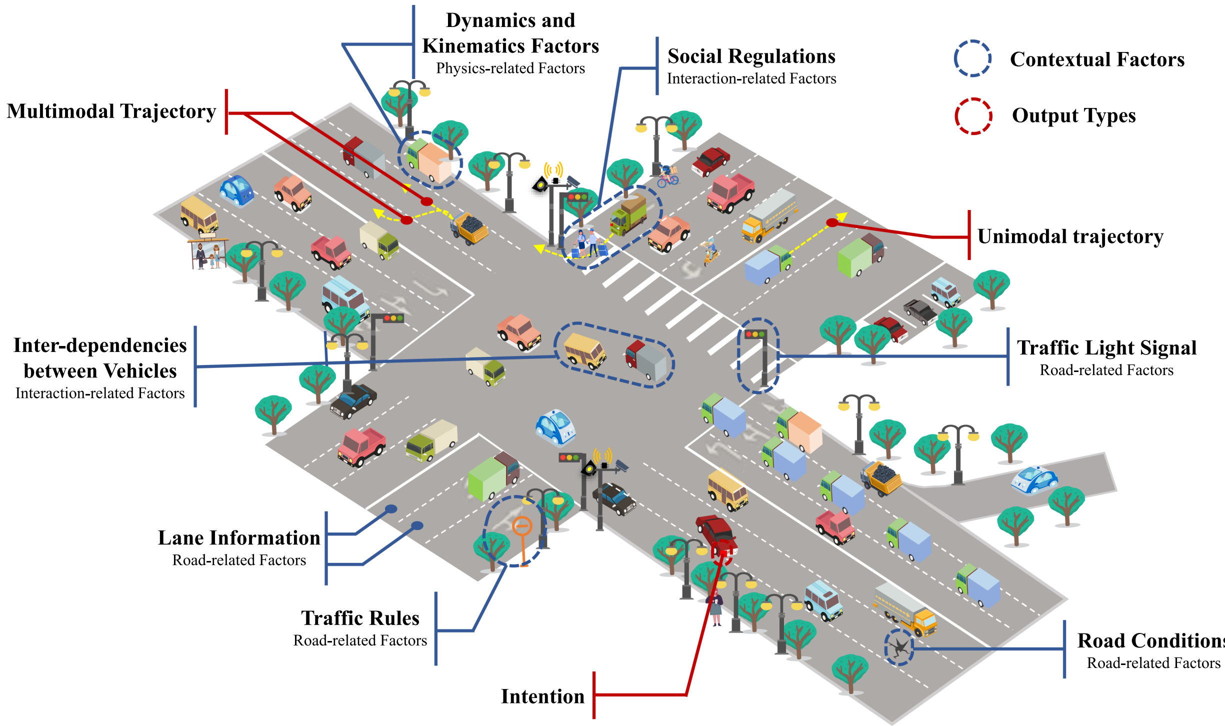

Fig. 3. The input and output factors of trajectory prediction.

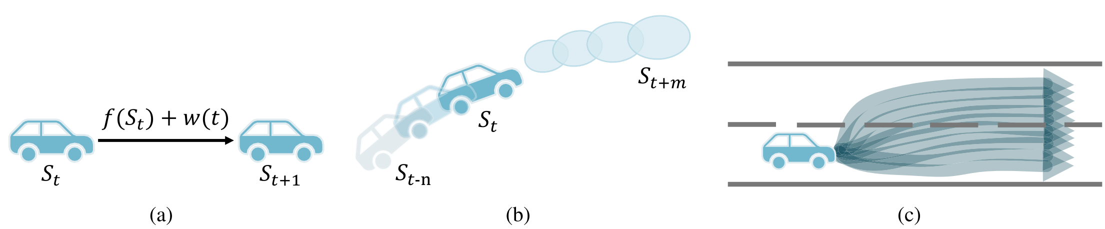

Fig. 4. Illustrations of physics-based methods: (a) the Single Trajectory methods, (b) the Kalman Filtering methods (based on [15]), (c) the Monte Carlo methods (based on [16]).

3) *Interaction-related factors* include the social regulations and inter-dependencies between vehicles' maneuvers.

3) Output Types: Trajectory-prediction methods need to provide the future trajectories of traffic participants, which can be unimodal or multimodal. In addition, some methods also provide the behavior intention of traffic participants. Therefore, the prediction algorithm can be divided into the following three categories according to the output type.

1) *Unimodal trajectory:* Prediction methods output a future trajectory for a single or multiple traffic participants.
2) *Multimodal trajectory:* Prediction methods generate multimodal future trajectories for traffic participants with the probability of each future trajectory.
3) *Intention:* Prediction methods produce behavior intentions to assist in prediction. Intention can be part of the final output, or just be an intermediate step in the method.

In the following sections, we will introduce different prediction methods, and analyze them according to the above classification approaches.

III. Physics-Based Methods

The physics-based methods employ the vehicle's dynamics or kinematics models. Typically, they include the Single Trajectory methods, the Kalman Filtering methods, and the Monte Carlo methods, as shown in Fig. 4.

A. *Physics Models*

Physics models contain *the dynamics models* and *the kinematics models*. Dynamics models can become very complex, including many inherent parameters, but complex dynamics models bring small gains in predictive accuracy and introduce an extra computation burden, such that a simple dynamics model is preferred for trajectory prediction. In the prediction task, the vehicle is usually regarded as a bicycle model, driven by the front wheels [17]–[19].

Thanks to a simple structure, kinematics models are used more often than dynamics ones. The commonly used include the Constant-Velocity (CV) and Constant-Acceleration (CA) models [15], [20], [21], Constant Turn Rate and Velocity (CTRV) and

Constant Turn Rate and Acceleration (CTRA) models [22], [23], the Constant Steering angle and Velocity (CSAV) and Constant Steering Angle and Acceleration (CSAA) models [24], etc.

B. Single Trajectory Methods

A simple method to predict vehicle trajectory is to directly apply the vehicle's current state to the physics model. This method applies to both the dynamics model [17]–[19], [25] and the kinematics model [22], [26], [27]. In [25] the linear bicycle model is used for collision avoidance, while Lytrivis *et al.* [22] and Miller *et al.* [26] use the CTRA model and the CV model, respectively. The advantage of this method lies in its high computational efficiency, and it is suitable for less constrained applications. However, they are not able to consider the road-related factors and the uncertainty of the current state is unreliable for long-term prediction.

C. Kalman Filtering Methods

Single Trajectory methods assume the states of vehicles are perfectly known without noises. In contrast, the Kalman Filtering (KF) methods are able to handle such noises, which model the uncertainty or noise of the current vehicle's state and its physics model by a Gaussian distribution [28]. The prediction and update steps are combined into a loop, the mean value and covariance matrix of the vehicle state can be obtained for each future time step, calculated as an average trajectory with related uncertainty [15], [24].

Compared to the previous method, the advantage is that it considers the uncertainty of the predicted trajectory. However, unimodal Gaussian distribution is not enough to represent different operations such that Kaempchen *et al.* [28] propose Interacting Multiple Model (IMM) to output Multimodal trajectories. Switched Kalman filter (SKF) [29] relies on a set of Kalman filters used to describe physical models of the vehicle and switch between them [28], [30]. Zhang *et al.* Ref. [31] proposes a method based on vehicle-to-vehicle communication and KF, enabling the host vehicle to predict trajectories of remote vehicles for obstacle avoidance. Recently, Lefkopoulos *et al.* [32] present a novel method called Interacting Multiple Model Kalman Filter (IMM-KF), which takes interaction-related factors into consideration. The proposed method uses the physics-based model to predict trajectories of traffic participants for multiple seconds.

D. Monte Carlo Methods

In general, an analytical expression for the predicted state distribution is usually unknown without any assumptions of the linearity or the model's Gaussian nature. Monte Carlo method can simulate the state distribution approximately. It randomly samples the input variables and applies the physics model to generate potential future trajectories. To ensure the feasibility of a maneuver, the generated trajectory samples can be filtered with a lateral acceleration lower than the actual allowable lateral acceleration [16], or a vehicle's physical limitations can be considered in the physics model such that the input of the model will be more realistic [33]. The Monte Carlo method can be used to predict the trajectories of traffic participants from a completely known state or from an uncertain state estimated by a filtering algorithm. Okamoto *et al.* [34] present a maneuver-based model that applies the Monte Carlo method to predict future trajectories by the identified maneuver. Wang *et al.* [35] use the Monte Carlo method to predict trajectories and utilize MPC to optimize the reference trajectories.

E. Summary

Physics-based methods utilize physics models to accomplish trajectory prediction with relatively low computational resources. Based on the classification approaches in Section II-B2 and II-B3, this paper classifies physics-based methods as shown in Table I. Physics-based methods are the first and simplest methods used by researchers. Although the accuracy of these methods is relatively low, more and more models use the idea of physics-based models to improve the accuracy. Physics-based methods have more accurate results when the movement of vehicles can be accurately described by kinematics or dynamics models, but the physical model of the traffic participants is constantly changing, such that most of these methods are only suitable for short-term prediction (no more than 1 s). The use of one or more physical models can obtain future trajectories of traffic participants quickly, but the choice of physical model and switch between them will bring an obvious prediction error. One way to solve this problem is to take interaction-related factors into consideration like IMM-KF [32]. To reach the state-of-art level, physics-based methods possibly need to combine with the learning-based methods, such as Ref. [36], which uses a learning-based discriminator to extract interaction information and generate model-based trajectories.

IV. Classic Machine Learning-Based Methods

Unlike physics-based methods that use several physics models, machine learning-based methods apply data-driven models to predict trajectories. According to the body of literature, classic machine learning-based methods for trajectory prediction of AVs include Gaussian Process (GP), Support Vector Machine (SVM), Hidden Markov Model (HMM), Dynamic Bayesian Network (DBN), K-Nearest Neighbors (KNN), Decision Tree, and so on. Since the most commonly used methods in classic machine learning are GP, SVM, HMM, DBN, this section will mainly introduce these methods.

A. Gaussian Process

The prototype trajectory method is one of the maneuver-based methods, which divides vehicles' trajectories into a collection of several types of prototype trajectories. The model measures the similarity between the historical trajectory and the prototype set to predict the possible trajectory. Gaussian Process (GP) [37] is an effective means used in the prototype trajectory method [38]–[40].

When GP is applied to predict trajectory, trajectories are regarded as the samples of GP, sampled along the time axis. The samples are represented by N discrete points to map to

TABLE I
SUMMARY OF PHYSICS-BASED METHODS

Physics-based Methods		Single Trajectory Methods	Kalman Filtering Methods	Monte Carlo Methods
Contextual Factors	Physics-based Factors	[17]–[19], [22], [25]–[27]	[15], [24], [28]–[32]	[16], [33]–[35]
	Road-related Factors	[22]		[35]
	Interaction-related Factors		[32]	[34]
Output Types	Unimodal Trajectory	[17]–[19], [22], [25]–[27]	[15], [24], [31]	
	Multimodal Trajectory		[28]–[30], [32]	[16], [33]–[35]
	Intention		[32]	[34]

the N-dimensional space. After that, the sample satisfies the N-dimensional Gaussian distribution in the N-dimensional space. Therefore, the main task of the GP model at the modeling stage is to determine the parameters of GP through the samples. In [41] HMM is used to estimate likely behaviors, then GP is employed to predict the trajectories. GP can also be used to model interaction-related factors, Trautman *et al.* [42] use GP for joint collision avoidance to solve the frozen robot problem. Guo *et al.* [43] apply GP and Dirichlet process (DP) to define motion processes and apply a non-parametric Bayesian network to extract potential motion patterns.

For methods based on prototype trajectory, each trajectory can be represented by the prototype set through training. Therefore, the main difference between these methods is how to construct the prototype trajectory. Govea *et al.* [44] obtain the prototype trajectories by statistically calculating the mean and the variance of all trajectory samples. Hermes *et al.* [45] divide the sample trajectories into several subsets and obtain several prototype trajectories after training to reflect vehicle movement changes. However, it is difficult to generalize these models to other scenes because the methods based on trajectory samples only trained for specific scenarios.

B. Support Vector Machine

Support Vector Machine (SVM) can learn and recognize driver's maneuver in a complex environment. The key of SVM is to find the support vector that meets the classification requirements and determine the optimal hyperplane which can maximize the interval of the classified data. When applied to the trajectory-prediction problem, driving maneuvers are usually defined into several categories: turning left, turning right, keeping straight, etc. Then it uses the kernel function to convert the input data to high-dimensional and perform linear classification in the space to find the driving maneuvers so as to predict the trajectory.

Mandalia *et al.* [46] first apply SVM to identifying lane-changing maneuver, using characteristics such as steering wheel angle, position, and acceleration for identification. Since SVM can output the characteristics of classification probability, Kumar *et al.* [47] propose a layered architecture method combining SVM and Bayesian filtering to identify lane-changing maneuvers so as to obtain more accurate identification results. In [48], [49], SVM is used to identify the maneuvers of traffic participants. Accordingly, SVM can identify vehicles' maneuvers, but SVM needs to define the driver's maneuver in advance, and preset maneuvers will also impact the final prediction results.

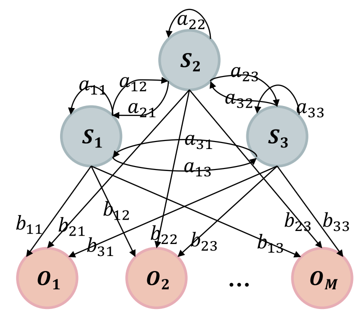

Fig. 5. Illustration of Hidden Markov Model (based on [51]).

C. Hidden Markov Model

SVM is effective in classification problems, but it is not as effective as Hidden Markov Model (HMM) in trajectory prediction. HMM is one of the most popular classic machine learning-based trajectory prediction methods. HMM is also a maneuver-based method that uses Markov Chain. The Markov Chain refers to a process containing finite events, the state at time $t+1$ of the system is only related to the previous time t, and the state transition probability is not related to time. The mathematical expression is:

$$\begin{aligned} &\mathrm{P}\left(S_{n+1}=s \mid S_1=s_1, S_2=s_2 \cdots, S_n=s_n\right) \\ &\quad = \mathrm{P}\left(S_{n+1}=s \mid S_n=s_n\right). \end{aligned} \tag{4}$$

In real life, we can only observe the distinct state that is exposed on the surface, but no intuitive representation of its hidden states exists. Therefore, it is necessary to establish a Markov process with hidden states and get the essential states of events through the observable states set related to the hidden states' probability, which is the so-called Hidden Markov Model. HMM is represented by (S, O, A, B, π) [50], as shown in Fig. 5:

- $S = \{S_1, S_2, \cdots S_N\}$ represents the hidden states sequence.
- $O = \{O_1, O_2, \cdots O_M\}$ represents the observation sequence.
- A represents the transition probability matrix between hidden states.
- B is the output matrix, representing the transition probability of hidden states to output states.
- π is the initial probability matrix, representing the initial probability distribution in hidden states.

When HMM is used in the trajectory prediction, the historical states of traffic participants are represented by observation sequence O, and HMM solves the most likely future observation sequence. Holger *et al.* [52] use the steering angle and global coordinates as the input of HMM to predict the driver's maneuvers. Based on HMM, Qiao *et al.* [53] propose an algorithm called HMTP* that selects parameters adaptively to simulate the real scenes at a dynamically changing speed. In [51], HMM combined with Fuzzy Logic is used for driver maneuver prediction. Besides, HMM can be integrated into decision-making and planning systems. In [54], HMM is used for trajectory prediction and risk assessment, and the results of are fed into the decision-making and planning system.

Although traditional HMM methods have achieved a great success in predicting driver's maneuvers, they do not consider the impact of interaction-related factors in the prediction process, such that its prediction results are not accurate enough in actual traffic scenes. Deo *et al.* [55] propose a vehicle trajectory prediction model based on HMM and Variational Gaussian Mixture Models (GMM) considering interaction-related factors. The vehicle interaction information is obtained by finding the optimal solution of the energy function. Zhang *et al.* [56] propose a GMM-HMM maneuver prediction model based on game theory, considering interaction-aware factors.

D. Dynamic Bayesian Network

To improve the accuracy of trajectory prediction, the prediction model should consider at least both vehicle states and the interaction effect between traffic participants. Dynamic Bayesian Network (DBN) mentioned by Koller *et al.* [57] can model such interactions. DBN is a maneuver-based method that uses the Bayesian Network and considers time sequence. The basic concepts and probabilistic inference of DBN are the same as Bayesian Networks. The difference is Bayesian Networks describe static systems, while Kevin *et al.* [58] introduce the concept of time templates to solve timing issues in probabilistic models. Time segment refers to a time template materialized according to DBN, which discretizes continuous time into countable points with preset time granularity.

Generally, the preset time granularity should be consistent with the actual state acquisition frequency, and DBN is trained according to the sensor sampling frequency as the time segment. Besides, the inference and learning methods of DBN need to be converted into Bayesian Networks before they can be directly applied. Common inference methods of Bayesian Networks include the Variable Elimination Method, Clique Tree Algorithm, and Sampling Algorithm. The learning methods of Bayesian Networks include Maximum Likelihood Estimation, Bayesian Estimation, EM algorithm, etc. Also, special inference methods for DBN with high complexity exist, such as the forward and backward inference method [65].

The architecture of DBN includes a behavior layer, a hidden layer, and an observation layer, as shown in Fig. 6. The behavior layer represents the network's input information, and the observation layer represents the driver's maneuver. Using this architecture, Gindele *et al.* [59] model the driving maneuvers of multiple vehicles. The input information includes all vehicle states, vehicle interaction relationships, road structures, observation states, etc. Schreier *et al.* [60] apply DBN

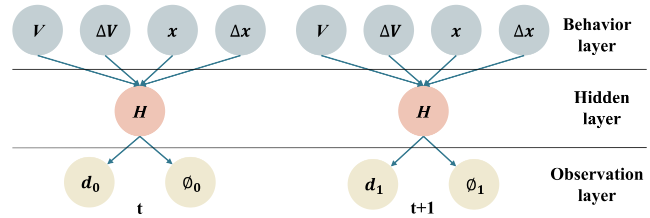

Fig. 6. Illustration of Dynamic Bayesian Network (based on [62]).

to judge driving maneuvers and utilize the kinematics model corresponding to each driving maneuver to predict the trajectory. In [61], the vehicle maneuver is predicted by game theory, and then the vehicle motion is judged by DBN which considers the interaction-related factors. He *et al.* [62] use DBN to identify vehicle following and lane-change maneuvers, and predict the trajectory of lane-change maneuver. In [63], DBN is designed to consider physics-related factors, road-related factors, and interaction-related factors. Li *et al.* [64] combine DBN with end-to-end models to predict pedestrian trajectories, where DBN is used to extract traffic participants' characteristics and dynamics information, end-to-end models treat the prediction problem as a sequential generation problem to generate the prediction trajectory.

DBN models the effect of interaction between traffic participants when applied to trajectory prediction and perform well in classic machine learning-based methods. As maneuver-based methods, DBN models obtain high recognition performance and have been used in several real-world tests [66]. However, DBN still faces the error problem from recognizing maneuvers to generating trajectories. Many methods can only judge two or three maneuvers, such as lane-keeping and lane-changing, and the model's generalization ability is not strong.

E. Summary

In summary, the classic machine learning-based methods determine the probability distribution by mining data features, which can be classified as shown in Table II. The classic machine learning-based methods provide new ideas for trajectory prediction, which promote the development of learning-based methods. With more factors to be considered, the accuracy of these methods keeps increasing, contributing to trajectory prediction. Most of these methods are maneuver-based methods, which can predict future trajectories by first judging maneuvers. In these methods, the maneuvers usually need to be provided or identified in advance.

V. Deep Learning-Based Methods

Most traditional prediction methods are only suitable for simple prediction scenes and short-time prediction tasks. Recently, trajectory prediction methods based on deep learning have become increasingly popular because they can not only consider the physics-related factors and road-related factors but also consider the interaction-related factors and adapt to more complex scenes. A general description of these methods is shown

TABLE II
SUMMARY OF CLASSIC MACHINE LEARNING-BASED METHODS

Classic Machine Learning-based Methods		GP	SVM	HMM	DBN
Contextual Factors	Physics-based Factors	[38]–[45]	[46]–[49]	[41], [50]–[56]	[59]–[64]
	Road-related Factors	[38]–[41], [43]	[46]–[49]	[41], [52], [54]–[56]	[59], [60], [62], [63]
	Interaction-related Factors	[42], [43]		[55], [56]	[59], [61]–[64]
Output Types	Unimodal Trajectory		[46]–[49]	[50], [53]	[61], [62]
	Multimodal Trajectory	[38]–[45]		[41], [54]–[56]	[59], [60], [63], [64]
	Intention	[38]–[41], [43]	[46]–[49]	[41], [50]–[53], [55], [56]	[60]–[64]

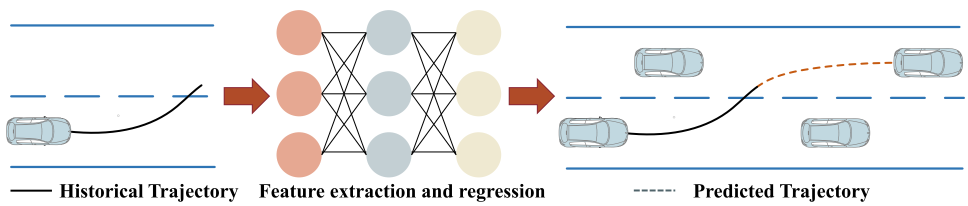

Fig. 7. Description of deep learning-based methods.

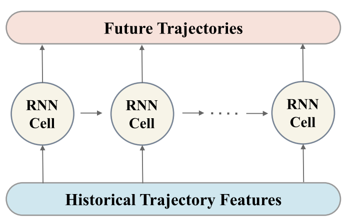

Fig. 8. Illustration of Recurrent Neural Network.

in Fig. 7. In the following, this paper summarizes the current popular deep learning-based trajectory prediction methods for AVs.

A. Sequential Network

The sequential network is used to extract features of historical trajectory and can be used as the output layer. The sequential network for trajectory prediction based on deep learning mainly includes Recurrent Neural Network (RNN), Convolutional Neural Network(CNN), and Attention Mechanism(AM).

1) Recurrent Neural Network: Different from the classic machine learning-based methods and CNN that can effectively process spatial information, RNN is designed to handle temporal information [67], [68]. It stores information of the previous time steps and utilizes hidden states together with the input to determine the output, as shown in Fig. 8. However, in practical application, it is found that when the number of time steps is large, the gradient of the RNN is more likely to attenuate or explode. Gated RNN, e.g., Long Short-Term Memory Network (LSTM) and Gated Recurrent Unit (GRU) can solve this problem. Trajectory prediction models using RNN can be divided into the single RNN models and the multiple RNN models.

A single RNN is utilized for maneuver-based and single-modal trajectory prediction, or applied to other auxiliary models to support more complex functions, such as interaction-aware prediction. LSTM is used in [69]–[71] as a sequence classifier to predict vehicles' maneuvers. To achieve the goal, cells of LSTM extract vehicle features, and the hidden states of the last cell will be fed to the output layer to predict maneuvers. In [69], [70], the input is fed to the fully connected layer to extract features, and then substituted into the three-layer LSTM; In [71], two layers of LSTM without embedding are used. Altché *et al.* [72] use a single-layer LSTM to accomplish trajectory prediction of the target vehicle. To predict the maneuver-based trajectory, Ding *et al.* [73] use an LSTM encoder that encodes the states of the target vehicle to predict its maneuver, and trajectory prediction is achieved by using the predicted maneuver and map information. Finally, based on interaction-related factors, traffic rules (such as red lights), and map information, this paper uses nonlinear optimization methods to optimize the initial future trajectory. In order to predict multi-modal trajectories, Zyner *et al.* [74] adopt the weighted Gaussian Mixture Model (GMM) for prediction, and its parameters are obtained by an encoder-decoder three-layer LSTM, and then predicted trajectories are clustered using the modal with the highest probability. Hyeon *et al.* [75] utilizes an encoder-decoder LSTM architecture. The LSTM encoder encodes the historical trajectory features, and the LSTM decoder solves the K most likely future trajectories through the beam search algorithm. Xing *et al.* [76] predicts trajectories for the first vehicle in the fleet, which uses GMM to distinguish driving styles and uses the LSTM and fully connected regression layer to analyze sequence data and driving styles to predict vehicle trajectory. By calculating the distance between the vehicle and the centerline, Chang *et al.* [77] propose an LSTM encoder-decoder baseline that takes into account map information and social information, and compares with the Nearest Neighbor (NN) regression method. Considering lane information, Kawasaki *et al.* [78] combine LSTM with KF for multi-modal trajectory prediction.

With the development of neural networks, several groups of RNN architectures are widely used. Two groups of LSTM networks are used by Dai *et al.* [79] to predict the target vehicle's trajectory. One group is used to model the trajectory of surrounding vehicles, and the other group is used to model the interaction between surrounding vehicles. Ding *et al.* [80] present a set of GRU encoders to describe the paired interactions between vehicles. Xin *et al.* [81] use an LSTM to predict the target lane of the target vehicle, and use another LSTM to predict the trajectory according to the target vehicle's state and the predicted target lane. To predict the multi-modal trajectory, Deo *et al.* [82] proposes six different LSTM decoders, which are related to six specific maneuvers. The encoder LSTM encodes the features of the historical trajectory. The one-hot vector that represents the specific maneuver of the vehicle connects the encoder and the decoder. The decoder LSTM predicts binary Gaussian distribution parameters to output the future trajectory and predict the probability of each of the six maneuvers. In [83], five RNNs and three fully connected layers are used to process the input data and output the three coefficients of the cubic polynomial, representing the future trajectory of the target vehicle. Tang *et al.* [84] use rigorous mathematical modeling from a probabilistic perspective to construct an MFP model with an end-to-end structure. The model contains a group of RNNs that share parameters in parallel, forming a dynamic state encoder based on the attention mechanism. Each encoder RNN represents a vehicle's trajectory by aggregating the history information and automatically learns multi-modal information through discrete latent codes. Then the decoder RNNs calculate the probability of the multi-modal trajectories to obtain the prediction trajectory. In [85], multi-modal trajectories are generated based on a multi-head attention layer, which uses LSTM as encoder-decoder and puts two attention layers in the middle. In [86], a recurrent attention and interaction model is presented to predict trajectories of pedestrians. Zhang *et al.* [87] propose a multiple LSTM-based framework that combines intention prediction and trajectory prediction. The intention of the vehicle at intersections is predicted by one LSTM model and the trajectory is predicted by another LSTM-based prior trajetories model.

2) Convolutional Neural Network: Recently, CNN has achieved success in many tasks, such as computer vision [88], [89] and machine translation [90]. Besides, Nikhil *et al.* [91] believe that using CNN to predict the trajectory is better than RNN because the trajectory has a strong spatio-temporal continuity. They apply a sequence-to-sequence structure, take the historical trajectory as input, and implement time continuity by stacking the convolutional layer after a fully connected layer, and output the future trajectory through a fully connected layer. Experiments show that using this CNN-based network runs faster. The general methods using CNN to process trajectory information is shown in Fig. 9.

However, most methods applying only the CNN framework use bird-eye image as the input. In [92], a set of possible future trajectories is generated by the vehicle state (velocity, acceleration, and yaw rate) and the raster image, the trajectory with the highest probability is found as the future trajectory through analyzing semantic features. Cui *et al.* [93] make a progress in embedding a bicycle vehicle kinematics model with CNN [94] for trajectory prediction, which also operates on raster maps. In [95], using a novel fast CNN architecture trajectory prediction of vulnerable road users (VRUs) is presented by context rasterization techniques [96].

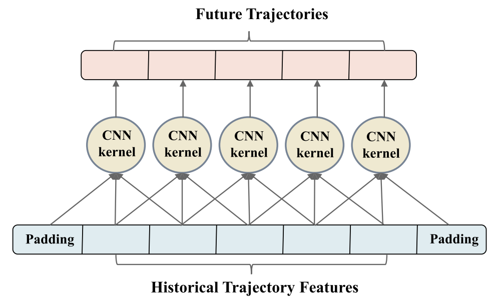

Fig. 9. Illustration of Convolutional Neural Network for trajectory prediction.

In addition, some other neural networks apply to trajectory prediction using the CNN framework. In [97], CNN is applied to the rasterized image meanwhile TCN is used to capture history trajectories features which will be concatenated with the raster feature and the current state. Zhang *et al.* [98] employ TCN to predict the lane-change maneuver and trajectories. In [99], a persistent Memory Augmented Neural Network (MANN) is used for trajectory prediction. CNN is applied to understand the scene image, scene features and trajectory features will be processed in MANN to generate multimodal trajectories. In recent years, new methods use CNN for trajectory prediction and achieve state-of-the-art results. The first one [100] uses CNN to output a heatmap to represent the agent's possible future. The second one [101] introduces the point cloud learning method into trajectory prediction to capture both spatial and temporal information.

3) Convolutional and Recurrent Neural Network: RNN is able to extract temporal features, which is very suitable to process time-series information; whereas, CNN is capable of extracting spatial features including the interaction-related factors between traffic participants. This has inspired some researchers to use a combination of RNN and CNN to process the temporal and spatial information for trajectory prediction. Deo *et al.* [102] use an LSTM encoder to extract the temporal information of surrounding vehicles, and then feed it into a social pooling layer [103] to form a social tensor. In this study, the social pooling layer captures interaction-related factors between vehicles after spatial rasterization, and then the social tensor is sent to a set of CNNs to learn the spatial correlation of vehicles. Finally, six LSTM decoders are used to generate distributions of six specific maneuvers, which include three lateral maneuvers (left lane change, right lane change, and keep lane) and two longitudinal maneuvers (brake, normal speed). Then it finds the maneuver with the highest probability and predict its future trajectory. Chandra *et al.* [104] propose a model called TraPHic based on the CNN-LSTM hybrid network to predict traffic participants' trajectories. The model feeds the state and the surrounding objects of the main vehicle into the CNN-LSTM networks to obtain their features, then connects these features and the LSTM decoder to obtain the predicted trajectory of the main vehicle, but this algorithm only predicts the trajectory of one object per operation. Xie *et al.* [105] also use the CNN-LSTM framework. They use "box" to detect and eliminate outliers in the vehicle trajectory to obtain valid trajectory data, which will be fed into the convolutional layer and the maximum pooling layer to extract interaction-aware features which will be fed into an LSTM and a fully connected layer for prediction. The hyperparameters of the model are optimized by the Grid Search (GS) algorithm.

To better predict the trajectory, researchers introduce High Definition (HD) maps information to make the predicted trajectory closer to the real trajectory [106]. HD maps generally include raster maps and vector maps, which contain semantic information about the road and can indicate line segments. Some methods employ CNN to extract the scene context information from the raster maps to take into account the road-related factors and the interaction-related factors. Because methods of using CNN to process raster maps belong to the category of perception system, which is not the focus. Therefore, this paper will briefly explain the process of trajectory prediction after getting semantic features from raster maps. Classic algorithms contain DESIRE [107], using stochastic 1-step policies. Hong *et al.* [108] encode semantic features with ConvNets to predict vehicles behaviors. Based on ConvNets, Chai *et al.* [109] find trajectory anchors through unsupervised learning, use GMM and semantic features to train their model. Except for the raster maps, the processing methods of vector maps are shown in section V-B.

4) Attention Mechanism: The attention mechanism allows the human to use limited attention resources to quickly filter out high-value information from a large amount of information. The attention mechanism (AM) in deep learning mimics the way humans think and is widely used in various types of deep learning tasks such as Natural Language Processing (NLP), image classification, and speech recognition [110]–[112]. AM is usually used in the trajectory prediction task [113]–[115]. In [116], the multi-head attention is used to extract the lane and vehicle attention to output the distribution of the future trajectories. In [117], AM models the interactions between traffic participants by extracting attentions from LSTM encoders, and in [118] each attention head models a possible way of interaction between the target and the combined context features. Vaswani *et al.* [119] propose the Transformer model, which uses substantial attention mechanisms to complete the sequence machine translation tasks without using RNN. The constraint of sequential computation remains for RNN, while the attention mechanism can perform a parallel calculation on sequential data. Since the Transformer model has achieved excellent results in machine translation, researchers apply the Transformer model for the trajectory prediction task, as shown in Fig. 10. In [120], the sequential step-by-step model of LSTMs and only-attention-based models, including the Transformer (TF) and the larger Bidirectional Transformer (BERT), are compared for predicting the future trajectories of pedestrians. It shows that the TF-based model has better performance especially in the long-term prediction, and can also cope with the missing input observations, which is the common phenomenon of real sensor data. In addition to modeling the trajectory sequences, TF can also model the interaction between traffic participants and environment [121]–[123]. Liu *et al.* [124] apply stacked transformers as the backbone, which integrate environmental information into trajectory proposal to predict future trajectory.

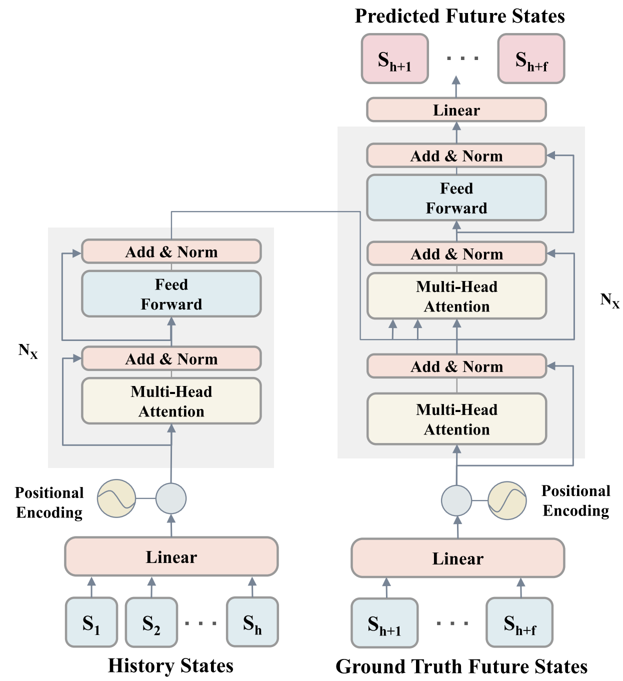

Fig. 10. Illustration of Transformer (based on [120]) for trajectory prediction.

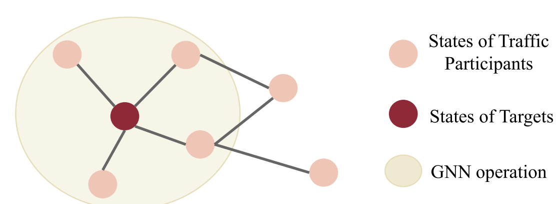

Fig. 11. Description of Graph Neural Network.

It can be seen that the transformer-based model has advantages in processing time-series data.

B. Graph Neural Network

When it comes to prediction methods considering interaction-related factors, each object in the environment can be regarded as a node to form a graph. Although some methods using RNN and CNN have achieved great success when extracting Euclidean spatial data features, the data in many practical application scenes are generated from non-Euclidean spaces. Because many classic deep learning-based methods are processing non-Euclidean spatial data, the performance of these methods is still unsatisfactory. Usually, each scene can be viewed as an irregular graph and each graph has an unordered node with a variable size, as shown in Fig. 11. The number of adjacent nodes of each node in the graph varies, resulting in some important operations, such as convolution, which are easy to calculate on the image but no longer suitable for direct use on the graph. Still, each node in the graph will have edges related to other nodes. This information can be used to capture the interdependence between objects. Therefore, Graph Neural Network (GNN) is very suitable for vehicle trajectory prediction problems based on interaction-related factors [125]. Diehl *et al.* [126] confirm this idea. They use two popular graph networks: Graph Convolutional Network (GCN) and Graph Attention Network (GAT) for trajectory prediction based on interaction-related factors and prove its effectiveness.

As for road-related factors, using CNN to process raster maps has a big computation burden and it is easy to lose information. In contrast, vector maps use polylines with multiple control points and their attributes to represent structured road information. These polylines form groups of vectors that can be used as nodes in GNN, which has been widely used in trajectory prediction. In the following, this paper will introduce the vehicle trajectory prediction methods based on GNN.

1) Graph Convolutional Network: Graph Convolutional Network (GCN) is the most popular graph neural network method. The graph convolutional network extends the convolution operation from traditional image data processing to graph data processing. The core idea is to learn a mapping function, which can extract interaction-aware features from the features of nodes in the graph and the features of their neighbors.

In the space-based graph convolutional network, a GCN-based trajectory prediction model called GRIP is proposed by Li *et al.* [127], which treats each vehicle as a node at each sampling time and considers the interaction-related factors. If two nodes represent the same vehicle and the sampling time is adjacent, an edge exists between the two nodes, representing the time relationship. If two nodes at the same time represent two vehicles, and the distance between the two vehicles is less than a fixed value, an edge exists between the two nodes, representing the spatial relationship and the interaction state of these objects. GRIP uses a GCN model composed of several convolutional layers and graphics operations to model the graph network. The output of GCN is fed to the LSTM encoder-decoder to predict the trajectory of surrounding vehicles. Although GRIP has a considerable improvement over the popular models at that time, GRIP uses a fixed graph network to represent the interaction-related factors between traffic participants, the generalization ability in complex scenes needs to be improved. Therefore, Li *et al.* [128] propose an upgraded version for GRIP, called GRIP++, which uses both fixed and dynamic graph networks to predict the trajectory of traffic participants. This method has higher accuracy than GRIP, and at the end of 2019 it ranked first in the Baidu Apolloscape dataset [129] ranking. Besides, GRIP uses LSTM encoder-decoder, while GRIP++ uses GRU encoder-decoder. Jeon *et al.* [130] propose a SCALE-Net model, which can predict any number of surrounding vehicles' trajectories while keeping the performance. SCALE-Net uses a edge-enhance graph convolutional network (EGCN) [131] to learn edge features in the traffic flow. For each moment, each vehicle is a node, and the node state is $X_l = [x, y, v_x, v_y, \theta]$ (represent x coordinate, y coordinate, speed in x-direction, speed in y-direction and heading angle respectively), and the edges between nodes are represented by $\Delta X = |X_m - X_l|$, showing a multi-dimensional state. The built graph is calculated by the multi-layer EGCN algorithm. At the next moment, the graph model will be rebuilt and the EGCN will run again. The output of EGCN is processed by a sequence model, which consists of an LSTM encoder-decoder, followed by a five-layer multi-layer perceptron (MLP). GCN is also used in pedestrian trajectory prediction tasks. For example, Mohamed *et al.* [132] model pedestrian trajectories as spatio-temporal graphs to replace clustering layers. The edges of the graph represent the interaction-related factors between pedestrians. To solve the problem for recursive units, the model uses GCN and temporal convolutional network (TCN) to operate on the spatio-temporal graph such that the model can predict the entire sequence at one time.

All of the above methods use space-based graph convolutional networks, but some papers use spectrum-based graph convolutional networks. Chandra *et al.* [133] use a two-layer GNN-LSTM structure to solve the trajectory prediction problem. The first layer uses an LSTM encoder-decoder to predict future trajectories of traffic participants, and the second layer models the interaction-related factors of traffic participants through a weighted dynamic geometric graph network (DGG) [134]. The spectrum in the graph is extracted by specific regularization of the eigenvalues after the LSTM encoder-decoder, and the spectrum sequence is fed into the LSTM network at the first layer to complete the prediction task. Zhao *et al.* [135] propose a spectrum-based GCN network that can share information among all vehicles in the scene to consider the change of the surrounding vehicles to adapt to the environment.

2) Graph Neural Network Using Vector Maps: Benz [136] first applies HD maps to trajectory prediction, and executes map topology based on the lane information associated with the vehicle to obtain its future trajectory along the lane. However, it does not consider interaction-related factors. Since the Argoverse dataset [77] with vector maps is proposed, researchers have used GNN to obtain the interaction features between vehicles, between vehicles and maps to improve the accuracy of trajectory prediction. Taking vehicles and vector maps in the scene as nodes, Gao *et al.* [137] propose VectorNet which uses GNN to achieve trajectory prediction. Liang *et al.* [138] use CNN to extract vehicle features and GCN to extract lane features from vector maps, and then combine these two features for trajectory prediction. Using VectorNet to extract map features, Zhao *et al.* [139] propose a target-driven method called TNT, which defines sparse goal anchors and selects the best trajectory to the target, and DenseTNT [140] estimates dense goal candidates and get better results than TNT. Zeng *et al.* [141] use LaneRCNN to obtain the representation of each participant's local lane map to encode their past trajectory and local map topology, and complete the interaction of the local lane map through the interaction module.

3) Other Graph Neural Network: The attention mechanism has now been widely used in sequence-based tasks. Its advantage is that it can amplify the impact of the most important part of the data. Veličković *et al.* [142] propose Graph Attention Network (GAT). When aggregating feature information, GAT uses an attention mechanism to determine the weights between nodes. Huang *et al.* [143] apply GAT to the trajectory prediction. The model firstly uses an LSTM encoder to encode trajectories of traffic participants, then uses GAT to calculate the weight of attention for each traffic participant and forms the interaction information of each participant at this moment by weighted

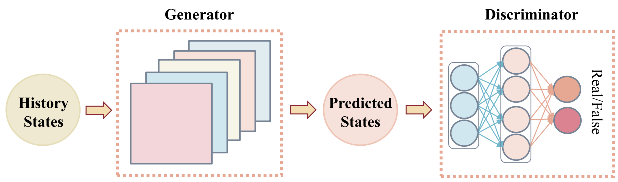

Fig. 12. Description of Generative Adversarial Network.

averaging these states. Finally, the model uses an LSTM decoder to generate predicted trajectories.

Besides, Zhang *et al.* [144] propose a social graph network applied to trajectory prediction. To effectively capture the social behaviors of traffic participants, the team uses a directed graph, which is dynamically constructed based on real-time position and speed direction. Based on the social graph, the LSTM network is constructed to collect social effects and trained by samples to generate end-oriented and interaction-aware representations. For the uncertainty of the future trajectory, the network uses a time stochastic method to sequentially learn the uncertainty in social interaction to form a priori model, then sample the prior model and use layered LSTM to decode step by step to generate the prediction trajectory. Recently, a graph-structured recurrent model named Trajectron++ is proposed in [145] to produce dynamically-feasible future trajectories, which represents a scene as a directed spatio-temporal graph and is designed to be tightly integrated with the planning system for AVs.

C. Generative Model

In the task of trajectory prediction, the multi-modality of the trajectory brings uncertainty and challenges to the research. To explain the inherent multi-modal distribution, some researchers use generative models to generate multi-modal trajectories. Generative models for trajectory prediction include Generative Adversarial Network (GAN) and Conditional Variational Auto Encoder (CVAE).

1) Generative Adversarial Network: Generative Adversarial Network (GAN) was first presented by Ian Goodfellow [146] in 2014. With superior performance, it has quickly become a major research hotspot in less than two years. GAN is essentially a generative model, which is mainly composed of two parts, namely Generator and Discriminator. The generator is utilized to generate a random sample similar to the real sample, and the discriminator is used for determining whether the data is true or false. Through the continuous game evolution of the generator and discriminator, GAN can obtain a generator with higher quality and a Discriminator with stronger judgment ability.

When applying GAN to trajectory prediction, the generator is utilized to generate the predicted trajectory, and the discriminator is utilized to judge whether the predicted trajectory is correct, as shown in Fig. 12. A typical application is that Gupta *et al.* [147] use GAN for pedestrian trajectory prediction called SGAN. The generator uses an LSTM encoder, pooling module, and LSTM decoder to generate the predicted trajectory, and the discriminator uses LSTM to determine whether the predicted trajectory is reasonable. In the model, the pooling module is social pooling, whose purpose is to help consider all pedestrians and reduce computation. Unlike the social pooling proposed in [102], the pooling module used here deals with the interaction between pedestrians. Based on SGAN, Yang *et al.* [148] design a pedestrian trajectory prediction model focusing on how to more effectively extract interaction-related factors and generate a variety of feasible trajectories, which adds a latent variable predictor on the basis of SGAN to estimate latent variables. Li *et al.* [149] use Environmental Attention Mechanism (EAM) for deep feature extraction, and then feed it into the GRU generator for trajectory prediction. Sadeghian *et al.* [150] propose a GAN-based model for predicting pedestrians and vehicles' trajectories, considering all vehicles' impact and the interaction between them for trajectory prediction. A feature extractor is applied, which uses CNN to extract features from the scene and an attention mechanism to consider the interaction-related factors. Hegde *et al.* [151] use the vehicle's coordinate information for the GAN network to predict the vehicle trajectory. Zhao *et al.* [152] propose a multi-agent tensor fusion network MATF-GAN which can preserve spatial structure information. The architecture combines the strengths of agent-oriented and spatial structure-oriented trajectory prediction methods and learns to represent relevant information about social interactions and physical constraints of the scene through end-to-end training. Wang *et al.* [153] propose a TS-GAN model, which uses a self-created convolutional social mechanism and a recurrent social mechanism to extract vehicle spatial and temporal information in the GAN network. Song *et al.* [36] use the vehicle state and the vector map information to generate model-based multi-modal trajectories, and use the learning-based discriminator to extract vehicle interaction information and obtain the optimal trajectories.

2) Conditional Variational Auto Encoder: The so-called Auto Encoder (AE) compresses data into a low-dimensional vector representation through the encoder and uses a decoder to decode the low-dimensional vector to obtain a reconstructed output. AE hopes to minimize reconstruction errors. However, AE is accused of simply "remembering" data, and its ability to generate data is poor. Kingma *et al.* [154] propose a Variational Auto Encoder (VAE) framework to use neural networks to parameterize the distribution in variational inference, thereby improving the generation ability of the model. In [155], a Conditional VAE (CVAE) is proposed to complete the structured prediction tasks. For trajectory prediction, combining CVAE and RNN variants into the form of encoder-decoder is an effective way for trajectory generation [75], [82], [84]. Some methods that use raw sensor data as input also use CVAE for multi-modal trajectory prediction [108], [156], [157]. These methods of using CVAE as the network framework have been mentioned above, and will not be repeated in this section.

D. Summary

In summary, deep learning-based trajectory prediction methods for AVs can be classified into Table III. More and more researchers apply deep learning-based methods to spatial and temporal prediction problems like trajectory prediction and

TABLE III
SUMMARY OF DEEP LEARNING-BASED METHODS

Deep Learning-based Methods		Sequential Network	GNN	Generative Model
Contextual Factors	Physics-based Factors	[69]–[87], [91]–[102], [104]–[109], [116]–[118], [120]–[124], [145]	[126]–[128], [130], [132], [133], [135], [137]–[141], [143], [144]	[36], [75], [82], [84], [108], [147]–[153], [156], [157]
	Road-related Factors	[69]–[75], [77], [78], [80]–[84], [86], [87], [92]–[102], [106]–[109], [116]–[118], [121]–[124], [145]	[132], [137]–[141], [143]	[36], [75], [82], [84], [108], [148]–[150], [152], [153], [156], [157]
	Interaction-related Factors	[73], [76], [77], [79]–[82], [84]–[87], [92], [100]–[102], [104]–[109], [116]–[118], [121]–[124], [145]	[126]–[128], [130], [132], [133], [135], [137]–[141], [143], [144]	[36], [82], [84], [107], [108], [147]–[153], [157]
Output Types	Unimodal Trajectory	[72], [73], [79], [81], [83], [91], [95], [96], [98], [104], [105], [120]	[126]–[128], [130], [133], [135], [137], [144]	
	Multimodal Trajectory	[74]–[78], [82], [84], [85], [85]–[87], [92]–[94], [97], [99]–[102], [106]–[109], [116]–[118], [120]–[124], [145]	[132], [138]–[141], [143]	[36], [75], [82], [84], [108], [147]–[153], [156], [157]
	Intention	[69]–[71], [73], [74], [76], [80]–[82], [84], [86], [87], [92], [98], [102], [108], [109]	[133], [135], [144]	[82], [84], [108], [147], [151], [153]

TABLE IV
THE MAINSTREAM APPROACHES FOR DEEP LEARNING-BASED METHODS

Classification	Methods	Year	State Encoder	Context Encoder	Interaction Module	Decoder	Description
RNN	MFP [84]	2019	RNN	CNN	Radial Basis Function	RNN	Learn latent variables to model the multimodel trajectories
CNN	CoverNet [92]	2020	CNN	CNN	-	Trajectory Set Generator	Apply the raster image as input
CNN	HOME [100]	2021	1D-CNN, GRU	CNN	Self-Attention	CNN	Output 2D topview heatmap
CNN	TPCN [101]	2021	PointNet++ [158]		Joint Learning	Displacement Prediction	Use point cloud learning-based methods
CNN and RNN	DESIRE [107]	2017	GRU	CNN	Social Pooling [103]	GRU	Use deep IOC framework to encode
CNN and RNN	CS-LSTM [102]	2018	LSTM	-	Social Pooling [103]	LSTM	Six LSTM decoders to generate distributions of six specific maneuvers
Attention Mechanism	MHA-JAM [118]	2021	LSTM	CNN	Attwntion Head	LSTM	Each attention head to generate a distinct future trajectory to address multimodality
Attention Mechanism	mmTransformer [124]	2021	Transformer	VectorNet	Transformer	MLP	Stacked Transformers to refine a set of fixed proposals
GNN	VectorNet [137]	2020	PointNet [159]		GNN	MLP	Operate on the vectorized HD maps and trajectories
GNN	DenseTNT [140]	2021		VectorNet		Goal Set Predictor	Directly output a set of trajectories from dense goal candidates
Generative Model	TS-GAN [153]	2020	LSTM	-	Auto-Encoder Social Convolution	LSTM	Incorporate GAN into modeling spatial and temporal information
Generative Model	PRIME [36]	2021	1D-CNN, LSTM	LSTM	Self-Attention	Model-based Generator	Model-based generator and learning-based evaluator

achieve state-of-the-art results. Thus, we summarize the mainstream methods based on deep learning, give the state encoder, context encoder, interaction module, decoder, and summary description of these methods, as shown in Table IV. Using sequential networks to extract historical trajectory features, processing trajectory features through different network structures, extracting interaction information of traffic participants and road information, and using sequential networks to obtain the final predicted future trajectory, has become the mainstream research direction of trajectory prediction. Deep learning-based methods have reached the state-of-art results in trajectory prediction tasks and can predict longer time than physics-based methods and classic machine learning-based methods. At present, more and more autonomous vehicle trials use deep learning-based methods to predict the future trajectory of traffic participants.

VI. REINFORCEMENT LEARNING-BASED METHODS

In recent years, the rapid development of reinforcement learning (RL) provides a new way to understand high-dimensional complex policies [160]–[162], which provides new ideas for trajectory prediction tasks of AVs [163], [164]. When RL is used in the field of trajectory prediction for AVs, most methods use the Markov decision process (MDP) [165] to maximize the expected cumulative reward. A MDP is a tuple $(\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}, \gamma)$, where $\boldsymbol{S}$ is a finite set of states, $\boldsymbol{A}$ is a finite set of actions, $\boldsymbol{P}$ is a state transition probability matrix, $P^a_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$, $\boldsymbol{R}$ is a reward function, $R^{\mathrm{a}}_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$, and γ is a discount factor. To find the best decision process over all policies, the optimal state-value function $v_\star(s)$ and the optimal action-value function $q_\star(s, a)$ can be calculated as

$$v_*(s) = \max_a \left[R^a_s + \gamma \sum_{s' \in A} P^a_{ss'} v_*(s') \right],$$
$$q_*(s, a) = R^a_s + \gamma \sum_{s' \in A} P^a_{ss'} \max_{a'} q_*(s', a'). \tag{5}$$

Using MDP, the RL-based methods can be classified as Inverse Reinforcement Learning (IRL) methods, Generative Adversarial Imitation Learning (GAIL) methods, and Deep IRL (DIRL) methods, which will be discussed below.

A. Inverse Reinforcement Learning

Usually, MDP assumes that the reward function is already provided. However, the driver's behavior is always complicated such that manually specifying the weight of the reward function

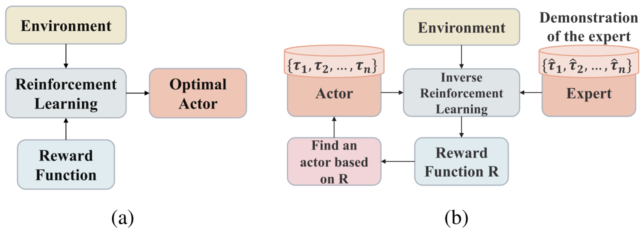

Fig. 13. Description of (a) RL and (b) IRL.

is inappropriate [166], [167]. IRL learns the reward function according to the expert demonstration (trajectory) to generate the corresponding optimal driving policy as shown in Fig. 13.

We divide IRL into maximum margin-based and maximum entropy-based methods according to the way of learning the weights of the reward function. Maximum margin-based methods optimize the reward function weights by minimizing the feature expectations between the expert demonstration and the predicted trajectory. In [168], the structured maximum margin is applied to learn mappings from features to reward and use these optimal policies in MDP to imitate expert's behavior. SCIRL is proposed by [169] which doesn't solve the direct RL problem but estimates the feature expectation of expert policies through structured classification. Silver *et al.* [170] use maximum margin planning framework to learn reward functions and learn driving maneuvers for AVs. However, most margin-based methods are ambiguous in the matching of feature expectations, because some degeneracies can also satisfy the optimal policy of expert demonstration.

Maximum entropy-based methods are more popular because they can use multiple reward functions to explain the ambiguity of experts' behavior [171], most of which are based on linear mapping and can be formulated as,

$$r(\Phi(s)) = \theta^\top \Phi(s), \tag{6}$$

where r is the approximation of reward function; Φ is a function to output the features of the state s, and the weight θ will be acquired by training. Several works apply maximum entropy-based IRL (MaxEnt-IRL) to behavior prediction for AVs. In [172], using MaxEnt-IRL acceptability-dependent behavior models are learned from expert's trajectories to generate the stochastic behavior, then the optimum behavior model is chosen by maximizing the social acceptability. Sharifzadeh *et al.* [173] leverage IRL with Deep Q-Networks (DQN) to extract the rewards with large state spaces. In [174], interaction-related factors are considered to accomplish probabilistic prediction for AVs. The distribution for future trajectories is formulated by driving maneuvers. A spatiotemporal state lattice is proposed by [175] to model driver behavior from expert's demonstrations.

Besides, some MaxEnt-IRL methods utilize sampled trajectories to accomplish prediction tasks. In [176], candidate trajectories are sampled first which will be selected with the minimal cost as the predicted trajectory. Wu *et al.* in [177] propose a method to learn the reward functions in the continuous domain by using the speed profile sampler to estimate the partition function. In [178], state sequences are sampled from the MaxEnt policy which will be fed to an attention-based trajectory generator to generate valued future trajectories. Xin *et al.* Ref. [179] use randomly pre-sampled policies in sub-space to approximate the optimal policy for reducing computational costs. In [180], an inverse optimal control (IOC) method using Langevin Sampling is proposed to learn the cost function of other vehicles in an energy-based generative model. Based on the decision-making mechanism, reward functions are learned using a polynomial trajectory sampler with discrete latent driving intentions in [181].

B. Generative Adversarial Imitation Learning

Ho *et al.* [191] propose GAIL in 2016, which uses the method of GAN to do imitation learning in RL. Instead of learning the reward function from experts' demonstration with IRL, GAIL directly extracts policies from data. Just as GAN, the core idea of GAIL is that the generator generates a trajectory similar to the expert trajectory as much as possible, and the discriminator tries to judge whether it is an expert trajectory as much as possible.

Many articles use GAIL to complete trajectory prediction for AVs. Kuefler *et al.* [182] extend GAIL to the optimization of RNN to demonstrate human driver behaviors, and policies and actions are evaluated by the discriminator. Li *et al.* [183] apply the information maximization theorem to extract the latent structure underlying expert demonstrations. In [184], a parameter-sharing extension of GAIL is proposed to model the interaction between multi-agent and can provide agents with domain-specific knowledge. To overcome the shortcomings of GAIL, which only models the next state using the current state, Choi *et al.* [185] propose a method combining a partially-observable Markov decision process (POMDP) within the GAIL framework, and the model is trained using the reward function from the discriminator.

C. Deep Inverse Reinforcement Learning

Since the prediction problem is nonlinear, it is necessary to use nonlinear mapping for generalizable function approximations. In [192], the deep inverse reinforcement learning (DIRL) framework is proposed to approximate complex and nonlinear reward functions, which can be expressed as,

$$r(\Phi(s)) = f(\theta, \Phi(s)), \tag{7}$$

where f is a nonlinear function. In this paper, a fully convolutional neural network (FCN) is applied in IRL for reward approximation. Some DIRL methods take historical trajectories as input. You *et al.* [186] consider the driving style and the road geometry, where the authors first use RL to design MDP, then learn the optimal driving policy from IRL, and use the deep neural network (DNN) to approximate the reward function. In [164], trajectories of traffic participants are encoded by LSTM and the reward network is learned by FCN.

Currently, more DIRL-based methods directly use raw perception data. Wulfmeier *et al.* [187] apply FCN for mapping the lidar data to traversability maps. The network is pre-trained to regress to a manual prior cost map and the initialize weights will be fine-tuned by the maximum entropy DIRL network.

TABLE V
SUMMARY OF REINFORCEMENT LEARNING-BASED METHODS

Reinforcement Learning-based Methods		IRL	GAIL	D-IRL
Contextual Factors	Physics-based Factors	[168]–[171], [173]–[181]	[182]–[185]	[164], [186]–[190]
	Road-related Factors	[168], [170], [173]–[181]	[182], [184], [185]	[186], [187], [189], [190]
	Interaction-related Factors	[171], [173], [174], [177], [180], [181]	[184]	[187], [190]
Output Types	Unimodal Trajectory	[168], [173], [175], [177]	[182]–[185]	[164], [186], [187], [189], [190]
	Multimodal Trajectory	[169]–[171], [174], [176], [178]–[181]		
	Intention	[170], [171], [173]–[176], [181]	[182]–[184]	[164], [186]–[190]

TABLE VI
DATASETS FOR AVS WHICH UTILIZED IN TRAJECTORY PREDICTION

Dataset	Year	Agents	Sensors	Scene	Duration and tracking quantity	Data type	Typical methods
NuScenes [195]	2020	vehicles pedestrians cyclists	lidar camera	urban	1000 driving scenes	trajectories, HD map	MHA-JAM [118], Trajectron++ [145]
Waymo Open Dataset [196]	2020	vehicles pedestrians cyclists	lidar camera	urban	103354, 20s 10Hz segments	trajectories, HD map	DenseTNT [140], Scene Transformer [123]
Lyft Level 5 [197]	2020	vehicles pedestrians cyclists	lidar camera	urban	1000+ hours, 16K miles of data from 23 vehicles	trajectories, HD map	Graph-LSTM [133]
Argoverse [77], [198]	2019	vehicles	lidar camera	urban	324,557 interesting vehicle trajectories, 1000 driving hours	trajectories, HD map	VectorNet [137], LaneRCNN [141]
INTERACTION [199]	2019	vehicles pedestrians	drone camera	urban highway	11 locations, 40000 vehicles	trajectories, HD map	IPTM [87]
HighD [200]	2018	vehicles	drone	highway	110500 vehicles, 147 driven hours	trajectories, lane	MHA-LSTM [117]
Apolloscape [129]	2018	vehicles pedestrians cyclists	lidar camera	urban	1000km trajectories	trajectories	GRIP [127]
KITTI [201]	2013	vehicles pedestrians cyclists	lidar camera	urban highway	50 sequences	image, point cloud	DESIRE [107], MANTRA [99]
NGSIM [194]	2006	vehices	camera	highway	90 min recording of two highways	trajectories, lane	CS-LSTM [102], TS-GAN [153]

In [188], using camera image, the driving behavior is modeled by DIRL, where the CNN is to extract the associated state features. Zhu *et al.* [189] use RL ConvNet and state visiting frequency (SVF) ConvNet to encode the vehicle's kinematics and obtain the weight of the reward function by back-propagating the loss gradient [193] between expert SVF from expert demonstration and policy SVF from lidar data. In [190], a convolutional LSTM considering the inertial, environment, and social is proposed to extract the feature map from lidar and trajectory data, which will be incorporated into the output reward map to predict the traversability map.

D. Summary

In summary, reinforcement learning-based trajectory prediction methods for AVs can be classified into Table V. Such methods use MDP to maximize the expected cumulative reward and generate optimal driving policies by learning expert demonstrations, most of which are planning-based methods. Combining with deep learning networks, these methods can better extract expert demonstrations and consider more factors. However, Most of them are computationally intensive and require long training periods.

VII. EVALUATION

The appearance of a variety of datasets has facilitated the performance of the learning-based prediction algorithms. Therefore, it is necessary to choose suitable metrics to evaluate the performance of each algorithm. This section will first introduce several datasets, then introduce the performance-evaluation metrics, finally the performance of the aforementioned works using different methods on the same NGSIM dataset [194] will be compared.

A. Datasets

To evaluate the quality of the trajectory prediction model, the predicted trajectory is usually compared with the ground truth trajectory, which is obtained from various datasets. These datasets are collected by sensors, such as lidar and cameras, and manually annotated or automatically generated to produce sequences of vehicles' movements.

The popular datasets used in trajectory prediction are summarized in Table VI. This paper introduces the datasets in reverse chronological order and lists the typical methods that use the dataset for trajectory prediction. Most of the methods mentioned in this paper take trajectories as input and some also use vehicle states or map information. However, since most trajectories in these datasets are obtained by learning methods from images or point clouds, some models directly use images or point clouds as input for end-to-end trajectory prediction.

B. Evaluation Metrics

Several evaluation metrics are usually used for vehicle trajectory prediction.

1) Root Mean Squared Error (RMSE): RMSE calculates the square root of the average of squared prediction error:

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{t=1}^{n}\left(Y_{\text{pred}}^{t} - Y_{GT}^{t}\right)^2}, \tag{8}$$

where n is the number of data samples in the prediction horizon, Y_{pred}^{t} and Y_{GT}^{t} are predicted results and ground truth trajectory at sample time t correspondingly. RMSE is sensitive to large prediction errors and one of the commonly used metrics for trajectory prediction.

2) Negative Log Likelihood (NLL): For a modeled trajectory distribution $f(Y)$:

$$NLL = -log(f(Y)), \tag{9}$$

where Y represents the ground truth trajectory and the NLL value is not a physical quantity. RMSE is used to calculate the models' average error, while NLL is more focused on determining the correctness of the trajectory in the maneuver-based models.

3) Average displacement error (ADE): The average L2 distance between the predicted trajectory and the ground truth.

$$ADE = \frac{1}{N_p \times T} \sum_{i=1}^{N_p} \sum_{t=1}^{T} \left| Y_{\text{pred}}^{t}[i] - Y_{GT}^{t}[i] \right|, \tag{10}$$

where N_p represents all the predicted objects, and T represents the prediction time. For multimodal prediction, minimum ADE (**mADE**) is usually used to indicate the minimum value of ADE over K predictions.

4) Final displacement error (FDE): The L2 distance between the final predicted results and the corresponding ground truth positions.

$$FDE = \frac{1}{N_p} \sum_{i=1}^{N_p} \left| Y_{\text{pred}}^{T}[i] - Y_{GT}^{T}[i] \right|, \tag{11}$$

where Y_{pred}^{T} and Y_{GT}^{T} are predicted results and ground truth at the final time step T correspondingly. For multimodal prediction, minimum FDE (**mFDE**) is usually used to indicate the minimum value of FDE over K predictions.

5) Miss Rate (MR): Based on the L2 distance of the final position, the ratio of cases where the predicted trajectory is not within 2.0 meters of the ground truth.
When the prediction results are multi-modal, assuming that the prediction results are K likely future trajectories, ADE, FDE and MR will be judged according to the optimal future trajectory, and they will be recorded as ADE_K, FDE_K, MR_K respectively.

6) Computation Time: Computation time is very important for the on-board performance of the method. The computing power of autonomous vehicles is limited, but the trajectory prediction model is generally complex and requires a huge computational resources. To achieve higher levels for autonomous driving, the computation of each module must be relatively fast to reduce the delay as small as possible. Therefore, the real-time performance or computational cost is very important for the model.

7) Prediction Horizon: Prediction horizon refers to the time steps in the future that can be predicted by the model. Generally, the longer the prediction horizon is, the lower the accuracy will be in a dynamics or even stochastic driving environment. However, to meet the requirements of the planning and control system, the trajectory prediction results with a certain period time should be fed into the system, such that the prediction time should not be too short and in accordance with other module.

C. Performance of Different Methods

For real-world autonomous driving, accuracy is one of the most important metrics for trajectory prediction methods. To allow readers to better compare the various methods and their accuracy, this paper compares the performance of trajectory-prediction methods on highway and urban scenes respectively. In Table VII, methods based on NGSIM I-80 and US-101 highway driving datasets [194] are compared using RMSE, while we use minADE, minFDE, MR to compare methods based on Argoverse [77] in Table VIII, which is recorded under urban conditions and the prediction horizon is 3 seconds. It can be seen from Tables VII and VIII, the longer the prediction time, the lower the prediction accuracy, and most learning-based methods surpass conventional methods. Besides, multimodal prediction is more consistent with human cognitive process and multimodal prediction is more accurate than unimodal prediction. GNN performs well in Table VII with the ability to capture structure road features, such that some state-of-the-art methods use GNN to encode HD map information and complete trajectory prediction. At present, most of the latest trajectory prediction methods use deep learning, but for AVs to carry out safer planning and control, trajectory-prediction methods need to be more accurate.

D. Applications

Since trajectory prediction plays an important role in ensuring the safety of AVs, major autonomous driving teams have embedded the trajectory prediction module into the development of AVs above the L4 level. However, due to the confidentiality of the software, many autonomous driving manufacturers have not mentioned the specific algorithm they use, so this section only summarizes the trajectory prediction methods used by the autonomous driving teams that have been clearly announced. Early real-world studies use physics-based methods for trajectory prediction [23]. Next, BMW uses Dynamic Bayesian Networks to determine the driving intentions of surrounding vehicles and performs experiments on highways [204]. The pioneer IV autonomous vehicle of the University of Science and Technology of China uses a knowledge-driven approach to obtain the future lane of the predicted vehicle and then uses LSTM to predict its future trajectory [205]. For the Baidu Apollo autonomous vehicle [206], a new model named Inter-TNT based on the advanced method TNT [139], is introduced as the prediction module. With the advancement of autonomous driving technology, more and more advanced and complex trajectory prediction methods will be applied to real vehicles.

VIII. Discussion and Directions

This section will discuss the advantages and disadvantages of different categories for trajectory prediction, and outline potential research directions to guide readers in this field.

TABLE VII
COMPARISON FOR TRAJECTORY PREDICTION METHODS FOR AVS BASED ON THE HIGHWAY DRIVING DATASET NGSIM

Classification	Models	RMSE(m)				
		1S	2S	3S	4S	5S
Single Trajectory methods	Constant Velocity [102]	0.73	1.78	3.13	4.78	6.68
Kalman Filtering methods	IMM-KF [32]	0.58	1.36	2.28	3.37	4.55
HMM	C-VGMM+VIM [55], [152]	0.66	1.56	2.75	4.24	5.99
RNN	M-LSTM [82]	0.58	1.26	2.12	3.24	4.66
RNN	MFP-1 [84]	0.54	1.16	1.90	2.78	3.83
CNN and RNN	CS-LSTM(M) [102]	0.62	1.29	2.13	3.20	4.52
Attention Mechanism	MHA-LSTM [117]	0.41	1.01	1.74	2.67	3.83
GNN	GRIP++ [128]	0.38	0.89	1.45	**2.14**	**2.94**
GNN	GISNet [135]	**0.33**	**0.83**	**1.42**	2.14	3.23
Generative Model	MATF-GAN [152]	0.66	1.34	2.08	2.97	4.13
Generative Model	TS-GAN [153]	0.60	1.24	1.95	2.78	3.72
IRL	L-IRL [164], [202]	1.12	2.29	2.31	3.38	4.45
GAIL	GAIL-GRU [164], [182]	0.69	1.51	2.55	3.65	4.71
DIRL	MEDIRL [164], [187]	1.35	2.57	2.83	3.69	4.88
DIRL	DN-IRL [164], [203]	0.54	1.02	1.91	2.43	3.76

TABLE VIII
COMPARISON FOR TRAJECTORY PREDICTION METHODS FOR AVS BASED ON ARGOVERSE UNDER URBAN CONDITIONS

Classification	Models	K=6			K=1		
		minFDE[1]	minADE[1]	MR[2]	minFDE[1]	minADE[1]	MR[2]
Physics-based	CV [77]	7.57	3.39	0.82	7.89	3.53	0.84
Classic Machine Learning-based	NN+map [77]	4.03	2.08	0.58	8.12	3.65	0.84
RNN	LSTM+map [77]	5.44	2.34	0.69	6.81	2.96	0.81
RNN	Jean [85]	1.49	0.93	0.19	4.18	1.86	0.63
Attention Mechanism	SceneTransformer [123]	**1.23**	**0.80**	0.13	-	-	-
Attention Mechanism	mmTransformer [124]	1.34	0.84	0.15	-	-	-
GNN	LaneGCN [138]	1.36	0.87	0.16	3.78	1.71	0.59
GNN	DenseTNT [140]	1.45	0.93	**0.11**	-	-	-
GNN	LaneRCNN [141]	1.45	0.90	0.12	**3.69**	**1.69**	**0.57**
Generative Model	PRIME [36]	1.56	1.22	0.12	3.82	1.91	0.59

[1] minADE/ minFDE: in meters.
[2] MR: the threshold for endpoint error is 2 m.

TABLE IX
THE PERFORMANCE OF THE TRAJECTORY PREDICTION METHODS

Methods	Accuracy	Prediction Horizon	Computation Cost	Applications
Physics-based	High in short-term prediction, low in other prediction horizon	Short	Small	Colision risk analysis
Classic Machine Learning-based	Good at recognizing maneuvers but generalization ability is poor	Medium	Medium	Maneuver recognition
Deep Learning-based	High in considering some factors	Long	Relatively high	More and more applied in real-world
Reinforcement Learning-based	Relatively high, prediction methods are relatively few	Long	High	More applied in planning

A. Discussion

This section discusses the performance of the trajectory prediction methods in terms of accuracy, computation time, prediction horizon, etc., analyzes its practical applications in AVs, and gives a summary in Table IX. Note that, we refer to short-term and long-term prediction to characterize prediction horizons of no more than 1 s and no less than 3 seconds, respectively.

1) Physics-Based Methods: they are suitable for the movement of vehicles, which can be accurately described by kinematics or dynamics models. Given a suitable physics model, these methods can be applied to a variety of scenarios at small computational cost and in a short time but without training. However, the prediction results based on such models heavily depends on the inputs and the model selection. The inputs are closely related to human or machine drivers, influenced by the driving environment or the interactions with other participants. Therefore, without the capability to describe such factors, physics-based models are limited to short-term prediction and in static scenes. Because of its simplicity and fast response, these methods can be easily used in real applications for AVs, such as collision risk analysis.

2) Classic Machine Learning-Based Methods: compared with physics-based methods, this type of method is able to consider more factors and its accuracy is relatively high with a longer prediction length at a higher computing cost. Most of these methods are maneuver-based methods, which predicts the trajectory with the maneuver known as a prior. However, vehicle maneuvers of human drivers are usually diverse and vary greatly in different scenarios such that the generalization ability of is poor. In real applications for AVs, such methods are used in scenarios such as lane change studies, leveraging their advantages in maneuver recognition.

3) Deep Learning-Based Methods: traditional trajectory prediction methods for AVs are only suitable for simple scenes and short-term prediction, but deep learning-based methods can make accurate prediction in a longer prediction horizon. By using RNN, CNN, GNN and other networks for feature extraction, interaction-related factors and map information are considered. Among them, it can adapt to more complex environments and a longer time horizon. Deep learning-based methods require to use a large amount of data for training. Besides, with the increase of consideration factors and the increase of the number of network layers, the computing costs and time increases sharply. Such methods can naturally generate multi-modal trajectories, which is consistent with the diversity of vehicles' maneuvers. In real applications for AVs, it is necessary to reach a balance between calculation time and model complexity to ensure the real-time performance and safety of AVs. At present, more and more real-world trials use these methods to predict the future trajectory of traffic participants.

4) Reinforcement Learning-Based Methods: they imitate the human decision-making process and obtain the reward function through learning the expert demonstration to generate the corresponding optimal driving policy. They can continuously evolve through learning and adapt to complex environments and long prediction horizons. Such methods probably generate higher accuracy trajectories than deep learning methods in a longer time domain. However, most of these methods are typically computationally expensive in their recovery of an expert cost function and require long training times. In real applications for AVs, reinforcement learning-based trajectory prediction methods are more applied to trajectory planning, taking its advantages in the decision-making process.

B. Potential Research Directions

With the continuous advancement of autonomous driving technology, the importance of trajectory prediction has been paid more and more attention. The trajectory prediction method has been developed from the traditional Kalman filter methods to the learning-based methods, which can handle more complex scenes. After summarizing the methods of the past two decades, this paper outlines the potential research directions as shown in Fig. 14 and discusses as follows.

1) Inclusion of more information: It can be seen that the methods based on the interaction-aware factors and map information are more suitable for real application scenes and are currently one of the most popular development directions. However, much more information needs to be considered in addition to the interaction-related factors. For example, most of the current methods do not consider the constraints based on explicit traffic rules, but in real scenes, traffic rules can reshape the maneuvers or even trajectories of vehicles. Similarly, information such as traffic lights, road signs, etc., can be also used as reliable inputs for prediction. In addition, other useful audio-visual information, such as vehicle turn signals, vehicle horns, etc., can be used as references for prediction. In the future, researchers are encouraged to use more information for trajectory prediction.
2) Introduction of more advanced algorithms: Just like the outstanding achievements of the Transformer model in the field of NLP [119], by introducing more advanced algorithms it can achieve higher prediction accuracy under the same input data. The current algorithm achieves high accuracy by adding HD maps, considering interaction-related factors, and generating the multimodal trajectory that conforms to the multi-modality of human intentions. In addition, more advanced algorithms need to be continuously proposed to further improve the ability of trajectory prediction algorithms with new structures and training methods. With the continuous iterative upgrade of the autonomous driving system, it has become the general trend to improve the predictive ability of AVs and meet the safety requirements of autonomous driving through more advanced algorithms.
3) Integration other key technologies of AVs: The effectiveness of the whole system can be greatly improved when the trajectory prediction results are considered for decision making, trajectory planning, and motion control. Take the motion control system as an example, most of the current motion control systems regard the movement of traffic participants as uniform linear motion, which is quite different from the real trajectory of traffic participants. When the trajectory prediction model is integrated, the local decision-making planning control system can better cope with the environment's changes and improve the safety of autonomous driving.
4) Improvement of model robustness: Most of the datasets are semi-automatically annotated and the ground truth trajectories have measurement noises. In real applications for AVs, various noises exist in the perception system, include tracking errors, location errors, map errors, etc., which will bring deviations and uncertainty. Therefore, robustness

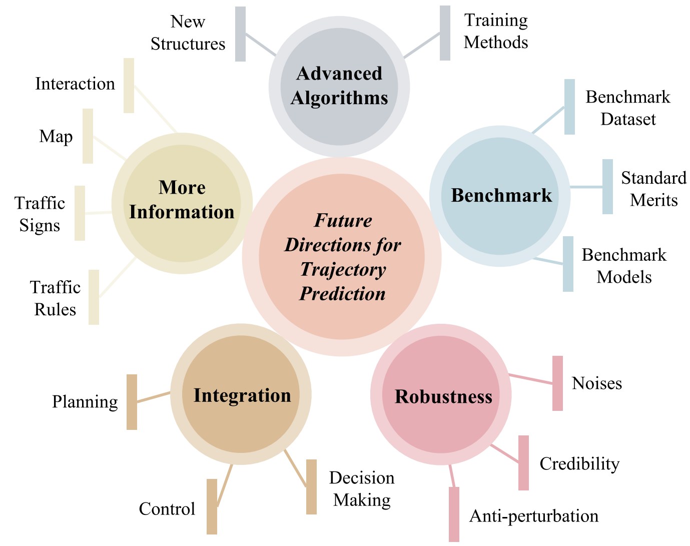

Fig. 14. Illustration of Potential Research Directions.

should be considered to improve the anti-perturbation ability of the real application for AVs. In addition, besides the location metrics (such as ADE, FDE), probabilistic metrics (NLL, mADE, mFDE) should also be applied to improve the credibility of the method and make the model better applicable to real-world autonomous driving.

5) Establishment of a benchmark: A benchmark is needed, with a standard unified metric and a map-available dataset in a more complex environment. This benchmark should allow a long-term and multi-modal prediction with obstacle avoidance scenes and non-convex constraints, and allow the use of different history horizons to predict future trajectories of different prediction horizons. Besides, a test set is needed to make an inference on the trained model, and make the computation time as a unified comparison. Moreover, in the real applications for AVs, since good perception and tracking are not always completed, the benchmark dataset should include test sets with inaccurate ground truth values, to be more suitable for real applications and better used for AVs.

IX. Conclusion

In this paper, a thorough analysis of the trajectory-prediction problem for AVs and the taxonomy for trajectory-prediction methods are proposed. Trajectory-prediction methods for AVs are comprehensively reviewed, which include the physics-based methods, the classic machine learning-based methods, the deep learning-based methods, and the reinforcement learning-based methods. The performance of each kind of method and the opportunities for applying it to real-world autonomous driving are discussed. Recent advances in trajectory prediction for AVs are encouraging, but it still faces various challenges and has potential research directions in the future which we have outlined to guide readers in this field.

Safety is crucial for autonomous driving. To break through the bottleneck of AVs and ensure their safety, AVs need to predict their surroundings just like human drivers. We hope our survey will improve the application of prediction systems in AVs and stimulate further research along the directions discussed.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable suggestions.

References

[1] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: A survey," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 315–329, Mar. 2020.

[2] F.-Y. Wang, "Metavehicles in the metaverse: Moving to a new phase for intelligent vehicles and smart mobility," *IEEE Trans. Intell. Veh.*, vol. 7, no. 1, pp. 1–5, Mar. 2022.

[3] D. Cao *et al.*, "Future directions of intelligent vehicles: Potentials, possibilities, and perspectives," *IEEE Trans. Intell. Veh.*, vol. 7, no. 1, pp. 6–10, Mar. 2022.

[4] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech J.*, vol. 1, no. 1, pp. 1–14, 2014.

[5] M. S. Shirazi and B. T. Morris, "Looking at intersections: A survey of intersection monitoring, behavior and safety analysis of recent studies," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 4–24, Jan. 2017, doi: 10.1109/TITS.2016.2568920.

[6] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. A. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behaviour prediction for autonomous driving applications: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 33–47, 2020.

[7] F. Leon and M. Gavrilescu, "A review of tracking and trajectory prediction methods for autonomous driving," *Mathematics*, vol. 9, no. 6, pp. 660–696, 2021.

[8] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Q.-H. Meng, "A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving," in *Proc. IEEE Int. Conf. Robot. Biomim.*, 2021, pp. 978–985, doi: 10.1109/ROBIO54168.2021.9739407.

[9] K. Santhoshk, P. Dograd, and P. Royp, "Anomaly detection in road traffic using visual surveillance: A survey," *ACM Comput. Surv.*, vol. 53, no. 6, pp. 1–26, 2020.

[10] A. Ben Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: A review," *Expert Syst. Appl.*, vol. 91, pp. 480–491, Jan. 2018.

[11] D. Ridel, E. Rehder, M. Lauer, C. Stiller, and D. Wolf, "A literature review on the prediction of pedestrian behavior in urban scenarios," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 3105–3112.

[12] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 895–935, 2020.

[13] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2569–2578.

[14] S. Chen *et al.*, "Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G," *IEEE Commun. Standards Mag.*, vol. 1, no. 2, pp. 70–76, Jul. 2017.

[15] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *Proc. IEEE 5th Int. Conf. Intell. Comput. Commun. Process.*, 2009, pp. 417–422.

[16] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 319–324.

[17] C. F. Lin and A. G. Ulsoy, "Vehicle dynamics and external disturbance estimation for vehicle path prediction," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 3, pp. 508–518, May 2000.

[18] R. Pepy, A. Lambert, and H. Mounier, "Reducing navigation errors by planning with realistic vehicle model," in *Proc. Intell. Veh. Symp.*, 2006, pp. 300–307.

[19] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 678–687, Dec. 2009.

[20] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Proc. 11th Int. Conf. Inf. Fusion*, 2008, pp. 1–6.

[21] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 549–562, Sep. 2007.

[22] P. Lytrivis, G. Thomaidis, and A. Amditis, "Cooperative path prediction in vehicular environments," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, 2008, pp. 803–808.

[23] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?," in *Proc. Intell. Veh. Symp.*, 2008, pp. 1068–1073.

[24] T. Batz, K. Watson, and J. Beyerer, "Recognition of dangerous situations within a cooperative group of vehicles," in *Proc. Intell. Veh. Symp.*, 2009, pp. 907–912.

[25] M. Brannstrom, E. Coelingh, and J. Sjoberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 658–669, Sep. 2010.

[26] R. Miller and Q. Huang, "An adaptive peer-to-peer collision warning system," in *Proc. IEEE Veh. Technol. Conf.*, 2002, pp. 317–321.

[27] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach-its situation assessment, decision making, and performance tradeoffs," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 528–540, Dec. 2006, doi: 10.1109/TITS.2006.883115.

[28] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. J. Dietmayer, "IMM object tracking for high dynamic driving maneuvers," in *Intell. Veh. Symp.*, 2004, pp. 825–830.

[29] B. Jin, J. Bo, S. Tao, H. Liu, and G. Liu, "Switched kalman filter-interacting multiple model algorithm based on optimal autoregressive model for manoeuvring target tracking," *IET Radar Sonar Navigation*, vol. 9, no. 2, pp. 199–209, 2015.

[30] H. Dyckmanns, R. Matthaei, M. Maurer, B. Lichte, and D. Stuker, "Object tracking in urban intersections based on active use of a priori knowledge: Active interacting multi model filter," in *Proc. Intell. Veh. Symp.*, 2011, pp. 625–630.

[31] R. Zhang, L. Cao, S. Bao, and J. Tan, "A method for connected vehicle trajectory prediction and collision warning algorithm based on v2v communication," *Int. J. Crashworthiness*, vol. 22, no. 1, pp. 15–25, 2017.

[32] V. Lefkopoulos, M. Menner, A. Domahidi, and M. N. Zeilinger, "Interaction-aware motion prediction for autonomous driving: A multiple model Kalman filtering scheme," *IEEE Robot. Automat. Lett.*, vol. 6, no. 1, pp. 80–87, Oct. 2021.

[33] M. Althoff and A. Mergel, "Comparison of Markov chain abstraction and monte carlo simulation for the safety assessment of autonomous cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1237–1247, Dec. 2011.

[34] K. Okamoto, K. Berntorp, and S. Di Cairano, "Driver intention-based vehicle threat assessment using random forests and particle filtering," *IFAC-PapersOnLine*, vol. 50, pp. 13 860–13 865, 2017.

[35] Y. Wang, Z. Liu, Z. Zuo, Z. Li, L. Wang, and X. Luo, "Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8546–8556, Sep. 2019.

[36] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, "Learning to predict vehicle trajectories with model-based planning," in *Conf. Robot Learn.*, 2022, pp. 1035–1045.

[37] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School Machine Learning*, Berlin, Heidelberg, Germany: Springer, 2003, pp. 63–71.

[38] J. Joseph, F. Doshi velez, and N. Roy, "A Bayesian nonparametric approach to modeling mobility patterns," *Autonom. Robot.*, vol. 31, no. 4, pp. 383–400, 2011.

[39] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, "A Bayesian nonparametric approach to modeling motion patterns," *Auton. Robots*, vol. 31, no. 4, pp. 383–400, 2011.

[40] Q. Tran and J. Firl, "Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression," in *Proc. IEEE Intell. Veh. Symp.*, 2014, pp. 918–923.

[41] C. Laugier *et al.*, "Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety," *IEEE Intell. Transp. Syst. Mag.*, vol. 3, no. 4, pp. 4–19, 2011, doi: 10.1109/MITS.2011.942779.

[42] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 797–803.

[43] Y. Guo *et al.*, "Modeling multi-vehicle interaction scenarios using Gaussian random field," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 3974–3980.

[44] D. A. V. Govea and T. Fraichard, "Motion prediction for moving objects: A statistical approach," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2004, pp. 3931–3936.

[45] C. Hermes, C. Wohler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," in *Proc. Intell. Veh. Symp.*, 2010, pp. 652–657.

[46] H. Mandalia and D. Salvucci, "Using support vector machines for lane-change detection," in *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, vol. 49, 2005, pp. 1965–1969.

[47] P. Kumar, M. Perrollaz, S. Lefèvre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *Proc. IEEE Intell. Veh. Symp.*, 2013, pp. 797–802.

[48] G. S. Aoude and J. P. How, "Using support vector machines and Bayesian filtering for classifying agent intentions at road intersections," Sep. 2009.

[49] G. S. Aoude, B. D. Luders, K. K. H. Lee, D. S. Levine, and J. P. How, "Threat assessment design for driver assistance system at intersections," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, 2010, pp. 1855–1862.

[50] S. Gambs, M.-O. Killijian, and M. Nunez del Prado Cortez, "Next place prediction using mobility Markov chains," in *Proc. First Workshop Meas., Priv., Mobil.*, 2012, pp. 1–6.

[51] Q. Deng and D. Söffker, "Improved driving behaviors prediction based on fuzzy logic-hidden Markov model (fl-hmm)," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 2003–2008.

[52] H. Berndt, J. Emmert, and K. Dietmayer, "Continuous driver intention recognition with hidden Markov models," in *Proc. Int. IEEE Conf. Intell. Transp. Syst.*, 2008, pp. 1189–1194.

[53] S. Qiao, D. Shen, X. Wang, N. Han, and W. Zhu, "A self-adaptive parameter selection trajectory prediction approach via hidden Markov models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 284–296, Feb. 2015.

[54] Y. Wang, C. Wang, W. Zhao, and C. Xu, "Decision-making and planning method for autonomous vehicles based on motivation and risk assessment," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 107–120, Jan. 2021.

[55] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A unified framework for maneuver classification and motion prediction," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 129–140, Jun. 2018.

[56] S. Zhang, Y. Zhi, R. He, and J. Li, "Research on traffic vehicle behavior prediction method based on game theory and hmm," *IEEE Access*, vol. 8, pp. 30210–30222, Feb. 2020.

[57] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[58] K. P. Murphy, "Dynamic Bayesian networks: Representation, inference and learning," Ph.D. dissertation, Univ. California, Berkeley, 2002.

[59] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 1, pp. 69–79, 2015, doi: 10.1109/MITS.2014.2357038.

[60] M. Schreier, V. Willert, and J. Adamy, "An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2751–2766, Oct. 2016.

[61] M. Bahram, A. Lawitzky, J. Friedrichs, M. Aeberhard, and D. Wollherr, "A game-theoretic approach to replanning-aware interactive scene prediction and planning," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3981–3992, Jun. 2015.

[62] G. He, X. Li, Y. Lv, B. Gao, and H. Chen, "Probabilistic intention prediction and trajectory generation based on dynamic Bayesian networks," in *Proc. Chin. Automat. Congr.*, 2019, pp. 2646–2651.

[63] J. Li, B. Dai, X. Li, X. Xu, and D. Liu, "A dynamic Bayesian network for vehicle maneuver prediction in highway driving scenarios: Framework and verification," *Electronics*, vol. 8, no. 1, pp. 40–58, Jan. 2019.

[64] Y. Li, X. Y. Lu, J. Wang, and K. Li, "Pedestrian trajectory prediction combining probabilistic reasoning and sequence learning," *IEEE Trans. Intell. Veh.*, vol. 5, no. 3, pp. 461–474, Sep. 2020.

[65] T. Heskes and O. Zoeter, "Expectation propagation for approximate inference in dynamic Bayesian networks," in *Proc. UAI-2002*, 12, Dec. 2012.

[66] G. Weidl, A. L. Madsen, D. Kasper, and G. Breuel, "Optimizing Bayesian networks for recognition of driving maneuvers to meet the automotive requirements," in *Proc. IEEE Int. Symp. Intell. Control*, 2014, pp. 1626–1631.

[67] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*.

[68] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[69] A. Zyner, S. Worrall, and E. Nebot, "A recurrent neural network solution for predicting driver intention at unsignalized intersections," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1759–1764, Jul. 2018.

[70] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in *Proc. Intell. Veh. Symp.*, 2017, pp. 1484–1489.

[71] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 1665–1670.

[72] F. Altché and A. D. L. Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 353–359.

[73] W. Ding and S. Shen, "Online vehicle trajectory prediction using policy anticipation network and optimization-based context reasoning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 9610–9616.

[74] A. Zyner, S. Worrall, and E. Nebot, "Naturalistic driver intention and path prediction using recurrent neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1584–1594, Apr. 2019.

[75] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1672–1678.

[76] Y. Xing, C. Lv, and D. Cao, "Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1341–1352, Feb. 2020.

[77] M.-F. Chang *et al.*, "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8748–8757.

[78] A. Kawasaki and A. Seki, “Multimodal trajectory predictions for urban environments using geometric relationships between a vehicle and lanes,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9203–9209.
[79] S. Dai, L. Li, and Z. Li, “Modeling vehicle interactions via modified lstm models for trajectory prediction,” *IEEE Access*, vol. 7, pp. 38287–38296, 2019.
[80] W. Ding, J. Chen, and S. Shen, “Predicting vehicle behaviors over an extended horizon using behavior interaction network,” in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 8634–8640.
[81] L. Xin, P. Wang, C. Y. Chan, J. Chen, and B. Cheng, “Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks,” in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2018, pp. 1441–1446.
[82] N. Deo and M. M. Trivedi, “Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMS,” in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1179–1184.
[83] K. Min, D. Kim, J. Park, and K. Huh, “Rnn-based path prediction of obstacle vehicles with deep ensemble,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10252–10256, Oct. 2019.
[84] C. Tang and R. R. Salakhutdinov, “Multiple futures prediction,” *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 15424–15434, Nov. 2019.
[85] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, “Multi-head attention for multi-modal joint vehicle motion forecasting,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9638–9644.
[86] X. Li, Y. Liu, K. Wang, and F.-Y. Wang, “A recurrent attention and interaction model for pedestrian trajectory prediction,” *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 5, pp. 1361–1370, Sep. 2020.
[87] T. Zhang, W. Song, M. Fu, Y. Yang, and M. Wang, “Vehicle motion prediction at intersections based on the turning intention and prior trajectories model,” *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 10, pp. 1657–1666, Oct. 2021.
[88] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.
[89] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[90] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 1243–1252.
[91] N. Nikhil and B. Tran Morris, “Convolutional neural network for trajectory prediction,” in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018.
[92] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, “Covernet: Multimodal behavior prediction using trajectory sets,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14074–14083.
[93] H. Cui *et al.*, “Deep kinematic models for kinematically feasible vehicle trajectory predictions,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 10563–10569.
[94] H. Cui *et al.*, “Multimodal trajectory predictions for autonomous driving using deep convolutional networks,” in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 2090–2096.
[95] F.-C. Chou *et al.*, “Predicting motion of vulnerable road users using high-definition maps and efficient convnets,” in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1655–1662.
[96] N. Djuric *et al.*, “Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving,” in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2095–2104.
[97] J. Strohbeck *et al.*, “Multiple trajectory prediction with deep temporal and spatial convolutional neural networks,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 1992–1998.
[98] Y. Zhang, Y. Zou, J. Tang, and J. Liang, “A lane-changing prediction method based on temporal convolution network,” 2020, *arXiv:2011.01224*.
[99] F. Marchetti, F. Becattini, L. Seidenari, and A. D. Bimbo, “Mantra: Memory augmented networks for multiple trajectory prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7143–7152.
[100] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, “Home: Heatmap output for future motion estimation,” in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 500–507.
[101] M. Ye, T. Cao, and Q. Chen, “Tpcn: Temporal point cloud networks for motion forecasting,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11318–11327.
[102] N. Deo and M. M. Trivedi, “Convolutional social pooling for vehicle trajectory prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1468–1476.
[103] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, and S. Savarese, “Social lstm: Human trajectory prediction in crowded spaces,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 961–971.
[104] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, “Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8483–8492.
[105] G. Xie, A. Shangguan, R. Fei, W. Ji, and X. Hei, “Motion trajectory prediction based on a CNN-LSTM sequential model,” *Sciece China. Inf. Sci.*, vol. 63, no. 11, pp. 1–21, Nov. 2020.
[106] S. Casas, C. Gulino, S. Suo, and R. Urtasun, “The importance of prior knowledge in precise multimodal prediction,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 2295–2302.
[107] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, “Desire: Distant future prediction in dynamic scenes with interacting agents,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 336–345.
[108] J. Hong, B. Sapp, and J. Philbin, “Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8454–8462.
[109] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, “Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction,” 2019, *arXiv:1910.05449*.
[110] D. Hu, “An introductory survey on attention mechanisms in NLP problems,” in *Proc. SAI Intell. Syst. Conf.*, Springer, 2019, pp. 432–448.
[111] V. Mnih *et al.*, “Recurrent models of visual attention,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.
[112] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, *arXiv:1409.0473*.
[113] D. Varshneya and G. Srinivasaraghavan, “Human trajectory prediction using spatially aware deep attention models,” 2017, *arXiv:1705.09436*.
[114] S. Haddad, M. Wu, H. Wei, and S. K. Lam, “Situation-aware pedestrian trajectory prediction with spatio-temporal attention model,” 2019, *arXiv:1902.05437*.
[115] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, “Soft hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection,” *Neural Netw.*, vol. 108, pp. 466–478, Dec. 2018.
[116] H. Kim, D. Kim, G. Kim, J. Cho, and K. Huh, “Multi-head attention based probabilistic vehicle trajectory prediction,” in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1720–1725.
[117] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, “Attention based vehicle trajectory prediction,” *IEEE Trans. Intell. Veh.*, vol. 6, no. 1, pp. 175–185, May 2020.
[118] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, “Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation,” in *Proc. IEEE Intell. Veh. Symp.*, 2021, pp. 165–170.
[119] A. Vaswani *et al.*, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[120] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, “Transformer networks for trajectory forecasting,” in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 10335–10342.
[121] Z. Huang, X. Mo, and C. Lv, “Multi-modal motion prediction with transformer-based neural network for autonomous driving,” 2021, *arXiv:2109.06446*.
[122] L. L. Li *et al.*, “End-to-end contextual perception and prediction with interaction transformer,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5784–5791.
[123] J. Ngiam *et al.*, “Scene transformer: A unified multi-task model for behavior prediction and planning,” in *Proc. Int. Conf. Learn. Representations*, 2021.
[124] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, “Multimodal motion prediction with stacked transformers,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7577–7586.
[125] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2020.
[126] F. Diehl, T. Brunner, M. T. Le, and A. Knoll, “Graph neural networks for modelling traffic participant interaction,” in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 695–701.

[127] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interaction-aware trajectory prediction," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 3960–3966.

[128] X. Li, X. Ying, and M. C. Chuah, "Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving," 2019, *arXiv:1907.07792*.

[129] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 2702–2719.

[130] H. Jeon, J. Choi, and D. Kum, "SCALE-Net: Scalable vehicle trajectory prediction network under random number of interacting vehicles via edge-enhanced graph convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 2095–2102.

[131] L. Gong and Q. Cheng, "Exploiting edge features in graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9211–9219.

[132] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14424–14432.

[133] R. Chandra *et al.*, "Forecasting trajectory and behavior of road-agents using spectral clustering in graph-LSTMS," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4882–4890, Jun. 2020.

[134] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[135] Z. Zhao, H. Fang, Z. Jin, and Q. Qiu, "Gisnet: Graph-based information sharing network for vehicle trajectory prediction," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–7.

[136] J. Ziegler *et al.*, "Making bertha drive-an autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, 2014, doi: 10.1109/MITS.2014.2306552.

[137] J. Gao *et al.*, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11525–11533.

[138] M. Liang *et al.*, "Learning lane graph representations for motion forecasting," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 541–556.

[139] H. Zhao *et al.*, "Tnt: Target-driven trajectory prediction," in *Proc. Conf. Robot Learn.*, 2020, pp. 895–904.

[140] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15303–15312.

[141] W. Zeng, M. Liang, R. Liao, and R. Urtasun, "Lanercnn: Distributed representations for graph-centric motion forecasting," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 532–539.

[142] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.

[143] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 6271–6280.

[144] L. Zhang, Q. She, and P. Guo, "Stochastic trajectory prediction with social graph network," 2019, *arXiv:1907.10233*.

[145] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 683–700.

[146] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Sys.*, vol. 27, 2014.

[147] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.

[148] B. Yang, G. Yan, P. Wang, C.-Y. Chan, X. Liu, and Y. Chen, "Tppo: A novel trajectory predictor with pseudo oracle," 2020.

[149] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6150–6156.

[150] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1349–1358.

[151] C. Hegde, S. Dash, and P. Agarwal, "Vehicle trajectory prediction using gan," in *Proc. Fourth Int. Conf. (IoT Social, Mobile, Analytics Cloud)*, 2020, pp. 502–507.

[152] T. Zhao *et al.*, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12126–12134.

[153] Y. Wang, S. Zhao, R. Zhang, X. Cheng, and L. Yang, "Multi-vehicle collaborative learning for trajectory prediction with spatio-temporal tensor fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 236–248, Jan. 2022.

[154] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. 2nd Int. Conf. Learn. Represent.*, 2014, *arXiv:1312.6114*.

[155] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 3483–3491, Jan. 2015.

[156] N. Rhinehart, K. M. Kitani, and P. Vernaza, "R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 772–788.

[157] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. Urtasun, "Implicit latent variable model for scene-consistent motion forecasting," in *Proc. Eur. Conf. Comput. Vis.*. Springer, 2020, pp. 624–641.

[158] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[159] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3 d classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[160] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[161] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[162] B. Hjaltason, "Predicting vehicle trajectories with inverse reinforcement learning," 2019.

[163] B. R. Kiran *et al.*, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[164] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Deep inverse reinforcement learning for behavior prediction in autonomous driving: Accurate forecasts of vehicle motion," *IEEE Signal Process. Mag.*, vol. 38, no. 1, pp. 87–96, Jan. 2020.

[165] R. Bellman, "A Markovian decision process," *J. Math. Mechanics*, vol. 6, no. 5, pp. 679–684, 1957.

[166] P. Wang, C.-Y. Chan, and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1379–1384.

[167] Y. Guan, S. E. Li, J. Duan, W. Wang, and B. Cheng, "Markov probabilistic decision making of self-driving cars in highway with random traffic flow: A simulation study," *J. Intell. Connected Veh.*, vol. 1, no. 2, pp. 77–84, 2018, doi: 10.1108/JICV-01-2018-0003.

[168] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 729–736.

[169] E. Klein, M. Geist, B. Piot, and O. Pietquin, "Inverse reinforcement learning through structured classification," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1–9.

[170] D. Silver, J. A. Bagnell, and A. Stentz, "Learning autonomous driving styles and maneuvers from expert demonstration," in *Experimental Robotics*. Heidelberg Germany: Springer, 2013, pp. 371–386.

[171] B. D. Ziebart *et al.*, "Maximum entropy inverse reinforcement learning," *Aaai*, vol. 8. pp. 1433–1438, 2008.

[172] M. Herman, V. Fischer, T. Gindele, and W. Burgard, "Inverse reinforcement learning of behavioral models for online-adapting navigation strategies," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 3215–3222.

[173] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to drive using inverse reinforcement learning and deep q-networks," 2016, *arXiv:1612.03653*.

[174] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2111–2117.

[175] D. S. González, O. Erkent, V. Romero-Cano, J. Dibangoye, and C. Laugier, "Modeling driver behavior from demonstrations in dynamic environments using spatiotemporal lattices," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 3384–3390.

[176] D. Xu *et al.*, "Learning from naturalistic driving data for human-like autonomous highway driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7341–7354, Dec. 2020.

[177] Z. Wu, L. Sun, W. Zhan, C. Yang, and M. Tomizuka, "Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5355–5362, Oct. 2020.

[178] N. Deo and M. M. Trivedi, "Trajectory forecasts in unknown environments conditioned on grid-based plans," 2020, *arXiv:2001.00735*.
[179] L. Xin *et al.*, "Accelerated inverse reinforcement learning with randomly pre-sampled policies for autonomous driving reward design," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 2757–2764.
[180] Y. Xu, T. Zhao, C. Baker, Y. Zhao, and Y. N. Wu, "Learning trajectory prediction with continuous inverse optimal control via langevin sampling of energy-based models," 2019, *arXiv:1904.05453*.
[181] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10239–10251, Aug. 2022.
[182] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 204–211.
[183] Y. Li, J. Song, and S. Ermon, "Infogail: Interpretable imitation learning from visual demonstrations," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3815–3825.
[184] R. Bhattacharyya *et al.*, "Modeling human driving behavior through generative adversarial imitation learning," 2020, *arXiv:2006.06412*.
[185] S. Choi, J. Kim, and H. Yeo, "Trajgail: Generating urban vehicle trajectories using generative adversarial imitation learning," *Transp. Res. Part C: Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103091.
[186] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robot. Auton. Syst.*, vol. 114, pp. 1–18, Apr. 2019.
[187] M. Wulfmeier, D. Rao, D. Z. Wang, P. Ondruska, and I. Posner, "Large-scale cost function learning for path planning using deep inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1073–1087, Sep. 2017.
[188] Q. Zou, H. Li, and R. Zhang, "Inverse reinforcement learning via neural network in driver behavior modeling," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1245–1250.
[189] Z. Zhu, N. Li, R. Sun, D. Xu, and H. Zhao, "Off-road autonomous vehicles traversability analysis and trajectory planning based on deep inverse reinforcement learning," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 971–977.
[190] C. Jung and D. H. Shim, "Incorporating multi-context into the traversability map for urban autonomous driving using deep inverse reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1662–1669, Apr. 2021.
[191] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 4565–4573, 2016.
[192] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," 2015, *arXiv:1507.04888*.
[193] M. Wulfmeier, D. Z. Wang, and I. Posner, "Watch this: Scalable cost-function learning for path planning in urban environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 2089–2095.
[194] "Traffic analysis tools," Accessed: Jan. 6, 2021. [Online]. Available: https://ops.fhwa.dot.gov/trafficanalysistools/index.htm
[195] H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11621–11631.
[196] S. Ettinger *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9710–9719.
[197] J. Houston *et al.*, "One thousand and one hours: Self-driving motion prediction dataset," in *Proc. Conf. Robot Learn.*, vol. 155, 2020, pp. 409–418.
[198] B. Wilson *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, 2021.
[199] W. Zhan *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," 2019, *arXiv:1910.03088*.
[200] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2118–2125.
[201] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
[202] K. Saleh, M. Hossny, and S. Nahavandi, "Long-term recurrent predictive model for intent prediction of pedestrians via inverse reinforcement learning," in *Proc. Digit. Image Comput.: Techn. Appl.*, 2018, pp. 1–8.
[203] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Neighbourhood context embeddings in deep inverse reinforcement learning for predicting pedestrian motion over long time horizons," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019.
[204] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr, "A combined model-and learning-based framework for interaction-aware maneuver prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1538–1550, Jun. 2016.
[205] S. Wang, P. Zhao, B. Yu, W. Huang, and H. Liang, "Vehicle trajectory prediction by knowledge-driven lstm network in urban environments," *J. Adv. Transp.*, vol. 2020, 2020.
[206] "Inter-tnt (jointly vectornet-tnt-interaction) evaluator," Accessed: Mar. 26, 2022. [Online]. Available: https://github.com/ApolloAuto/apollo/blob/master/docs/technical_documents/jointly_prediction_planning_evaluator.md

Yanjun Huang received the Ph.D. degree in 2016 from the Department of MME, University of Waterloo, Waterloo, ON, Canada. He is currently a Professor with the School of Automotive studies, Tongji University, Shanghai, China. He has authored or coauthored several books, more than 60 papers in journals and conference. His research interests mainly include improving vehicle performance in terms of safety, energy-saving, and intelligence by using advanced control and learning methods.

He was the recipient of the IEEE VTS 2019 Best Land Transportation Paper Award, the 2018 Best paper of Automotive Innovation. He is serving as AE or EBM of IET Intelligent Transport System, SAE *International Journal of Commercial Vehicles International Journal of Vehicle Autonomous Systems.*

Jiatong Du received the B.S. degree in 2020 from the School of Automotive Studies, Tongji University, Shanghai, China, where he is currently working toward the Ph.D. degree with the School of Automotive Studies. His research interests include machine learning, deep learning, trajectory prediction, trajectory planning, intelligent transportation systems, and autonomous vehicle.

Ziru Yang received the B.S. degree in 2021 from the School of Automotive Studies, Tongji University, Shanghai, China, where she is currently working toward the M.S. degree with the School of Automotive Studies. Her research interests include trajectory prediction and planning, cooperative vehicle infrastructure system, and autonomous vehicle.

Zewei Zhou received the B.S. degree in vehicle engineering from Chongqing University, Chongqing, China, in 2020. He is currently working toward the M.S. degree with the School of Automotive Studies, Tongji University, Shanghai, China. His research interests include machine learning, trajectory prediction, trajectory planning, intelligent transportation systems, and autonomous vehicle.

Lin Zhang received the Ph.D. degree in automotive engineering from Jilin University, Changchun, China, in 2019. He is currently a Postdoctoral Fellow with the School of Automotive Studies, Tongji University, Shanghai, China. His research interests include vehicle control in terms of safety, saving energy, and intelligence, including vehicle dynamics and control, HEV control, and trajectory planning.

Hong Chen (Senior Member, IEEE) received the B.S. and M.S. degrees in process control from Zhejiang University, Hangzhou, China, in 1983 and 1986, respectively, and the Ph.D. degree in system dynamics and control engineering from the University of Stuttgart, Stuttgart, Germany, in 1997. In 1986, she joined the Jilin University of Technology, Changchun, China. From 1993 to 1997, she was a Wissenschaftlicher Mitarbeiter with the Institut fuer Systemdynamik und Regelungstechnik, University of Stuttgart. Since 1999, she has been a Professor with Jilin University and hereafter a Tang Aoqing Professor. Recently, she joined Tongji University as a Distinguished Professor. Her research interests include model predictive control, nonlinear control and applications in mechatronic systems, automotive systems.