Vehicular Communications 46 (2024) 100733

Contents lists available at ScienceDirect

Vehicular Communications

journal homepage: www.elsevier.com/locate/vehcom

Machine learning for autonomous vehicle's trajectory prediction: A comprehensive survey, challenges, and future research directions

Vibha Bharilya, Neetesh Kumar *

Department of Computer Science and Engineering, Indian Institute of Technology, Roorkee, 247667, India

ARTICLE INFO

Keywords:
Autonomous vehicle
Trajectory prediction
Machine learning
Deep learning
Reinforcement learning

ABSTRACT

The significant contribution of human errors, accounting for approximately 94% (with a margin of ±2.2%), to road crashes leading to casualties, vehicle damages, and safety concerns necessitates the exploration of alternative approaches. Autonomous Vehicles (AVs) have emerged as a promising solution by replacing human drivers with advanced computer-aided decision-making systems. However, for AVs to effectively navigate the road, they must possess the capability to predict the future behaviour of nearby traffic participants, similar to the predictive driving abilities of human drivers. Building upon existing literature is crucial to advance the field and develop a comprehensive understanding of trajectory prediction methods in the context of automated driving. To address this need, we have undertaken a comprehensive review that focuses on trajectory prediction methods for AVs, with a particular emphasis on machine learning techniques including deep learning and reinforcement learning-based approaches. We have extensively examined over two hundred studies related to trajectory prediction in the context of AVs. The paper begins with an introduction to the general problem of predicting vehicle trajectories and provides an overview of the key concepts and terminology used throughout. After providing a brief overview of conventional methods, this review conducts a comprehensive evaluation of several deep learning-based techniques. Each method is summarized briefly, accompanied by a detailed analysis of its strengths and weaknesses. The discussion further extends to reinforcement learning-based methods. This article also examines the various datasets and evaluation metrics that are commonly used in trajectory prediction tasks. Encouraging an unbiased and objective discussion, we compare two major learning processes, considering specific functional features. By identifying challenges in the existing literature and outlining potential research directions, this review significantly contributes to the advancement of knowledge in the domain of AV trajectory prediction. Its primary objective is to streamline current research efforts and offer a futuristic perspective, ultimately benefiting future developments in the field.

1. Introduction

Annually, approximately 1.35 million deaths occur due to road crashes, with 1,140 reported deaths in 2018 according to the Australian Automobile Association (AAA) [1]. There were 1,194 fatal car accidents in 2022 in Australia. This is an increase of 5.8% from 2021. National fatalities have stayed basically flat during the past ten years [2]. In the United States, the NHTSA's investigation reveals that around 94% of severe road crashes can be attributed to driver errors [4]. Further, human error is consistently identified as a major factor in road crashes, emphasizing the need to address this preventable distress. To assist human drivers in avoiding critical situations, advanced motorized vehicles employ Advanced Driver Assistance Systems (ADAS), which have rapidly evolved since their inception in the 1950s. Researchers are actively exploring the efficiency of ADAS in warning drivers and preventing crashes. The rapid technological progress, including the use of high-end sensors, powerful machine learning techniques, and innovations from companies like Google and Tesla Motors, has significantly impacted the automation industry. Automotive and tech companies have demonstrated the feasibility of Automated Driving Systems (ADS) through successful test fleet operations. The Society of Automotive Engineers (SAE) classifies ADS into six levels of vehicle automation [94], with a focus on full automated operation.

Autonomous Vehicles (AVs) are expected to play a significant role in reducing crashes and enhancing road safety in the foreseeable future. The rapid development of perception, planning, and control systems

* Corresponding author.
E-mail addresses: vibha_b@cs.iitr.ac.in (V. Bharilya), neetesh@cs.iitr.ac.in (N. Kumar).

https://doi.org/10.1016/j.vehcom.2024.100733
Received 1 July 2023; Received in revised form 26 December 2023; Accepted 23 January 2024
Available online 29 January 2024

for AVs in recent years is noteworthy. However, the production of AVs in large quantities will not be feasible until their safety is fully established. One of the critical technologies in AVs is the ability to forecast the future states of the surrounding environment in real time, as human drivers can. This capability will further enhance safety measures. Before beginning a new driving operation, such as acceleration or a lane change, a human driver typically scans the surrounding traffic to predict how it will behave in the future. Future trajectories can be used to model future traffic participant states, which can then be used to construct decision-making or planning algorithms as well as to foresee potential dangers. However, accurately predicting future traffic participant trajectories is attracting a lot of attention and is quickly becoming one of the key points to improving the safety of autonomous driving. This is because of the variety of manoeuvres that traffic participants make, the complex interactions between traffic participants and environments, the uncertainty of sensory information, the computation burdens, and the computing time requirements of AVs.

1.1. Motivation

Numerous techniques have been developed for the trajectory prediction, and several scholars have pursued this area of research. Some of the review papers have covered various trajectory prediction techniques, in the same line, Lefèvre et al. [112] provided an analysis of motion prediction and risk assessment techniques used for AVs before 2014. Mohammad et al. [165] discussed strategies for behaviour prediction at crossings based on drivers' actions. Further, Mozaffari et al. [140] offered a review of deep learning-based approaches focused on vehicle behaviour analysis. Leon et al. [114] and Liu et al. [124] wrote reviews on trajectory prediction for AVs, where, Leon et al. covered deep learning and stochastic methods, and Liu et al. focused solely on deep learning methods. Karle et al. [100] offered three distinct prediction models as a classification of these models and compared them based on the underlying study methodology. Gomes et al. [65] reviewed the literature on Intention-Aware and Interaction-Aware trajectory prediction for autonomous vehicles and examined how manoeuvre goals and their interaction with other manoeuvres affect the performance of trajectory prediction techniques. In Ghorai et al. [62], a survey covered the identification and monitoring of dynamic agents and objects encountered by an autonomous ego vehicle. The main topics of the review were delved into 2D and 3D dynamic object identification techniques based on DL employed in AV research. Huang et al. [92] thoroughly examined trajectory prediction techniques for AVs put forth over the last two decades, excluding vision-based techniques. Recently, Benrachou et al. [18] encompassed research on both data-driven and model-based algorithms, which aim to forecast the movement of surrounding traffic. Table 1 provides a summary of the related state-art-of surveys, along with the different categorisations approaches, and contributions.

Motion prediction involves anticipating the behaviour, manoeuvres, or trajectory of an object, depending on the desired level of abstraction. The term "behaviour" encompasses general actions and their execution style, such as "following the road and maintaining a safe distance." On the other hand, "manoeuvres" refer to discrete actions that an object can perform without requiring a detailed specification, such as "turning right." Trajectories, on the other hand, provide the most detailed type of prediction by describing an object's position over discrete time steps [100]. Previous surveys have predominantly emphasized motion prediction and behaviour prediction in the realm of Autonomous Vehicles (AVs). Further, other state-of-the-art surveys are mixed of trajectory prediction for vehicles and pedestrians. A comprehensive and dedicated review on autonomous vehicles trajectory prediction accounting machine learning methods is remained relatively unexplored. Furthermore, several advancements in the domain of trajectory prediction, in recent years, including computer vision-based methods, reinforcement learning etc., have not been addressed in the existing surveys which are also needed to be explored. Consequently, there exists substantial potential

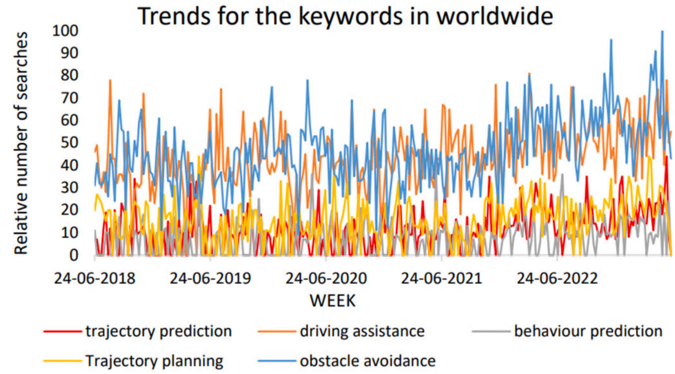

Fig. 1. Google trends for specific keywords.

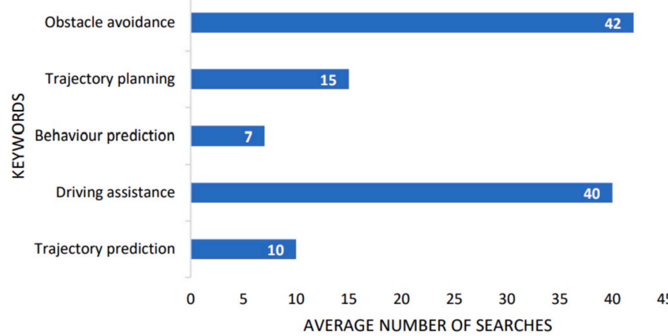

Fig. 2. Average count for the specific keywords related to trajectory prediction.

for further exploration and investigation within this domain. Thus, the motivation behind writing this survey paper is to actively contribute to the research in the trajectory prediction field specifically for AVs.

1.2. Google trends

In recent years, Autonomous Driving (AD) has become increasingly popular in the automotive industry. Prominent automobile manufacturers, including Tesla, General Motors, and BMW, have made significant investments and focused on trajectory prediction and related technologies for Autonomous Vehicles (AVs) for the development of AD technology. The worldwide search trends for keywords related to AD, such as trajectory prediction, driving assistance, behaviour prediction, trajectory planning, and obstacle avoidance, are illustrated in Fig. 1 and Fig. 2. In Fig. 1, the comparison of related keywords demonstrates changes over the same time period. It shows that driving assistance and obstacle avoidance generate similar levels of excitement worldwide. However, within the field of AD, there have been recent advancements in keywords such as trajectory prediction, trajectory planning, and behaviour prediction. Notably, trajectory prediction focuses on a more specific domain within autonomous driving. In Fig. 2, the average number of searches related to the keyword worldwide is depicted. Notably, driving assistance and obstacle avoidance keywords receive a higher number of searches compared to other terms like trajectory prediction, trajectory planning, and behaviour prediction. Trajectory prediction is currently evolving in the field of AD, indicating increasing interest and development in this area.

1.3. Key contributions

This comprehensive survey on the state-of-the-art machine learning-based trajectory prediction methods for Autonomous Vehicles (AVs) provides a taxonomy of the different approaches, as shown in Fig. 3, including conventional methods, deep learning-based methods, and reinforcement learning-based methods, and discusses the advantages and

Table 1
Summary of related state-art-of the surveys: Related work, the title of the work, categorisation of each work and their contribution.

S.No.	Ref. & Year	Topics Discussed	Categorisation	Contributions	Limitation
1.	[140] & 2020	Deep Learning-Based Vehicle Behaviour Prediction for Autonomous Driving Applications: A Review	Classification based on three criteria: input representation, output type, and prediction method.	An assessment of contemporary deep learning techniques for predicting vehicle behaviour.	1. The classification approach omits reinforcement learning and conventional methods. 2. This survey lacks a thorough description of the datasets utilized.
2.	[114] & 2021	A Review of Tracking and Trajectory Prediction Methods for Autonomous Driving	Categorized based on its primary prediction approach, including neural networks, stochastic methods, and hybrid techniques.	This article offers a study of tracking and trajectory prediction approaches that only include deep learning and stochastic methodology methods.	1. A thorough examination of the attributes of trajectory prediction methods is lacking. 2. The survey did not incorporate information about the datasets and evaluation metrics.
3.	[124] & 2021	A Survey on Deep-Learning Approaches for Vehicle Trajectory Predictions in Autonomous Driving	Grouped according to the manner in which data is represented, learning techniques utilized, and objective functions employed.	Examine and characterize current deep learning-based trajectory forecasting techniques in this work.	This survey only covers methods in deep learning.
4.	[100] & 2022	Scenario Understanding and Motion Prediction for Autonomous Vehicles—Review and Comparison	A categorisation of prediction models are presented, which includes physics-based, pattern-based, and planning-based approaches.	Examine and compared the three specific prediction methods and Demonstrate a trade-off between holism and interpretability in contemporary approaches.	The survey neglected to include information concerning the datasets and evaluation metrics.
5.	[65] & 2022	A Review on Intention-aware and Interaction-aware Trajectory Prediction for Autonomous Vehicles	The current classification of prediction methods is framed within the context of intention-aware and interaction-aware trajectory prediction techniques.	Examine and characterize the physics-based methods and ML techniques.	This survey lacks a well-defined taxonomy for prediction models.
6.	[62] & 2022	State Estimation and Motion Prediction of Vehicles and Vulnerable Road Users for Cooperative Autonomous Driving: A Survey	A categorisation of prediction models are presented into three classes: physics-based, manoeuvre-based, and interaction aware motion models.	The major focus of this review is on perception sensors and navigation.	1. Reinforcement learning-based methods are not covered in this survey. 2. The survey does not encompass the evaluation metric and datasets for vehicle trajectory prediction.
7.	[92] & 2022	A Survey on Trajectory-Prediction Methods for Autonomous Driving	Classification based on physics-based methods, the classic machine learning-based methods, the deep learning-based methods, and reinforcement learning-based methods	Choose both heuristic and contemporary trajectory prediction techniques for a specific timeframe and provide a comparative summary.	This survey does not encompass computer vision-based prediction methods.
8.	[18] & 2022	Use of Social Interaction and Intention to Improve Motion Prediction Within Automated Vehicle Framework: A Review	Motion prediction solutions should be categorized into four principal strategies: (1) Prediction methods (intention-aware or interaction-aware), (2) Classes (model-based or data-driven), (3) Algorithms (which type of model), and (4) Datasets (classified according to the point of view: top-down view data or vehicle-view data).	Cover the machine learning techniques and the primary emphasis of this paper has been on approaches that are intention-aware or incorporate interaction-awareness.	This survey does not encompass reinforcement learning-based methods as well as traditional approaches.
9.	**This survey**	Machine Learning for Autonomous Vehicle's Trajectory Prediction: A comprehensive survey, Challenges, and Future Research Directions	The categorisation involves the utilisation of conventional methods, computer vision-based methods, cutting-edge deep learning techniques, and reinforcement learning-based methods.	This assessment presents a concise overview of traditional methods, computer vision-based approaches, and conducts a comprehensive assessment of prevalent deep learning and reinforcement learning-based techniques employed in trajectory prediction for autonomous vehicles. Additionally, it includes a discussion of the advantages and disadvantages associated with these methods.	-

limitations of each method. This study focuses on vehicle trajectory prediction algorithms, as other traffic participants, such as adjacent vehicles, directly affect the ego vehicle. In the end, the paper highlights the challenges and future research directions in this field. The significant contributions of this survey are enlisted as follows:

1. This survey offers an empirical study on autonomous vehicles trajectory prediction methods and extensively focuses on machine learning-based methods. For the better understanding, an overview on AV's trajectory prediction problem, related terminology, and conventional methods are also briefly provided.
2. A concise assessment of conventional methods such as Physics-based methods, Sampling methods, and Probabilistic models in trajectory prediction is presented, along with a discussion of their advantages and disadvantages.
3. A comprehensive evaluation is provided for the prevalent deep learning and reinforcement learning based-methods used in trajectory prediction for autonomous vehicles.

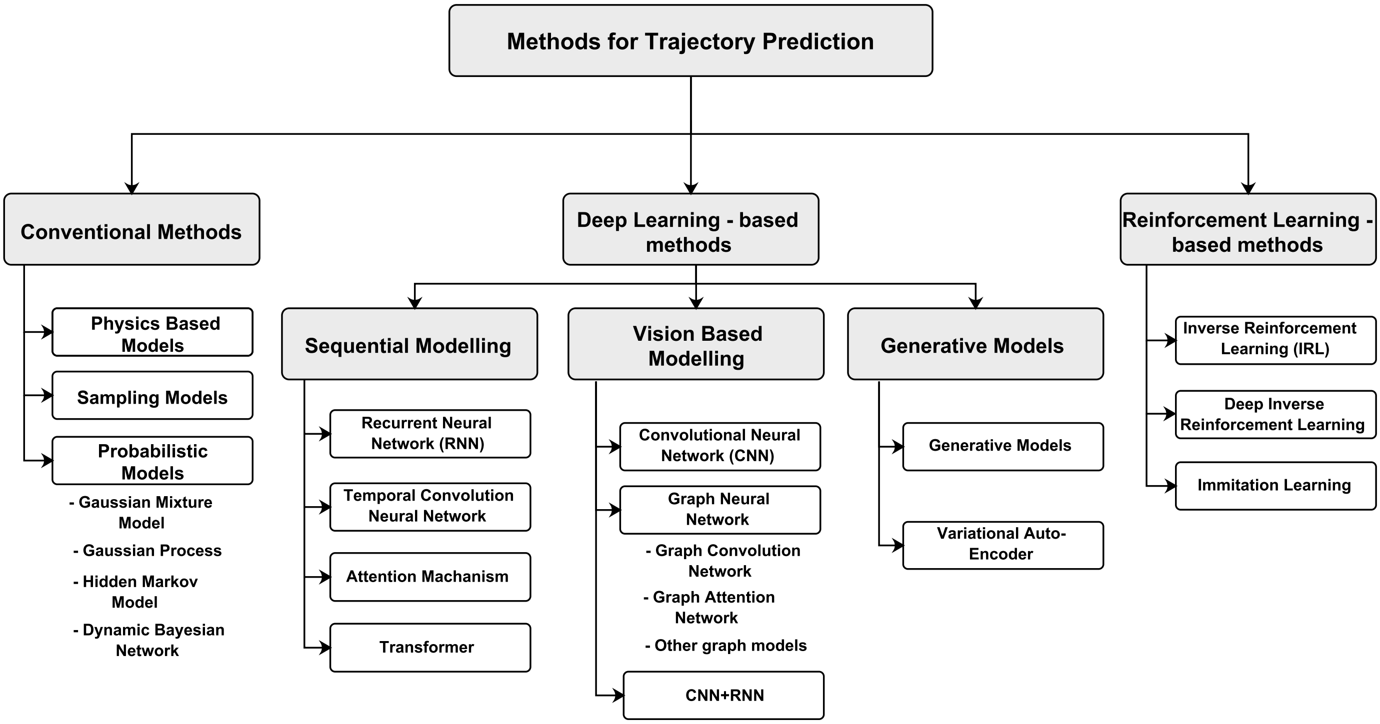

Fig. 3. Categorisation of methods for trajectory prediction task.

4. An analytical summary is provided for the metrics and datasets used to evaluate the performance of trajectory prediction methods.
5. A comparison of the methods is conducted, analyzing the strengths and weaknesses of each approach. Furthermore, challenges and potential research avenues are identified.

1.4. Review strategies

In crafting a survey paper, the initial steps involve an extensive literature search across reputable databases such as IEEE Xplore, Elsevier, Conferences, and Google Scholar. This search is driven by carefully selected keywords related to autonomous driving, trajectory prediction, and machine learning. The screening process entails reviewing titles and abstracts to pinpoint papers that align with the survey's focal points. Full-text reviews follow to comprehensively assess the content and relevance of the selected papers. Once the papers are collected, they are categorized based on various parameters such as prediction methods, datasets used, evaluation metrics, and advantages or limitations addresses. The review also delves into identifying gaps in current research and proposes potential directions for future exploration and improvement. Throughout this process, proper citation and referencing are maintained according to the chosen citation style, ensuring academic integrity in the survey paper.

1.5. Paper organisation

The road map of the survey has been presented in pictorial form in Fig. 4. There are nine sections in this paper. Section 2 presents a generic problem formulation, provides definitions of the terminologies used, and the methods are categorized based on various criteria. In Sections 3, 4, and 5 of the paper, comprehensive reviews on conventional-based methods, deep learning-based methods, and reinforcement learning-based methods are conducted respectively. Section 6 discusses the commonly used evaluation metrics and datasets. Section 7 discusses the performance of different methods, and Section 8 highlights the current challenges in the literature and potential new research directions. The key concluding remarks are given in section 9.

2. Problem description

In the context of Automated Driving (AD), accurately predicting the trajectories of other road users poses a significant challenge for AV's software. It requires a comprehensive understanding of the spatio-temporal dynamics of the environment, including the past states of observable road users and their interaction patterns, irrespective of their quantity and types. Trajectory prediction involves two main steps. First, it is essential to track and gather relevant information about neighbouring road users to obtain precise and reliable trajectories. Second, based on the acquired knowledge, future trajectories of these neighbouring road users need to be predicted. To accomplish these tasks, the AV's software must have access to mapping data encompassing the road scene and the surrounding area (referred to as road context). This includes information such as road and crosswalk locations, lane directions, and other relevant map-related details. Additionally, the software needs to identify and monitor Surrounding Vehicles (SVs) as well as Target Vehicles (TVs) for accurate trajectory prediction. To tackle the inherent ambiguity of the problem, we approach vehicle trajectory prediction as a probabilistic task. We define the future trajectories of TVs as the sequence of their future states, denoted as Y_{TV_s}:

$$Y_{TV_s} = \{e_j^t, e_j^{t+1}, e_j^{t+2}, \dots, e_j^{t+f}\}_{j=1}^N \tag{1}$$

Here, N represents the number of TVs, f is the size of the prediction window, and e_j^t denotes the state of vehicle j at time step t. The problem is formulated by computing the posterior distribution P(Y_{TV_s}|C), where $C = X \cup I$ represents the available information to the ego vehicle. The historical states, captured in X, encompass the observations of N traffic participants up to time step t-1:

$$X = \{e_j^0, e_j^1, e_j^2, \dots, e_j^{t-1}\}_{j=1}^N \tag{2}$$

These historical states typically include attributes such as position, velocity, acceleration, orientation, etc. Additionally, I denote optional environmental information that can be considered or omitted based on availability. In this formulation, the goal is to estimate the future trajectories Y_{TV_s} of the traffic participants given the available information C.

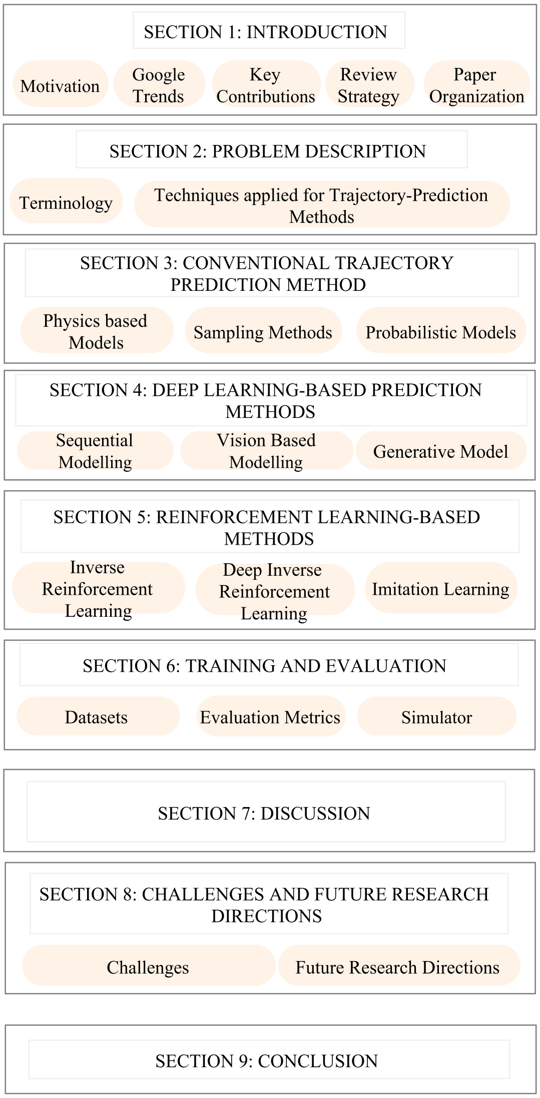

Fig. 4. Road Map of the Paper.

The posterior distribution P(Y_{TV_s}|C) represents the probability distribution of the future trajectories conditioned on the available information. To manage computational complexity, the prediction of each TV can be performed independently. At each stage, one vehicle is selected as the target TV, and its trajectory distribution, P(Y_{TV}|C), is computed:

$$Y_{TV_s} = \{e_j^t, e_j^{t+1}, e_j^{t+2}, ..., e_j^{t+f}\}_{j=1}^{N} \tag{3}$$

Here, T represents the chosen TV, and the trajectory prediction for that specific vehicle is determined.

2.1. Terminology

In the field of trajectory prediction, several terminologies are commonly used to describe different aspects of the prediction process. Here are some key terminologies.

1. **Trajectory:** A trajectory refers to the path or motion of an object or entity over time. It represents the series of positions or states that an object traverses.
2. **Manoeuvre:** The term “manoeuvre” refers to the specific actions or movements performed by a vehicle or object as it navigates through its environment. Manoeuvres can include various actions, such as lane changes, turns, merges, accelerations, decelerations, and stops.

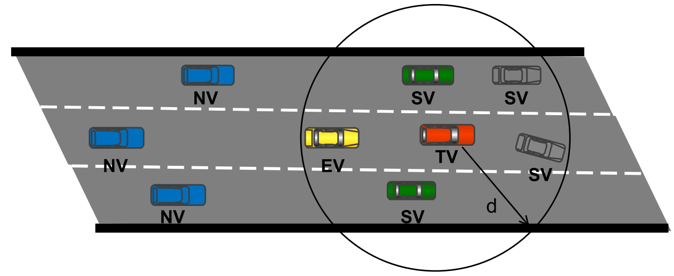

Fig. 5. This illustration showcases the terminology used and the restricted view of the onboard sensors in EVs. In order to categorize vehicles as either SVs or NVs, a criterion is used where vehicles within a certain distance threshold (d) of the TV are considered to have an impact on its behaviour. The two SVs that are not observable by the EV are depicted in grey. However, limited observability can lead to inaccurate predictions, such as when the preceding vehicle of the TV changes lanes, which is not visible to the EV, allowing the TV to accelerate [140].

3. The vehicles whose trajectory we are interested in forecasting are called **Target Vehicles (TVs)**.
4. **Ego Vehicle (EV)** is an autonomous vehicle that monitors its surroundings to forecast TV trajectory.
5. The prediction model examines **Surrounding Vehicles (SVs)** since they may have an impact on how TV will behave in the future. SVs may be chosen using a variety of criteria depending on the modelling assumptions used in the study.
6. The remaining vehicles in the driving environment that are deemed to have no bearing on the behaviour of the TV are known as **Non-Effective Vehicles (NVs)**.
7. **Unimodal Trajectory** - Generate the single trajectory of single or multiple traffic participants in the given scenes.
8. **Multimodal Trajectory** - Generate the multiple trajectories of single or multiple traffic participants in the given scenes.

The proposed terminology is illustrated in Fig. 5, through a driving scenario. The vehicles in the scenario are divided into SVs and NVs using a distance-based criterion as an example.

2.2. Techniques applied for trajectory-prediction methods

Trajectory prediction methods in autonomous driving can be broadly classified into the following categories.

2.2.1. Conventional methods

Conventional methods for trajectory prediction refer to traditional approaches that have been commonly used to forecast the future trajectories of road users in Autonomous Driving (AD). These methods typically rely on well-established mathematical and statistical techniques to make predictions based on historical data and predefined models. Some of the commonly used conventional methods are:

1. *Physics-based Models*: These methods rely on the laws of physics and kinematics principles to predict the future trajectory of a vehicle. They consider factors such as current position, velocity, acceleration, and road constraints to estimate the future path [201].
2. *Kinematic models*: These models assume that the motion of objects can be described by simple mathematical equations, such as constant velocity or constant acceleration models. They estimate future positions based on the object's current state and its assumed motion dynamics [17].
3. *Kalman filters*: Kalman filters are widely used for tracking and prediction tasks. They combine measurements from sensors with predictions from a mathematical model to estimate the current state of an object and make predictions about its future trajectory [113].

4. *Markov models*: Markov models capture the probabilistic dependencies between successive states of an object. They use historical data to estimate transition probabilities and make predictions based on the most likely sequence of states [188].
5. *Probabilistic Models*: Probabilistic approaches consider uncertainty in trajectory prediction by representing the future trajectories as probability distributions. These models leverage statistical techniques to estimate the most likely trajectory and provide a measure of confidence [223].
6. *Bayesian Filters*: Bayesian filters, such as Kalman filters and particle filters, are widely used for trajectory prediction. These filters combine measurements from sensors with a dynamic model to estimate the future trajectory of a vehicle. They can handle noisy sensor data and provide real-time predictions [116].

Conventional methods for trajectory prediction are often computationally efficient and relatively easy to implement. However, they may have limitations in handling complex scenarios with intricate interactions and uncertainties. As a result, there has been a growing interest in exploring more advanced machine learning-based approaches, such as deep learning and reinforcement learning, to improve the accuracy and robustness of trajectory predictions.

2.2.2. Deep learning-based methods

Deep learning-based methods have gained significant attention in recent years for trajectory prediction in Autonomous Vehicles (AVs). These methods leverage the power of artificial neural networks to learn complex patterns and relationships from large amounts of data. Here are some common deep learning-based approaches for trajectory prediction:

1. *Recurrent Neural Networks (RNNs)*: RNNs are widely used in trajectory prediction due to their ability to model sequential data. Models such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) can capture temporal dependencies and predict future trajectories based on past observations [68] [169].
2. *Convolutional Neural Networks (CNNs)*: CNNs are primarily used for image processing tasks, but they can also be applied to trajectory prediction by treating trajectory data as image-like representations. CNNs can extract spatial features from trajectory data and learn to predict future trajectories based on these features [43].
3. *Generative Adversarial Networks (GANs)*: GANs consist of a generator network and a discriminator network. They can be employed for trajectory prediction by training the generator to generate realistic future trajectories and the discriminator to differentiate between real and generated trajectories. GANs can capture the distribution of training data and generate diverse and plausible trajectory predictions [230] [28].
4. *Variational Autoencoders (VAEs):* VAEs are generative models that learn a latent representation of the input data. They can be used for trajectory prediction by learning the latent space representation of past trajectories and generating future trajectories conditioned on this latent representation. VAEs enable the generation of diverse and probabilistic trajectory predictions [20].
5. *Transformer Models:* Transformer models, originally introduced for natural language processing tasks, have also shown promise in trajectory prediction. These models can capture long-range dependencies and interactions between different agents in the scene. By attending to relevant spatial and temporal information, transformer models can generate accurate trajectory predictions [153].

Deep learning-based methods have demonstrated improved performance in capturing complex patterns, handling diverse scenarios, and generating more accurate trajectory predictions compared to conventional approaches. However, they require large amounts of labelled training data and computational resources for training and inference.

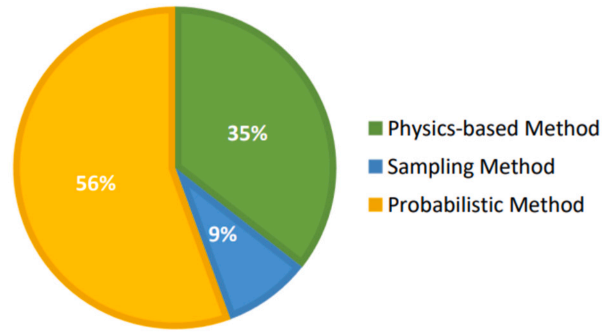

Fig. 6. Participation of Research articles in trajectory prediction task using conventional approaches.

Additionally, the interpretability of the learned models can be a challenge, making it important to validate the predictions and understand the model's limitations in real-world scenarios.

2.2.3. Reinforcement learning-based methods

Reinforcement Learning (RL) methods have been explored for trajectory prediction in Autonomous Driving (AD), offering a unique approach to learn optimal policies for predicting future trajectories. While RL is traditionally associated with decision-making and control, it can also be utilized in the context of trajectory prediction. Here are some RL methods used for trajectory prediction:

1. *Inverse Reinforcement learning (IRL):* The key idea behind IRL is to observe and analyze expert demonstrations, typically provided by human drivers, and then infer the underlying reward function that motivates their actions. This inferred reward function can be used to predict future trajectories that align with the observed expert behaviour [235].
2. *Deep Inverse Reinforcement Learning (Deep IRL):* Deep IRL is an extension of Inverse Reinforcement Learning (IRL) that combines deep neural networks with the IRL framework to predict trajectories in AD. Deep IRL aims to infer the underlying reward function from expert demonstrations using deep learning techniques, allowing for more complex and high-dimensional representations of the reward function [215] [157].
3. *Imitation learning (IL):* IL for trajectory prediction enables autonomous systems to mimic the behaviour of human drivers and generate trajectories that align with expert demonstrations. It leverages the knowledge and expertise of human drivers to make more human-like predictions and navigate the environment in a manner that is similar to how humans would drive [85].

By applying RL methods to trajectory prediction, models can learn from data and interactions with the environment to make accurate predictions about future trajectories. However, it is important to consider the trade-off between the complexity of RL algorithms and the availability of training data, as well as the challenges of generalisation to various driving scenarios and uncertainties in the real-world environment.

3. Conventional trajectory prediction method

This section classifies prediction methods into three dominant classes, Physics-based Models, Sampling Methods, and Probabilistic models, and Table 2 presents a brief overview of the Conventional methods for trajectory prediction with their limitations and advantages. In Fig. 6, several conventional methods and their involvement in addressing the trajectory prediction task in Autonomous Vehicles (AVs) are depicted. The analysis of the papers reveals that 56% of the papers focus on probabilistic methods, 35% of the papers focus on sampling methods, and the remaining 9% of the papers are dedicated to sampling methods in this survey.

Table 2
Summary of the Conventional Trajectory prediction methods.

Based-on	Sub-category	Limitations	Advantageous
Physics-Based trajectory prediction	Dynamic models [154,21,122,90,146,99]	The complexity of dynamics models can be very large and they can have many intrinsic properties.	Instead of prediction, dynamic models are employed for motion control.
	Kinematics model [9,161,150,129,16,17]	Only capable of predicting short-term trajectory of traffic agents.	Kinematics models are simple structure as compared to dynamic models.
	Kalman Filtering methods [222], [113]	It presupposes that the equations for prediction models are linear, which is unrealistic in many real-world circumstances.	The noise of the current state of the vehicle can be handled.
	Monte Carlo methods [144], [187]	Computationally inefficient when considering the large number of parameters.	This strategy is straightforward and understandable.
Sampling Methods	[83], [177], [195], [7]	These models cannot be easily generalised to various scenes because they were only trained for certain ones.	They can withstand noise and uncertainty better.
Probabilistic Models	Gaussian Mixture Models [89], [10], [194], [42], [223]	They might not be appropriate for modelling distributions with more complexity.	Several different probability distributions can be modelled using Gaussian Mixture Models.
	Gaussian Process [111], [179], [72], [181], [78]	Before incorporating the data into the prediction models, make certain assumptions about it.	They are able to accurately assess their own uncertainty.
	Hidden Markov Models [152], [41], [188]	They do not take into account how interaction-related aspects affect the process of prediction.	HMM techniques have been quite successful at predicting driving manoeuvres.
	Dynamic Bayesian Network [141,64,160,13,76,116]	DBN struggles with error from manoeuvre recognition through trajectory generation.	DBN simulates the impact of both vehicle states and traffic participant interactions.

Table 3
Summary of Physics-based models with design and implementation parameters.

Work	Input	Model	Case Study	Evaluation metrics	Predict (second(s))	Data
[222]	Speed, Heading	Kalman filter	Straight and corner scenario; Curve scenario	Relative distance, Relative speed, Time to collision	3 s	Simulator
[113]	Coordinates	Kalman Filter	Cruise control; Lane change	RMSE	1 s, 2 s, 3 s, 4 s	NGSIM I-80, Simulator
[144]	Coordinates, Map	Random forests, Particle filter, Monte Carlo simulation	Multi-class classification scenario	Precision, Recall, F1 score	-	NGSIM I-80, Numerical simulation: training 80 datasets
[187]	Coordinates, Velocity, Map	Model predictive control+montecarlo simulation	Lane change; Turning at the intersection	Crash probability	7 s, 8 s, 10 s, 12 s	Simulation: 1000000 samples

"- This information is not available/applicable for work".

3.1. Physics based models

The first class of suggested physics-based prediction models uses classical mechanics' motion equations as a foundation for modelling the target object's future motion. Either dynamic or kinematic models can be used to describe the physical behaviour. A dynamics model considers the lateral and longitudinal tire forces causing the motion, but a basic dynamics model is typically chosen to balance predictive accuracy and computational effort. In contrast, kinematics models are more commonly used due to their simple form, and the Kalman Filtering (KF) techniques can handle disturbances, such as uncertainty or noise, in the vehicle's current condition. For instance, Zhang et al. [222] proposed a vehicle-to-vehicle communication and KF-based approach to enable a host vehicle to predict the trajectories of remote vehicles and avoid obstacles. Lefkopoulos et al. [113] also introduced the Interacting Multiple Model Kalman Filter (IMM-KF), a new technique that incorporates interaction-related parameters for more accurate trajectory prediction using a physics-based model over a few seconds. The Monte Carlo approach can be used to roughly simulate the state distribution by applying a physics model to a sample of input variables at random, generating potential future trajectories. This method can be used to predict traffic participant trajectories from either a fully known or an unknown state evaluated by a filtering mechanism. Okamoto et al. [144] use the Monte Carlo approach in their manoeuvre-based model to predict future trajectories based on the recognized manoeuvre. Similarly, Wang et al. [187] use the Monte Carlo approach to predict trajectories and then use Model Predictive Control (MPC) to refine the reference trajectories. In summary, Physics-based Models are characterized by their excellent explainability, robust performance, and high accuracy, particularly for short-term prediction tasks. These models are well-suited for safety assessment purposes, typically focusing on predictions within a time horizon of no more than 1 second. However, they may have limitations in capturing complex manoeuvres, often relying on simplified assumptions and limited adaptability in unknown or dynamic environments. A concise overview of these models, including design and implementation parameters such as input features, model architecture, case studies, evaluation metrics, prediction horizon in (seconds), and dataset details, is presented in Table 3.

3.2. Sampling methods

These techniques involve sampling the future states of traffic participants. Instead of predicting a single trajectory, these approaches generate a distribution of possible vehicle states, which makes them more robust to noise and uncertainty. There are two main types of sampling: generating multiple trajectory segments or particle states. In their study, Houenou et al. [83] combined a manoeuvre-based approach with a model-based approach assuming Constant Yaw Rate and Acceleration (CYRA) to develop a trajectory prediction method. They identified the manoeuvre and selected the best trajectory from a set generated by minimizing a cost function. Meanwhile, Tran and Firl [177] utilized Monte Carlo Simulation (MCS) to predict multimodal trajectories

Table 4
Summary of Sampling-based models with design and implementation parameters.

Work	Input	Model	Case Study	Evaluation metrics	Predict (second(s))	Data
[83]	Coordinates, Yaw angle, Velocity, Acceleration, Yaw rate	Constant Yaw Rate and Acceleration, Kalman filtering	Lane-change or Overtaking	Mean error	[0 s, 1 s[, [1 s, 2 s[, [2 s, 3 s[and [3 s, 4 s[	Self-collected data: Ring roads of Beijing, China
[177]	Coordinates, Velocity	Kalman filters, Constant velocity model, Gaussian process regression, Monte Carlo method	Intersection	-	10 frames	Two different urban intersections in Karlsruhe, Germany
[195]	Position, Velocity	Monte Carlo simulation, Gaussian processes	Closing lane scenario, Free flow with sensor limitations	Error ellipse	-	Simulation
[7]	Velocity	Ordinary differential equations	Negative velocity of the follower, Velocity of the follower diverges to -∞	Average distance	-	Simulator

“- This information is not available/applicable for work”.

and a normalized three-dimensional Gaussian Process (GP) regression model to learn vehicle behaviour at a junction. Similarly, Wissing et al. [195] proposed an interaction-aware trajectory prediction method that used MCS to simulate interactions and forecast the distribution of potential future positions for the target vehicle. This approach leveraged the Intelligent Driver Model (IDM) [7] to account for the interacting behaviours of traffic participants, and the particles were disseminated using a lane-change driving model that considered three different lateral moves and aspects of the driving scenario with each run of the MCS. To summarize, sampling methods are essential tools for trajectory prediction, and the choice of method depends on the specific problem and the properties of the distribution of trajectories. However, these methods face challenges such as computational complexity, the requirement for efficient sampling strategies, and the possibility of overlooking important trajectory regions. Table 4 provides a summary of these models, encompassing design and implementation parameters like input features, model architecture, case studies, evaluation metrics, prediction horizon in (seconds), and dataset details.

3.3. Probabilistic models

A probabilistic framework in trajectory prediction refers to the use of probability theory to model and estimate the likelihood of future trajectories of objects or entities, such as vehicles, pedestrians, or other moving objects. It involves representing uncertainty and variability in the prediction process and providing probabilistic distributions or confidence measures for the predicted trajectories. In a probabilistic framework, trajectory prediction is typically formulated as a conditional probability problem, where the goal is to estimate the probability distribution of future trajectories given the observed past trajectories, sensor measurements, and other relevant information. This involves incorporating probabilistic models, statistical techniques, and machine learning algorithms to capture the uncertainties and dependencies in the data. Table 5 delivers a concise overview of probabilistic models, covering design and implementation parameters such as input features, model architecture, case studies, evaluation metrics, prediction horizon in (seconds), dataset details, and other supplementary information (remark).

3.3.1. Gaussian mixture model

A Gaussian Mixture Model (GMM) is a probabilistic model that is often used in trajectory prediction to capture the uncertainty and complexity of the data. It represents the distribution of the trajectories as a combination of multiple Gaussian distributions, each representing a possible mode or cluster of trajectories. A Semantic-based Intention and Motion Prediction (SIMP) was proposed by Hu et al. [89]. It uses multiple 2D GMM to model the probability distribution of movement patterns in driving scenarios and Deep Neural Networks (DNNs) to calculate the likelihood of entering the intersection area. GMM was also employed in other methods to model specific motion patterns [10], [194]. Although classic Hidden Markov Model (HMM) approaches have been quite successful at predicting drivers' moves, they do not take the impact of interaction-related aspects into account during the prediction process, therefore the results of their predictions are insufficiently accurate in real-world traffic situations. An interaction-related vehicle trajectory prediction model based on HMM and Variational GMM is proposed by Deo et al. [42]. The knowledge about vehicle interactions is discovered by locating the energy function's ideal solution. A GMM-HMM manoeuvre prediction model that takes interaction-aware elements into account is proposed by Zhang et al. [223] based on game theory. Jiang et al. [96] developed a GMM-HMM recognizer based on joint mutual information maximisation to estimate the driver's lane-changing intention. This recognizer was incorporated as a node in the Dynamic Bayesian Network (DBN) framework. To summarize, GMMs provide a versatile and robust method for trajectory prediction by capturing complex patterns and variations in the data. They are capable of handling multimodal distributions, which allows for representing different manoeuvre types or behaviour patterns exhibited by vehicles. However, it's important to note that training and inference with GMMs can be computationally demanding. Additionally, determining the optimal number of Gaussian components or modes in the model can be a challenging task.

3.3.2. Gaussian process

When utilizing Gaussian Process (GP) for trajectory forecasting, trajectories are considered as samples taken from a GP along the time axis. These samples are represented by N discrete points, which are mapped to an N-dimensional space. In this N-dimensional space, the samples adhere to a Gaussian distribution. During the modelling step, the GP model's main objective is to estimate the GP parameters based on these samples. By fitting the GP to the observed trajectory samples, the model captures the underlying patterns and dynamics of the data. The GP parameters, such as the mean and covariance, define the characteristics of the GP and determine the shape and uncertainty of the predicted trajectories [79]. In the study by Laugier et al. [111], GPs were employed to predict trajectories following the evaluation of likely behaviours using Hidden Markov Models (HMMs). Trautman et al. [179] addressed the frozen robot problem by utilizing GP for joint collision avoidance. Additionally, GP can be utilized to simulate interaction-related aspects in trajectory prediction tasks. Guo et al. [72] employed GPs and the Dirichlet process (DP) to construct motion processes and utilized a non-parametric Bayesian network to extract potential motion patterns. The prototype set was trained to represent each trajectory using methods based on prototype trajectories. The primary differentiation among these approaches lies in the technique employed to generate the prototype trajectory. Govea et al. [181]'s statistical analysis of the mean and

Table 5
Summary of Probabilistic methods with design and implementation parameters.

Work	Input	Model	Case Study	Evaluation metrics	Data	Remark
[188]	Acceleration, Yaw rate	Markov transition probability matrix, Frenet coordinate, SoftMax regression	Normal driving scenario, Cut in scenario, Road construction scenario	Time to collision, Time headway	NGSIM US-101	-
[223]	Coordinates, Speed, Acceleration	Game theory, GMM-HMM	Road with three lanes	Vehicle front drivable space, Collision risk, Ride comfort index	NGSIM I-80	Sliding time window method for feature extraction
[116]	Coordinates, Speed, Distance	Dynamic bayesian network	Lane-change, Higway	F1 score	NGSIM I-80, US-101	Universal algorithm for feature extraction
[89]	Coordinates	Mixture density network, Gaussian mixture model	Three driving lanes, and seven vehicles are considered, Sudden Change of Reference Vehicle, Typical lane change	Receiver operating characteristic curve, Recall, Precision, F1 score, RMSE	NGSIM US-101: 17,179 frames, 80% training, 20% testing	Negative log-likelihood loss
[10]	Coordinates, Angle, Velocity	Boosted decision trees, Agglomerative hierarchical clustering, Boosting	Highway scenarios	F1-score, Mean absolute position error	434 lane change manoeuvres; 70/30 ratio into training and test set	Pseudo-loss functions
[194]	Map	Gaussian mixture regression, Mixture of Experts	Highway scenarios	Balanced accuracy, Receiver operator characteristic curve, RMSE	Data collection at Stuttgart in Germany; training: 130 623 Trajectories, test: 20000 Trajectories	-
[42]	Coordinates, 3 second	Variational gaussian mixture models, Hidden markov models	Mean Absolute Error, Median Absolute Error, Execution time	Lane pass, Overtake, Left overtake	Data captured on Californian freeways: 52 video sequences	Prediction horizon:5 second
[72]	Coordinates, Velocity	DP-GP mixture model, Gibbs sampling	Highway traffic scenarios, Intersection traffic scenarios	Mean velocity	NGSIM I-80, US-101	
[76]	Vehicle ID, Lane ID	Dynamic bayesian network	Expressway scenarios, Two-lane prediction scenario	Accuracy, Precision, Recall, F1-score	NGSIM US-101	Training and inference perform on MATLAB

"- This information is not available/applicable for work".

variance of each sample of a trajectory yields the prototype trajectories. In their study, Hermes et al. [78] focused on capturing variations in vehicle movement through training. They divided the sample trajectories into multiple subsets and generated several prototype trajectories as an outcome of their research. In summary, GP is a valuable tool in trajectory prediction for AVs, providing several advantages such as flexibility, probabilistic forecasts, adaptability, and potential integration with other techniques. However, one limitation of approaches based on trajectory samples is their limited applicability to new contexts, which hinders their adaptability to diverse scenarios and environments.

3.3.3. Hidden Markov model

In trajectory prediction using the Hidden Markov Model (HMM), the observation sequences are comprised of the previous states of the traffic participants. The HMM algorithm is applied to estimate the most likely future observation sequence based on these past observations. Qiao et al. [152] offers a technique called HMTP* based on HMM that adaptively selects parameters to replicate real situations at a pace that changes over time. In [41], HMM and fuzzy logic are utilised to anticipate driver manoeuvres. HMM can also be included in planning and decision-making processes. HMM is employed in [188] for risk assessment and trajectory prediction, with the outcomes being supplied into the system for making decisions. A behaviour prediction method based on the HMM is proposed by Li et al. [117], considering the direction of incoming cars. To ensure the reliability of the prediction results, multiple sets of initial values are generated. Additionally, this approach aims to reduce the model's dependency on data, resulting in improved prediction performance. Ren et al. [155] propose the lane-changing behaviour recognition model based on the Continuous Hidden Markov Model (CHMM) is developed to identify the lane-changing behaviour of nearby vehicles. In summary, The HMM is highly beneficial for trajectory prediction due to its ability to capture temporal dependencies, handle missing or noisy data, and account for the uncertainty involved in predicting future trajectories. However, an assumption of HMM is that the hidden states are Markovian, implying that the probability of transitioning to a future state depends solely on the current state.

3.3.4. Dynamic Bayesian network

By incorporating time sequence and leveraging the Bayesian Network framework, the Dynamic Bayesian Network (DBN) offers a manoeuvre-based approach for trajectory prediction. DBN and Bayesian networks share fundamental concepts and methodologies for conducting probabilistic inferences. However, one distinction is that Kevin et al. [141] introduced the concept of time templates to address timing considerations in probabilistic models, while Bayesian Networks typically represent static systems. In the context of DBN, a time segment refers to a time template that discretizes continuous time into discrete points with a predetermined time granularity. Multiple vehicles driving manoeuvres are modelled by Gindele et al. [64]. All vehicle states, interaction relationships, observation statuses, road structures, etc. are included in the input data. DBN is used by Schreieret al. [160] to evaluate driving manoeuvres, and they use the kinematics model associated with each manoeuvre to forecast the trajectory. Game theory is used in [13] to anticipate the vehicle movement, and DBN, which takes interaction-related elements into account, then judges the vehicle motion. He et al. [76] employ DBN to recognise lane-change and vehicle-following motions and to forecast the trajectory. In [116], DBN is designed to consider the kinematic factors of vehicles, Vehicle manoeuvres, and their interdependence, and road-related information. To summarize, when utilized for trajectory prediction, DBN takes into account the interactions between traffic participants, leading to improved performance in conventional machine learning-based methods. However, DBN still encounters challenges in accurately recognizing manoeuvres and generating trajectories. Many existing methods are limited to distinguishing only two or three manoeuvres, such as lane-keeping and lane-changing, which restricts the model's ability to generalize to

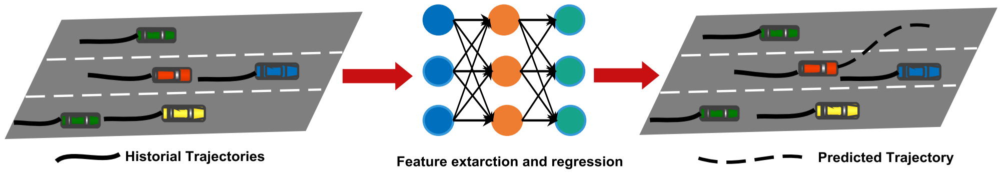

Fig. 7. The illustration of Deep learning-based methods.

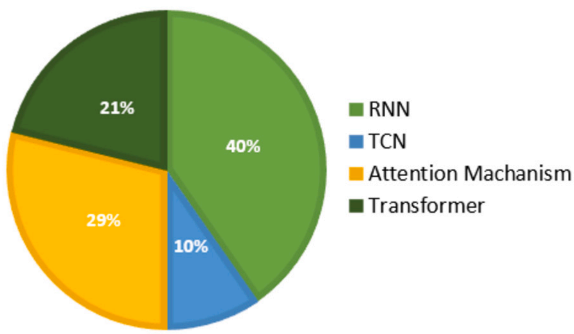

Fig. 8. Participation of Research articles in trajectory prediction task using Sequential learning-based approaches.

a wide range of scenarios. Consequently, there is a need for further advancements in DBN-based approaches to enhance their manoeuvre recognition capabilities and improve the model's generalisation ability in trajectory prediction tasks.

Traditional trajectory prediction methods excel at short-term prediction, their suitability for time-series forecasting is limited. As a result, their effectiveness diminishes over an extended prediction horizon, primarily due to infrequent consideration of temporal dependencies, often bypassed in their application. A potential solution to address the limitations is to incorporate models with enhanced temporal awareness, such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory networks (LSTMs) [238], and transformers [153] are known for their superior temporal awareness. By integrating these advanced models into trajectory prediction systems, one can improve the effectiveness over extended prediction horizons and enhance the overall accuracy of predictions.

4. Deep learning-based prediction methods

Conventional prediction techniques are only effective in basic prediction scenarios and short-term prediction assignments. Deep learning-based trajectory prediction models have gained popularity due to their ability to consider various factors that contribute to accurate predictions. These models take into account physical factors, such as the position, velocity, acceleration, size, and shape of vehicles. They also consider road-related factors like traffic signs, traffic lights, road geometry, and road obstacles. Additionally, interaction-related factors, including the distance between vehicles, relative speeds, and the presence of communication systems, are considered. Fig. 7 provides a general overview of these methods. The following sections outline the most prevalent deep learning-based methods used for trajectory prediction in Autonomous Vehicles (AVs).

4.1. Sequential modelling

Deep learning-based trajectory prediction methods often involve using a sequential network to extract features from historical trajectories and can serve as the output layer. These networks typically include Recurrent Neural Networks (RNNs), Temporal Convolutional Neural Networks (TCNs), Attention Mechanisms (AMs), and Transformers. Fig. 8 provides a visual representation, in percentages, of the distribution of research papers utilizing different algorithms in sequential modelling for trajectory prediction. It can be observed that TCNs are less commonly used in the AVs trajectory prediction task compared to other algorithms such as RNNs, AMs, and Transformers.

4.1.1. Recurrent neural network

The Recurrent Neural Network (RNN) was designed to handle temporal information, as opposed to conventional machine learning methods and Convolution Neural Networks (CNNs), which excel at processing spatial information [68], [173]. It maintains a record of past-time step data and combines input and hidden states to generate the desired output. However, when dealing with a large number of time steps, the RNN's gradient can either weaken or explode, causing issues. To tackle this problem, gated RNNs like the Long Short-Term Memory Network (LSTM) and Gated Recurrent Unit (GRU) have been developed. RNN-based trajectory prediction models are categorized as either single RNN models or multiple RNN models.

1) *Single RNN:* To predict trajectories based on manoeuvres or single-modal trajectory prediction, a single RNN model is employed. Additionally, it can be incorporated into auxiliary models to facilitate more complex capabilities, such as interaction-aware forecasting. In various studies [238,240,149], LSTM has been utilized as a sequence classifier for vehicle manoeuvre prediction. In these studies, LSTM cells extract vehicle attributes, and the output layer predicts movements using the final cell's hidden states. Fully connected layers are used to extract features and input them into three-layer LSTMs in [238,240], while two LSTM layers without embedding are employed in [149]. Altché et al. [8] used a single-layer LSTM to estimate the target vehicle's trajectory. Ding et al. [49] used an LSTM encoder to predict the target vehicle's manoeuvre by encoding its states. The LSTM encoder-decoder incorporating social and geographic information was compared to the Nearest Neighbour (NN) regression method in [27]. In [239], Zyner et al. utilized a weighted Gaussian Mixture Model (GMM) to forecast multi-modal trajectories. The GMM's parameters were obtained using an encoder-decoder three-layer LSTM model, and the predicted trajectories were clustered using the modal with the highest probability. Xing et al. [206] used GMM to identify driving styles, LSTM and fully connected regression layers to assess sequence data and driving styles to predict vehicle trajectory, with the first vehicle in the fleet following its predicted trajectory. Kawasaki et al. [102] integrated LSTM and KF for multi-modal trajectory prediction while considering lane information.

2) *Multiple RNNs*: The development of neural networks has resulted in the widespread usage of various types of RNN architectures. In Xin et al. [205], two separate LSTMs are employed to predict the target lane and trajectory of a vehicle based on its current state and expected lane. Deo et al. [44] suggest six LSTM decoders, each connected to a different manoeuvre, to forecast multi-modal trajectories. Dai et al. [40] utilizes two groups of LSTM networks to simulate the motion and interaction of nearby vehicles. Ding et al. [48] present a group of GRU encoders to characterize paired interactions between vehicles. Min et al. [136] use multiple RNNs and fully connected layers to generate the cubic polynomial coefficients that describe the target vehicle's future trajectory. Tang et al. [174] employ an attention mechanism to create a dynamic state encoder consisting of multiple RNNs sharing parameters to predict the multi-modal trajectory. Multi-modal trajectories are generated using an LSTM encoder-decoder and a multi-head attention layer in [133]. A paradigm with several LSTMs is proposed by Zhang et al.

Table 6
Summary of Recurrent Neural Network based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon (PH), Advantages and Limitation of each work, Summary of each work and Evaluation Metric (EM).

Work	#Trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[238] 2018	1	1.3 s	To more clearly see the areas of conflict and inaccuracies, the data is shown as an overlay on the map.	Interaction between vehicles does not take into account.	Single RNN: The fully connected layer of the model and the three LSTM layers are utilised for the output manoeuvres.	Accuracy
[49] 2019	1	4 s	Makes use of a policy anticipation network to make high-level policy decisions.	Interaction modelling for prediction is not investigated.	Single RNN: RNN to encode the vehicle's historical data.	RMSE
[27] 2019	1,3,6,9	3 s	Investigation into the use of rich maps for 3D object tracking and motion prediction.	Interaction between vehicles and road structure is absent.	Single RNN: LSTM encoder-decoder that incorporates social and geographic information.	minADE, minFDE, DAC, MR
[239] 2020	3	5 s	Clustering technique that captures the range of potential routes from the prediction output.	Modelling for only roundabout scenario.	Single RNN: Encoder-Decoder three-layer LSTM model incorporates all observations for a single track.	Euclidean Distance, MHD
[206] 2020	1	5 s	Inter-vehicle communication signals were used.	Prediction for the leading vehicle only.	Single RNN: LSTM layer to extract the time-sequence patterns for vehicle trajectory prediction.	RMSE,MHD
[205] 2018	1	5 s	1. Not impacted by internal dataset imbalance; 2. Lateral position's small prediction boundaries.	Model is not generalise for crossings and unstructured roads.	Multiple RNN: One LSTM to forecast the target lane and another LSTM to predict the trajectory based on the estimated target lane of the target vehicle.	RMSE
[44] 2018	1,6	5 s	Trajectories based on manoeuvre classes.	Assumption regarding six surrounding vehicles.	Multiple RNN: One LSTM encoder for the input sequence, six LSTM decoders to a different manoeuvre, and One LSTM decoder, to forecast trajectory.	RMSE
[40] 2019	1	6 s	Add shortened connections between the two LSTM layers.	Predicted trajectories don't show lane change procedure characteristics.	Multiple RNN: one group of LSTM is used to simulate the motion of nearby vehicles, and the other is used to simulate how nearby vehicles interact.	RMSE
[48] 2019	1	4 s	Modelling the pair-wise interaction using GRU.	Experiments restricted to a dataset of highways.	Multiple RNN: The trajectory encoding for each individual vehicle is obtained using an RNN encoder network.	NLL, accuracy
[174] 2019	1,12	5 s	All agents in the scenario do interactive, parallel step-wise rollouts.	The combination of discrete and continuous latent variables in the model is not investigated.	Multiple RNN: RNNs running in parallel to depict the agents in a scene.	NLL, minADE, minFDE
[224] 2021	1	5 s	Statistically examining numerous earlier traffic flow paths.	1. vehicle turning signals is not included; 2. This is not encounters in the conflict zone from approaching vehicles.	Multiple RNN: Two LSTM blocks: One for intention prediction, another for trajectory prediction.	MHD

in [224] for both trajectory and intention prediction. Xu et al. [203] introduce a student-teacher network for trajectory prediction, where the student algorithm is based on an LSTM Encoder-Decoder model, and the instructor algorithm is based on a Convolutional Graph Network. Although RNNs are widely used for analyzing and predicting data series, such as trajectory prediction, they have limitations in simulating spatial relationships, such as vehicle interaction, and processing image-like data, such as the context of a driving scene. This is why complex RNN-based solutions often require multiple techniques to overcome the limitations of a single RNN. In summary, RNNs offer a powerful approach for trajectory prediction by effectively modelling temporal dependencies. They can handle variable-length sequences and provide interpretability. However, they can suffer from vanishing or exploding gradients and sequential computation limitations. Understanding these factors is crucial when applying RNNs to trajectory prediction tasks. Table 6 summarizes the RNN-based approaches for trajectory prediction, providing information on the Prediction Horizon (PH) in seconds (s) and the number of predicted trajectories. The table also includes the Evaluation Metrics (EM) used for training and testing, along with highlighting the strengths and weaknesses of each study. Moreover, Table 7 examines input modalities and their observed history length (in second (s)), comparing encoding and decoding techniques in the models. It contrasts methods using diverse loss functions, optimisation approaches, and activation functions/learning rates during model training, also considering the platform on which these models are implemented. Additionally, it enumerates the number of samples in training, validation, and test datasets for each method.

4.1.2. Temporal convolutional networks

Temporal Convolutional Networks (TCNs) are a popular type of deep neural network architecture used in trajectory prediction tasks. In trajectory prediction tasks, TCNs are trained on historical trajectory data and are used to predict the future trajectory of a vehicle or pedestrian. Compared to recurrent networks, TCNs have been shown to outperform them in tasks such as handwritten recognition [163], audio synthesis [145], and time-series data [232]. One advantage of TCNs is their ability to handle variable-length sequences without information leakage. Bai et al. [14] employed causal convolution, dilated convolution, residual connection, and a completely connected network to create TCN. Zhang et al. [225] utilized TCN to predict lane-change manoeuvres and trajectories. In [171], CNN processes the rasterized image while TCN collects features from historical trajectory data that are combined with the raster feature and the present state. DeepTrack [101] is a lightweight deep learning algorithm with accuracy comparable to top trajectory prediction algorithms, but with a much smaller model size and reduced computational complexity. DeepTrack encodes the vehicle dynamic using TCN and reduces model complexity by using depthwise convolution as the fundamental building block. In [115], a TCN encoder and a Multi-Layer Perceptron (MLP) decoder are used, where the position and speed of the vehicles are sequentially entered and encoded as a context vector during the encoding procedure. To improve prediction accuracy, an intention recognition module is included with a TCN encoder. In [12], Mozhgan et al. integrate a dilated convolutional network-based encoder-decoder with a mixture density network to predict potential multimodal pathways taken by target vehi-

Table 7
Summary of Recurrent Neural Network based methods with design and implementation parameters.

Work	Input	Perceive (second(s))	Encoding	Decoding	Loss function	Optimiser	Activation function	Learning rate	Platform	Data
[238] 2018	Coordinates, Heading, Speed	5 s	LSTM	-	-	ADADELTA	-	Learning rate:0.03	Tensorflow, Nvidia Geforce 1080 GPU	Self-collected at roundabout; training:4560, validation:1658, testing:2074 tracks, Train: 6 hours, Test: 60 ms
[49] 2019	Coordinates, Heading, Speed	-	RNN	RNN	Negative Log-Likelihood (NLL)	-	Modified softmax layer	-	-	CARLA: single vehicle 21, 260 frames
[205] 2018	Coordinates, Map	5 s	LSTM layer	LSTM layer	l_2 loss	-	-	Learning rate:1.0	TensorFlow, GPU	NGSIM I-80 dataset (70% for training, 30% for validation)
[44] 2018	Coordinates, Map	3 s	LSTM	LSTM	NLL	ADAM optimizer	Leaky ReLU activation	Learning rate:0.001	Keras	NGSIM US-101, I-80
[40] 2019	Coordinates	5 s	LSTM	LSTM	Mean Square Error(MSE) loss	ADAM optimizer	-	Learning rate:0.001	Tensorflow	NGSIM I-80 and US-101, train 1200 trajectories in I-80 dataset
[48] 2019	Coordinates, Velocity, Acceleration	2 s	RNN Encoder	Fully connected layers	NLL	-	ReLU activation	-	Pytorch	NGSIM I-80 and US-101, 1,669 train, 509 test
[174] 2019	Coordinates	NGSIM-3 s; Argoverse -2 s	Bidirectional GRUs	Bi-directional GRUs	-	ADAM optimizer	ReLU activation	Initial learning rate: 0.001	Single nVidia Titan X GPU; pyTorch 1.0.1	CARLA simulator; NGSIM US-101, I-80; Argoverse
[224] 2021	Position, Velocity, Map	3 s	LSTM encoder	LSTM decoder	Boundary loss,NLL	-	-	Learning rate: 0.005	Pytorch	NGSIM 1 hour, INTERACTION; training data, and testing data in the 3:1

"- This information is not available/applicable for work".

cles. It is evident that TCN possesses benefits when it comes to handling time-series data. In summary, TCNs offer a powerful approach for modelling temporal dependencies in trajectory prediction tasks. They excel at capturing short-term and long-term dynamics, perform efficient parallel computation, and have interpretable receptive fields. However, spatial relationships and long-term memory might require additional considerations. Table 8 presents a summary of TCN-based approaches for trajectory prediction, including the prediction horizon (in second (s)), the number of trajectories predicted, and the evaluation metrics used for training and testing. The table also highlights the strengths and weaknesses of each study. Furthermore, Table 9 examines input modalities and history length (in second (s)), comparing encoding and decoding techniques in TCN-based models. It differentiates methods by considering various factors like loss functions, optimisation approaches, activation functions, learning rates, other training parameters, and the platform during the training process. It also lists the number of samples in training, validation, and test datasets for each method.

4.1.3. Attention mechanism

The Attention Mechanism (AM) is a cognitive model that approximates human thought processes by allowing for the efficient extraction of high-value information from a large volume of data using limited attentional resources. It is frequently used in deep learning tasks such as speech recognition [36], image classification [137], and natural language processing [86], with self-attention [182] being a popular method for identifying the weights and new context vectors based on the input sequence. Several recent studies have employed the attention mechanism for trajectory prediction and intention estimation. Hao et al. [74] proposed an encoder-decoder architecture combining GRU and self-attention, while Yan et al. [212] investigated a self-attention architecture with two types of self-attention mechanisms for the driving lane and driving context. Kim et al. [103] used self-attention to concentrate on features from the target vehicle's preferred lane, and Fu et al. [56]and Yu et al. [216] employed attention between the encoder and decoder components to selectively draw attention to particular context vector properties.

According to Wu et al. [196] and Meng et al. [132], the model can learn important spatial and temporal components for predicting and anticipating the movements of nearby vehicles. These models use a spatial attention layer to combine data from surrounding vehicles and a temporal attention layer to account for the temporal relationships between object agents. Lin et al. [123] proposed the STA-LSTM, which combines spatial and temporal information with an attention mechanism to explain how past trajectories and nearby vehicles affect the ego vehicle. Additionally, Kim et al. [105] proposed a model with a Baseline Network and Trajectory Proposal Attention, which is designed to simulate interaction-aware prediction. More recent work includes TP2Net [87], a trajectory prediction network that uses temporal pattern attention to extract latent multimodal driving information, and yang et al. [213] investigating the spatiotemporal dynamics between the ego vehicle and nearby cars, utilized spatiotemporal attention mechanisms in LSTM networks to perform lane change prediction and the trajectories of the vehicles. Several studies have employed multi-head attention and AM to extract information from lanes and vehicles, and model traffic interactions by analyzing attentions extracted from LSTM encoders. For instance, Kim et al. [104] utilizes multi-head attention to extract lane and vehicle information to predict future trajectory distributions. Messaoud et al. [135] also employs attention extracted from LSTM encoders to model traffic interactions. In Messaoud et al.'s [134] model, each attention head simulates a possible interaction between the target and

Table 8
Summary of Temporal Convolution Neural Network based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon (PH), Advantages and Limitation of each work, Summary of each work and Evaluation metric (EM) - This information is not available for work.

Work	# Trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[225] 2020	1	-	1. Estimating the vehicle's position and steering-wheel angle; 2. Consider the steering wheel angle to be a sign of lane-changing behaviour.	Merging and weaving zones are not evaluated.	TCN extracts features from the past and current states of actor of interest.	MAE,MSE
[171] 2020	1,3,6	3 s	Employ rasterized graphics to depict the location of the actor of interest and its changing surroundings.	The effects of surrounding vehicles on the ego vehicle is not clear.	TCN to predict the long-term lane-changing trajectory and driving behaviour	minADE, minFDE, MR, DAC
[101] 2022	1	5 s	1. Utilize depthwise convolution; 2. Lowering the size and operational complexity of models.	Due to ambiguity in driver behaviour, the model failed to forecast in several situations.	TCN utilized as an encoder to encode the vehicle dynamics.	RMSE,ADE FDE
[115] 2022	1	5 s	This takes into account the effects of the vehicle's driving behaviour.	The model is only appropriate for highway scenes.	Many TCN blocks make up a TCN, which encodes the input as a context vector.	RMSE
[12] 2022	1,3	4.8 s	1. Included intersections without signals; 2. The model doesn't depend on a specific action.	Interactions between the surrounding vehicles and other road users are absent.	Working with a mixed-density layer and dilated convolutional networks.	RMSE

Table 9
Summary of Temporal Convolution Neural Network based methods with design and implementation parameters.

Work	Input	Perceive (second(s))	Encoding	Decoding	Loss function	Optimiser	Activation function	Learning rate	Other parameters	platform	Data
[225] 2020	Coordinates	-	-	TCN	Mean Square Error(MSE) loss	Adam optimizer	ReLU	-	Dropout rate:0.1	TensorFlow	Driving simulator; validation: two-thirds of the training dataset, testing: one-third of samples
[171] 2020	Coordinates, Velocity, Acceleration, Orientation, Yaw rate, Map	2 s	-	5 layers of convolutional blocks	-	Rectified Adam Optimizer with Lookahead	Mish activation function	Initial learning rate:0.001	40 epoch	PyTorch	Argoverse 1; training: 205,942, validation:39,472, testing:78,143
[101] 2022	Coordinates	3 s	TCN	LSTM	-	ADAM optimizer	ReLU activation	Learning rate: 0.001	30 epochs	PyTorch	NGSIM's I-80 and US-101; 8 million data, 70% training, 10% validation, 20% test data.
[115] 2022	Coordinates, Map	3 s	TCN	MLP	MSE loss	Adam optimizer	ReLU activation;	Learning rate:0.001	-	-	HighD, training:54 files, test: 6 files of vehicle trajectory,
[12] 2022	Coordinates, Speed, Heading	0.56 s	Two distinct dilated convolutions layer	Residual blocks, MDN dense layer	Custom loss	-	-	Learning rate:0.002	Epoch 250, Dilation:1,2,4,8, Batch size:800, Decay rate:0.96, Grid search technique	Python, TensorFlow	ACFR:more then 60 hours of data, more than 14000 trajectories for training, 3000 for validation, 5943 for test

"- This information is not available/applicable for work".

context features. Hasan et al. [75] involves two Multi-Head Attention layers to capture the social and temporal interactions among vehicles. Additionally, incorporated a Multi-Head Attention-based decoder that includes an LSTM layer to decode the social and temporal interactions of the vehicles in a step-by-step manner. In summary, the attention mechanism in trajectory prediction improves the model's ability to focus on relevant information, handle variable-length sequences, provide interpretability, and enhance robustness to noise. However, it comes with potential drawbacks related to computational overhead, model complexity, attention bias, and data dependency. Table 10 presents a comprehensive summary of Attention-based approaches for trajectory prediction. It includes important information such as the prediction horizon (measured in seconds (s)), the number of trajectories predicted, and the evaluation metrics used for training and testing. Additionally, the table provides insights into the strengths and weaknesses of each study. Furthermore, Table 11 assesses input modalities and history length (in second(s)), comparing encoding and decoding techniques in the model. It distinguishes methods based on factors like loss functions, optimisation approaches, activation functions/learning rates, additional training parameters, and the training platform. It also includes the sample counts for each method in training, validation, and test datasets.

4.1.4. Transformer

Transformer is a neural network design that utilizes an attention mechanism concept and has been employed in various projects such as object detection [73], image segmentation [32], posture estimation [178], tracking, and trajectory prediction [192]. It was initially utilized for machine translation in Natural Language Processing (NLP) [22] and outperformed recurrent neural networks. Researchers have found the Transformer model to be effective for trajectory prediction,

Table 10
Summary of Attention based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon (PH), Advantages and Limitation of each work, Summary of each work, and Evaluation Metric (EM).

Work	# Trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[74] 2020	1	4 s	When the predicted time length increases, the attention mechanism keeps historical data from being lost.	Forecasts future position by using only the vehicle's prior position as input.	Attention mechanism combined with GRU decoder to forecast vehicle position.	RMSE
[212] 2020	1	5 s	A lane attention system combines real-time lane information.	1. The model is tested for highway scenarios; 2. The road information excluded.	Context attention and lane attention are the two spatial-attention strategies used to explain how vehicles interact.	RMSE
[103] 2021	1,5,6,12	6 s	Extracting the joint properties related to the lane and the trajectories of the nearby agents.	Speed and yaw angle are not included with the positional coordinate of vehicles.	The self-attention technique used to concentrate on features from the target vehicle's preferred lane.	ADE,FDE
[56] 2021	1	3 s	1. Construct a spatial and temporal navigation map; 2. Predict current location and velocity of surrounding vehicles.	1. Unstructured and urban roads are not addressed; 2. Select the most influential six vehicles around the ego-vehicle.	Attention mechanism occurs between the encoder and decoder components.	RMSE
[216] 2021	1	5 s	Constraint net used to extract and model the external environmental constraints.	It is challenging to generalise the model in complex conditions.	The most remarkable vehicles are chosen at each time step using an attentional decoder with LSTM.	RMSE
[196] 2021	1	5 s	1. Concurrently model spatiotemporal interactions; 2. Trained the model in an end-to-end fashion	1. Referring to the entire scene's agents as the neighbourhood; 2. Certain cases at the roundabout fail.	Multi-head attention is utilised to represent temporal correlations of interactions, and a State Gated Fusion (SGF) layer is applied to combine spatial and temporal interactions.	ADE,FDE
[132] 2021	1	5 s	Reflecting the spatial relationship between the surrounding vehicles and the target vehicle.	Predicting the longitudinal trajectory is challenging.	Describe the spatial and temporal attention modules separately.	RMSE,NLL
[104] 2020	1	5 s	1. Interactions can unsupervisedly learn to focus on a few key vehicles; 2. Model is scalable with any number of nearby vehicles	The model is for highways only.	Both a vehicle attention layer and a lane attention layer are present in the encoder. The decoder has a vehicle attention layer only.	RMSE
[135] 2020	3	5 s	1. Non-local social pooling modelling the interaction with a multi-head attention mechanism; 2. Incorporate the data about the vehicle class	1. There are no scenarios for mixed or heterogeneous traffic; 2.Road structure is not included.	the attention layer in between the encoder-decoder based on LSTM,models the interactions between the target and the neighbouring vehicles	RMSE,Min and Max RMSE
[134] 2021	1,15	6 s	1. Combined representation of the agents and the static scene; 2. Attention maps are shown visually.	Specify the interaction space explicitly using a defined distance.	Each attention head simulates an interaction between the combined context features and the target.	minADE, minFDE, MR, off-road rate
[123] 2021	1	5 s	The inception-based module was used to maintain the spatial information of the surrounding vehicles.	1. Fewer nearby vehicles taken into account; 2. Trajectory prediction depend on manoeuvre.	LSTM-based attention is used to extract the target vehicle's driving intentions and temporal driving behaviours.	RMSE
[213] 2022	1	5 s	1. Encourages a deeper understanding of the lane-change process in vehicles; 2.Target vehicle's time series attention module and the nearby vehicles' spatial attention module.	When changing lanes, neither the pose nor the change in speed of a target vehicle is taken into account.	LSTM networks with a spatiotemporal attention mechanism for extracting trajectory features.	RMSE
[105] 2022	1	5 s	Modelling the link between the vehicles using grid-based discretisation.	Road structure and their interaction is not included.	LSTMs and spatial-temporal attention mechanisms are combined for explainability.	RMSE
[87] 2022	1,6	3 s	1. Considers all interactions, including agents-agents, lanes-agents, lanes-lanes, and agents-agents; 2. A novel quantitative evaluation metric proposed.	Performance margin is insignificant with the other models.	An attention module combines the characteristics of many proposals that reflect various actions or behaviours.	minADE, minFDE, minLaneFDE
[75] 2023	1	5 s	Encoding social and temporal interaction using multi-head attention.	1. Model is not adopted for urban driving; 2. This is not using spatial HD maps, traffic signals, or lane kinds	Social and temporal attention layer in both encoder-decoder.	RMSE

with Quintanar et al. [153] modifying a standard transformer to incorporate past trajectories as an input feature extracted from aerial view photo datasets. Another approach suggested by Liu et al. [127] involves a multi-modal prediction architecture consisting of stacked transformers that gather features from historical trajectories, road data, and social interaction. Meanwhile, Zhao et al. [228] utilized a transformer network with residual layers to predict trajectories that account for interaction, using fully linked feed-forward networks and pooling operations to integrate geographical data and enable the transformer to learn interaction aspects. The Spatio-Temporal Transformer Networks (S2TNet) [29] use spatio-temporal Transformer to represent spatio-temporal interactions and temporal Transformer to handle temporal sequences. Chen et al. [31] propose a novel non-autoregressive model for predicting vehicle trajectories based on transformers, utilizing a self-attention module to define the dynamic variation in social behaviour and a graph attention module to represent the interactions between vehicles. The Structural Transformer suggested by Hou et al. [82] is a recurrence-free multi-sequence learning network that grasps interactions between surrounding vehicles along both temporal and geographical dimensions simultaneously. Huang et al. [93] propose a Transformer-based multi-modal trajectory prediction model using a multi-head attention Transformer layer to model the relationship between interacting agents. The Scene-Transformer is a transformer-based model introduced by Ngiam et al. [142] that uses attention to mix features from agent interactions and

Table 11
Summary of Attention-based methods with design and implementation parameters.

Work	Input	Perceive (second(s))	Encoding	Decoding	Loss function	Optimiser	Activation function	Learning rate	Other parameters	Platform	Data
[103] 2021	Coordinates, Lane information	2 s	CNN-LSTM	Fully connected layer	Mean absolute error loss, Cross-entropy loss	Adam optimizer	-	Learning rate:0.0003	Batch size: 32	-	NuScenes: 245,414 trajectories, 1,000 different scenes and Argoverse: 324,557 scenarios
[56] 2021	-	5 s	LSTM	LSTM	Root mean squared error loss	Adam optimizer	-	Learning rate:0.002	Decay rate:0.96, Epoch:300, Batch size:16	TensorFlow	NGSIM US101;training data:34101, testing data:8526
[216] 2021	Coordinates, Map	3 s	LSTM	LSTM	-	Adam optimizer	Leaky ReLU activation with α:0.1	-	Epoch:128, Batch size:8	-	NGSIM I-80 and US-101
[196] 2021	Coordinates	3.2 s	Multi-Head Attention, Graph Attention Network,LSTM	LSTM	L2 loss, Variety loss	Adam optimizer	ReLU activation function	Learning rate: 0.0001	8 parallel heads	Python, Pytorch	NGSIM and highD: 70% training, 10% validation, and 20% testing data
[132] 2021	Coordinates	5 s	LSTM	LSTM	-	Adam optimizer	-	Learning rate:0.001	Dropout:50%	Pytorch	NGSIM; training:70%,verification and testing:30%
[134] 2021	Coordinates, Velocity, Acceleration, Yaw rate; Map	2 s	LSTM, CNN	LSTM	Negative log-likelihood loss, Cross-entropy loss, Off-road loss	-	Softmax activation	-	16 parallel attention operations	-	NuScenes; train:32,186, validation:8,560, test:9,041 instances
[213] 2022	Coordinates, Velocity, Acceleration, Yaw rate	10 s	LSTM	LSTM	Mean square error	-	-	Learning rate:0.003 with StepLR adjustment strategy	Epochs:210; Batch size:100	Pytorch	NGSIM I-80:4960 lane change trajectories; train and test
[105] 2022	Coordinate, Map	2 s	1D CNN, A feature pyramid network,GCN	-	Hinge loss, Winner-Takes-All Loss, Distance error loss	Adam optimizer	-	Learning rate:0.001	40 epochs with Batch size:128	Pytorch	Argoverse
[87] 2022	Coordinates, Velocity, Acceleration, Class	3 s	LSTM	LSTM	Mean squared error, Cross entropy loss	Adam optimizer	Leaky rectified linear unit activation, Sigmoid activation	Learning rate: 0.0001; Minimum learning rate: 0.000001	Mini-batch size:128	Python, PyTorch	HighD:14 records, training 10, testing:2, verification:2 record; NGSIM I-80
[75] 2023	Coordinates	3 s	LSTM, Multi-head attention	LSTM, Multi-head attention	-	Adam optimizer with α : 0.001, β : 0.999	-	Learning rate: 0.0001	Batch size:32	-	BLVD:654 videos; choose the scenario involving highways

“- This information is not available/applicable for work”.

road graphs in both space and time. The LaneTransformer proposed by Wang et al. [190] combines the characteristics of the features between the roadways and the agents using a stack of transformer blocks, and high-order interactions are aggregated using an attention-based block. Wang et al. [185] propose a mixture-of-experts approach utilizing a transformer to model the interactions between vehicles explicitly considering their driving styles for building a multimodal motion planner. The study in [59] proposes a dual Transformer model to demonstrate the relationship between intentions and trajectories for the target vehicle. As demonstrated by these studies, the use of transformers provides several advantages in handling time-series data in trajectory predictions. To summarize, transformers have shown their potential in trajectory prediction by capturing complex dependencies and interactions. They offer scalability, transfer learning capabilities, and the ability to handle multiple agents. However, they require substantial computational resources and may have challenges in interpretability and data efficiency. Table 12 summarizes the Transformer-based approaches for trajectory prediction, presenting key details such as the prediction horizon (measured in seconds (s)), the number of trajectories predicted, and the evaluation metrics employed for training and testing. Furthermore, the table highlights the strengths and weaknesses of each study. Moreover, Table 13 assesses input modalities, observes length, and compares encoding and decoding techniques in the transformer-based model. It distinguishes methods based on factors like loss functions, optimisation approaches, activation functions, learning rates, additional training pa-

Table 12
Summary of Transformer based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon, Advantages and Limitation of each work, Summary of each work, and Evaluation Metric (EM) - Predicting any number of trajectories for the work.

Work	# trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[153] 2021	1	5 s	Providing more information (position and direction), to assess its impact on the model's performance.	The inputs are only the track history of the target vehicle.	Modified a standard transformer and utilising the augmented data for the context of vehicles.	ADE, FDE
[127] 2021	1,6	3 s	Create a region-based training technique that guarantees that each proposal can capture a certain mode.	Interaction of vehicle to infrastructure is missing.	Stack transformers are used to model multimodality at feature level with fixed independent proposals.	minADE minFDE MR
[228] 2021	1	5 s	To gauge how socially interactive agents are, a channel-wise module is inserted.	Trajectory prediction depends on historical data, yet future trajectories are still unpredictable.	Transformer networks with residual layers are used to predict the trajectory and learn interaction aspects.	ADE, FDE, RMSE
[29] 2021	1	3 s	Extract information about interactions both in terms of spatial and temporal dimensions.	There needs to be integrated map data.	Transformer is made to record all traffic agents' spatiotemporal interactions, not just their spatial neighbours.	WSADE, WSFDE
[31] 2022	-	5 s	The graph attention and the sparse self-attention mechanisms are used for social interaction and temporal interaction, respectively.	Environmental data, such as detailed maps and traffic conditions, are excluded.	Transformer uses social and temporal attention modules to capture correlations from raw trajectory data.	RMSE
[82] 2022	1	5 s	1. Simultaneously learns the temporal and spatial relationships between various SVs; 2. An approach without recurrences to enhance the speed-accuracy trade-off in long-term prediction.	The effectiveness of the attention weights in the attention layers is unconfirmed in terms of their ability to reveal the driver's true attention to the other SVs.	Structural Transformer employed by utilising the two-layer encoder-decoder architecture for parallel trajectory prediction for multiple SVs.	Final position error, Time cost
[93] 2022	1,6	3 s	Modifies the multi-head attention method to support multi-modal prediction.	Traffic conditions and rules are excluded.	Extracting relationships between interacting actors using a transformer encoder.	minADE minFDE MR
[142] 2022	1,6	5 s	1. Model the query using a masking method; 2. Use the attention method to integrate features from different road components, agent interactions, and time steps.	Captures agent-to-agent interactions, excluding infrastructure interactions with agents.	transformer to carry out conditional motion prediction, goal-conditioned prediction, and motion prediction.	minADE, minFDE, miss rate, and mAP.
[185] 2023	-	5 s	This takes into account the driving habits of other vehicles.	Insufficient generality for trajectory planning.	Transformers are used to encode the vehicle interaction.	ADE,FDE
[59] 2023	1	4 s	The intention prediction of the target vehicle used as input for the trajectory prediction module.	Only highway scenarios may be predicted using this approach.	Dual transformer utilised: one for intention prediction, the other for trajectory prediction.	RMSE
[190] 2023	6	3 s	Reduce the model's parameters to make it smaller in size.	The number of generated trajectories for the target vehicle has to be predetermined.	Stack of transformers to connect the highways' and agents' features.	minADE, minFDE, miss rate

rameters, and the training platform. It also includes the sample counts for each method in training, validation, and test datasets.

4.2. Vision based modelling

There are two types of prediction methods, which differ in how they formulate predictions. The first is the Bird-Eye-View (BEV) approach, which uses an algorithm to process data from a top-down, map-like view. The second is ego-camera prediction, which involves viewing the world through the perspective of the ego-vehicle. However, the ego-camera approach is generally more challenging than the BEV approach due to various factors [176]. Firstly, the BEV approach offers a broader field of view and more accurate predictions, whereas the ego-camera approach has a narrower field of vision and a limited prediction horizon. Additionally, the ego-camera approach is more prone to obstructions than the BEV approach. Despite these difficulties, the ego-camera approach is still more beneficial than the BEV approach because most vehicles do not have access to cameras that can locate target agents and BEVs on the road. Therefore, a prediction system should be able to view the world from the perspective of the ego vehicle, as demonstrated in Fig. 9. The illustrations of various vision-based techniques and their contribution to solving the trajectory prediction task in Autonomous Vehicles (AVs) are depicted in Fig. 10. Each technique makes a roughly equal contribution to the trajectory prediction in AVs research paper. This section highlights the inclusion of Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs) in addressing this domain.

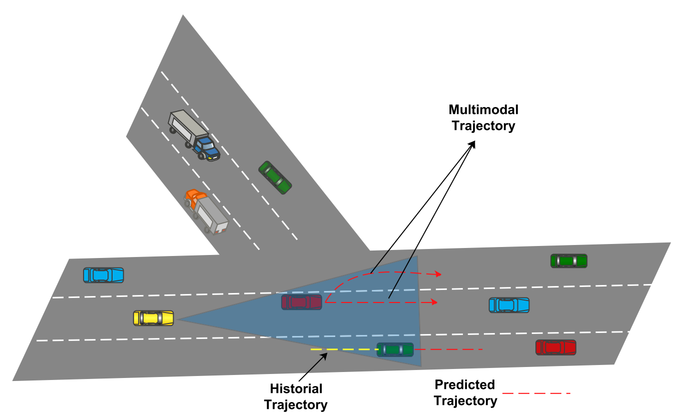

Fig. 9. The ego-camera prediction algorithm adopts the perspective of the ego-vehicle to observe the environment, identifying relevant information about the target and surrounding vehicles, such as bounding boxes, RGB frames, position, speed, type, etc. This enables the algorithm to make predictions about their future locations.

4.2.1. Convolutional neural network

Convolutional Neural Networks (CNNs) have been successfully applied to various computer vision tasks, including trajectory prediction. Although CNNs are primarily designed for image data, they can be adapted for trajectory prediction by treating the trajectory sequence as a structured grid-like input. Recently, CNN has shown success in vari-

Table 13
Summary of Transformer based methods with design and implementation parameters.

Work	Input	Perceive (Second(s))	Encoding	Decoding	Loss function	Optimiser	Activation function	Learning rate	Other parameters	Platform	Data
[127] 2021	Past trajectories, Map	2 s	Transformer	Three-layer MLP	Huber loss, KL divergence, and an auxiliary loss	AdamW optimizer	-	Initial learning rate:0.001	Weight decay:0.0001 and gradient max norm:0.1	-	Argoverse:205,942 training, 39,472 validation and 78,143 testing cases
[228] 2021	Coordinates	3 s	Multi-head attention	Masked multi-head attention	L2-loss	Adam optimizer	-	Learning rate: 0.01	Dropout value: 0.1	PyTorch	NGSIM US-101 and I-80: train,test, validation
[29] 2021	Coordinates, shapes, Categories	6 s	Spatio-temporal Transformer, Temporal Transformer	Temporal Transformer	L2-loss	Adam optimizer	-	-	Dropout ratio:0.1	Pytorch	ApolloScape
[31] 2022	Coordinates	3 s	Social Attention Learning, Temporal Attention Learning	Cross-attention learning, Temporal Attention Learning	Cross-entropy loss, MSE loss and NLL loss	-	-	-	α -entmax transformation	Pytorch	NGSIM U.S. 101 and I-80 and HighD; training:70%, validation:10%, and test sets:20%
[82] 2022	Coordinates, Velocity, Acceleration	10, 20, 50, 100 steps	Spatial Encoding, Temporal Encoding, Two structural multi-head self-attention layers and two dense layers	Spatial Encoding, Temporal Encoding, structural multi-head self-attention layers and dense layers	MAE loss	Adam optimizer	ReLU activation	Learning rate:0.0002	Batch size: 128; 100 Epochs	Python 3.7, Tensorflow-2.6	NGSIM U.S. 101 and I-80: 71304 samples: training: 64174 and testing set:7130 samples
[93] 2022	Coordinates, Map	2 s	Map Encoder: fully connected layer, max-pooling operation; Agent-agent Encoder:multi-head attention; Agent-map Encoder:multi-modal attention	Trajectory decoder:fourlayer MLP; score decoder:fourlayer MLP	Minimum over N (MoN) loss, Cross-entropy loss	Nadam optimizer	Activation functions:ELU	Learning rate that starts with 0.0001 and decays by 50% after every 20 epochs	Training epochs: 100; Batch size:64	Tensorflow	Argoverse:205,942 training, 39,472 validation, and 78,143 testing sequences
[142] 2022	Coordinates, shapes, Categories, Velocity, Map	-	Factorized self-attention, Cross-attention	Self-attention, 2-layer MLP	Displacement loss	Adam optimizer	-	-	Total epochs: 150; Linear ramp-up: 0.1 epochs; batch size:64	TensorFlow	Argoverse:324,000 segment;Waymo Open Motion Dataset (WOMD):104,000 segments
[185] 2023	Coordinates, Yaw angle, Map	-	MLP layer, Sinusoidal encoding and 3-layer PointNet[]	Driving style decoder: dot product attention layer and an MLP layer, trajectory decoder: scaled dot product attention layer and separate MLP	Imitation loss, Multimodal loss	-	-	-	K-means clustering for driving style labels based on vehicle trajectories.	-	Lyft: train and test
[59] 2023	Distance, Speed	3 s	Multi-head attention module	Two multi-head attention modules	RMSE loss	Adam optimizer	Tanh activation, Softmax activation	Learning rate:0.005	Batch size:15.	Tensorflow 2.4.0	NGSIM U.S. 101 and I-80:3972 training and 993 test samples; highD: 9780 training and 2445 test samples
[190] 2023	Coordinates, Map	2 s	Attention block, CNN and MLP	MLP	MSE loss,Euclidean distance	Adam optimizer	-	Learning rate from 0.0005 to 0.005	64 epochs with a batch size of 128	TensorRT	Argoverse: 205,942 training, 39,472 validation and test 78,143 sample

“- This information is not available/applicable for work”.

ous tasks, including machine translation [60] and computer vision [19]. In the context of trajectory prediction for autonomous vehicles, CNN is commonly used for vision-based prediction, where features are extracted from images captured by frontal cameras. Nikhil et al. [143] found that using CNN for trajectory prediction is superior to RNN, as trajectory has significant spatio-temporal continuity. They stacked the

Table 14
Summary of Convolution Neural Network based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon (PH), Advantages and Limitation of each work, Summary of each work, and Evaluation metric (EM) - Predicting any number of trajectories for the work/This information is not available for work.

Work	# trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[39] 2019	-	6 s	In closed-course tests, the suggested technique was tested within self-driving cars.	The process creates a raster image that encodes the context of each actor,	Using MobileNetV2 [158] as a feature extractor.	Displacement errors
[51] 2020	1	3 s	A fleet of self-driving cars were fitted with the system after rigorous offline and online testing.	Vehicles interactions are not taken into account.	CNN inputs a raster image to extract significant features.	Displacement errors
[147] 2020	1,5,10,15	6 s	Frame the problem of trajectory prediction as a classification task over a wide range of trajectories.	The model outperforms for the urban driving datasets only.	The model's foundation is ResNet-50 [175].	minADE,FDE, HitRate
[130] 2020	1,5	4 s	The ability of the memory to assimilate fresh samples lowers the error on unobserved data.	The past trajectory of the target vehicle and road-related information are considered.	A MANN with a non-episodic memory.	ADE,FDE
[226] 2021	1,3	-	To forecast various agents' driving trajectories.	Model's three output trajectories only and their associated confidence intervals.	Apply the ResNet [175] model to the input images to learn complicated feature representations.	NLL

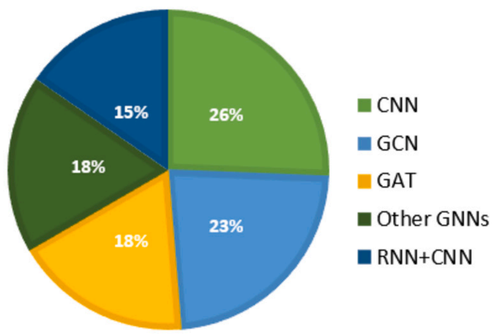

Fig. 10. Participation of Research articles in trajectory prediction task using Vision-based approaches.

convolutional layer after a fully connected layer to create time continuity and used a fully connected layer to output the future trajectory, taking the past trajectory as input. This CNN-based network operates faster, according to experiments.

However, most techniques that use the CNN framework take a Bird's-Eye View (BEV) as their input, displaying a top-down view of the traffic situation. BEV images can be created using multiple data sources, including LiDAR point clouds, Occupancy Grids (OG), and High-Definition Maps (HD-Maps). Some recent studies utilized CNN to extract features from sophisticated BEV representations. For example, MobileNetV2 [158], a memory-effective CNN designed for mobile apps, was used in [39], [51] to output potential trajectories and their likelihoods. The trajectory prediction of Vulnerable Road Users (VRUs) is addressed in [37] through a new rapid CNN architecture that utilizes context rasterisation techniques [51]. In [147], the vehicle state and the raster image were used to build a set of potential future trajectories, and the trajectory with the highest probability was selected as the future trajectory by examining semantic properties. A novel rapid CNN architecture was proposed in [130] for trajectory prediction of vulnerable road users, where a Memory Augmented Neural Network (MANN) was used to produce multimodal trajectories.

Recent studies have also proposed new techniques that forecast trajectory using CNN and produce cutting-edge results. For instance, Gilles et al. [63] generates a heatmap of the agent's potential future, while Ye et al. [214] uses the point cloud learning method to incorporate both spatial and temporal data into trajectory prediction. Zhuoren et al. [226] used ResNet-50 [175] to anticipate the trajectories of AVs such as vehicles and pedestrians. ResNet-50 [175] can effectively collect information from multiple dimensions to produce superior forecasts with the three trajectories and their confidence levels. While processing raster maps with CNN involves significant computational costs and information loss, vector maps can be used as nodes in Graph Neural Networks (GNN) for trajectory prediction. To summarize, CNNs offer advantages in capturing spatial patterns and recognizing spatial relationships in trajectory data. They are efficient in terms of parameter sharing and can handle larger datasets. However, they may struggle with modelling temporal dependencies and handling variable-length sequences. Table 14 presents a summary of CNN-based approaches for trajectory prediction, including the prediction horizon measured in seconds (s) and the number of trajectories predicted. The table also provides an overview of the evaluation metrics used for training and testing, as well as highlighting the strengths and weaknesses of each study. Additionally, Table 15 evaluates input modalities, observes length, compares encoding and decoding techniques of CNN-based methods, and distinguishes methods based on factors like loss functions, optimisation approaches, activation functions/learning rates, and the training platform. Additionally, it provides sample counts for each method in training, validation, and test datasets. The approaches for predicting vehicle trajectory based on GNN will be covered in the following sections.

4.2.2. Graph neural network

When considering prediction techniques that take interaction-related factors into account, each element of the environment can be viewed as a node in a graph. However, many real-world applications generate data from non-Euclidean spaces, and traditional deep learning-based methods that analyze Euclidean spatial data perform poorly in such cases. Each scene can be represented as an irregular graph with variable-sized unordered nodes, and some crucial operations, such as convolution, are not directly applicable to the graph due to variations in the number of nearby nodes. Nevertheless, every node in the graph is connected to other nodes by edges, which can be used to determine the interdependence of various objects. Graph Neural Networks (GNNs) are highly suited for vehicle trajectory prediction challenges based on interaction-related elements [197]. The methodology is described in Fig. 11. This idea is supported by Diehl et al. [47], who demonstrate the effectiveness of trajectory prediction using two well-known graph networks: the Graph Convolutional Network (GCN) and the Graph Attention Network (GAT). In addition to assessing input modalities and observing length (in seconds(s)), Table 16 compares encoding and decoding procedures of graph neural network-based methods such as GCN, GAT and other methods. It differentiates methods based on optimisation approaches, loss functions, activation functions/learning rates, training parameters, and training platforms. Additionally, it provides sample numbers for each method in the training, validation, and test datasets.

Table 15
Summary of Convolution Neural Network-based methods based on design and implementation parameters.

Work	Input	Perceive (second(s))	Encoding	Decoding	Loss function	Optimiser	Learning rate	Platform	Data
[39] 2019	Image, state	-	MobileNetV2 [158]	-	L2-norm loss, Mixture-of-Experts (ME) loss	Adam optimizer	Initial learning rate:0.0001	TensorFlow	240 hours of data; manually driving; 7.8 million examples, train/validation/test split 3:1:1
[51] 2020	Images	-	CNN	LSTM	Displacement error	Adam optimizer	Initial learning rate:0.0001	TensorFlow	240 hours of data; manually driving; 7.8 million examples, train/validation/test split 3:1:1
[147] 2020	Speed, Acceleration, Yaw rate; Images	-	ResNet-50,	Fully connected layer	Cross-entropy loss, L1 loss	-	Learning rate:0.0001, Drop value:0.1	-	Internal self-driving dataset: 11 million data points, train:1 million, validation:300,000, and test: 300,000 data points; NuScenes: train:32,186, train-val: 8,560, validation:9,041 set
[130] 2020	Coordinates, Image	2 s	Gated recurrent units	Gated recurrent units	Mean squared error loss	Adam optimizer	Learning rate: 0.0001	-	KITTI: train:8613, test:2907 trajectories

"- This information is not available/applicable for work".

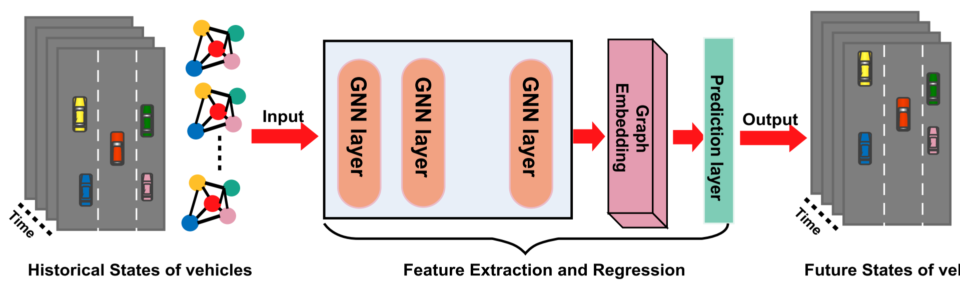

Fig. 11. Depiction of the graph neural network for trajectory prediction task.

4.2.2.1. Graph convolutional network The Graph Convolutional Network (GCN) is a popular technique in the field of graph neural networks. It extends the convolution operation from traditional image data processing to graph data processing. The key idea is to create a mapping function that can extract interaction-aware features from the node features in the network and their neighbouring nodes. Li et al. [120] proposed GRIP, a graph convolutional network-based trajectory prediction model that considers the interaction-related factors by treating vehicles as nodes within the network at each sampling time. GRIP utilizes a fixed graph network to describe the interaction-related characteristics between traffic participants and employs an LSTM encoder-decoder to forecast the trajectory of nearby vehicles using the output of GCN. To improve the accuracy of GRIP, Li et al. [119] proposed GRIP++, which uses both fixed and dynamic graph networks and achieved top ranking in the Baidu Apolloscape dataset [91] at the end of 2019. Jeon et al. [95] proposed SCALE-Net, which can predict the trajectories of any number of nearby vehicles while maintaining performance by using an Edge-Enhance Graph Convolutional Network (EGCN) [66] to learn edge features in the traffic flow. Chandra et al. [26] proposed a two-layer GNN-LSTM structure to resolve the trajectory prediction issue by using an LSTM encoder-decoder in the first layer to predict the future trajectories of traffic participants and a weighted dynamic geometric graph network in the second layer to represent the interaction-related characteristics of traffic participants. Zhao et al. [231] proposed a spectrum-based GCN network that allows all vehicles in the scene to communicate information to take into account how the surrounding vehicles are changing and adapting to the environment. Sheng et al. [164] proposed the GSTCN network, which uses a GCN to address spatial interactions, a CNN to capture temporal data, and a gated recurrent unit network to encrypt and decrypt the spatiotemporal properties to produce future trajectory distributions. Xu et al. [208] proposed a group vehicle trajectory prediction model with a global spatiotemporal graph that can thoroughly analyze the temporal and geographical association between previous vehicle trajectories. Dongwei et al. [209] suggested the MVHGN forecast, a graph neural network-based model for predicting the future paths of heterogeneous traffic-agents that employs a multi-view logical network by fusing various logical correlations and the multi-view logical characteristics derived by the graph convolution module. In summary, GCNs offer a promising approach for trajectory prediction by explicitly modelling the spatial dependencies and relationships among objects. They can effectively capture contextual information and handle irregular graph structures. However, scalability, graph construction, and temporal dependency modelling should be carefully considered when applying GCNs to trajectory prediction tasks. Table 17 provides a summary of GCN-based approaches for trajectory prediction including the prediction horizon measured in seconds (s) and the number of trajectories predicted, along with the evaluation metrics used for training and testing. The table also highlights the strengths and weaknesses of each study.

4.2.2.2. Graph attention network The method for collecting data from the one-hop neighbourhood varies greatly between Graph Attention Network (GAT) and GCN, with GAT employing the attention mechanism in place of the statically normalized convolution process. Velickovi'c et al. [241] proposed the GAT. In [139], an encoder-decoder design was used along with GAT to extract spatial interaction information from a heterogeneous digraph, consisting of automobile and local road map vertices. The Repulsion and Attraction Graph Attention (RAGAT) model was introduced in [50], which uses two stacked GATs to predict trajectories based on free space and vehicle condition information. In [138], a three-channel system with a heterogeneous edge-enhanced graph attention network was developed to address the heterogeneity of vehicles

Table 16
Summary of Graph Neural Network-based methods with design and implementation parameters.

Work	Input	Perceive (second(s))	Encoding	Decoding	Loss function	Optimiser	Activation function	Learning rate	Other parameters	Platform	Data
[120] 2019	Coordinates	3 s	Graph convolutional layer, Graph operation layer, LSTM	LSTM	Custom loss	Stochastic Gradient Descent optimizer	Tanh activation function	Starting learning rate:0.001	-	PyTorch	NGSIM I-80 and US-101: train:75% and test:25% sets.
[69] 2021	Coordinates, Map	Argoverse: 2 s; Waymo open:1 s	Hierarchical GNN	-	Smooth l1 loss, l1 loss, Cross-entropy loss	-	-	Initial learning rate:0.001	Hill climbing algorithm; Batch size:64	-	Argoverse: train: 205942, validation:39472, test:78143 sequences,;Waymo Open Dataset: 100,000 scenes, each 20 seconds long
[217] 2021	Coordinates, Map	2 s	2 Lane convolution layers, Shortcut layer, Lane pooling layer	Two layers MLP	Smooth-L1 loss, Cross entropy loss	Adam optimizer	Relu activation	Initial learning rate: 0.01	Batch size: 64; Epochs:30	-	Argoverse: train: 205942, validation:39472, test:78143 sequences
[164] 2022	Coordinates	3 s	GCN,CNN, GRU	GRU	Negative Log Likelihood(NLL) loss	Stochastic Gradient Descent (SGD) optimizer	-	Initial learning rate: 0.1	Epochs:250; Batch:128	PyTorch	NGSIM I-80 and US-101: 13,218 segments of trajectories, randomly split into training, validation, and testing sets.
[209] 2023	Coordinates, Class	-	Time convolution layer, GRU	GRU	Mean Square Error(MSE) loss	Adam optimizer	Nonlinear Relu	Initial learning rate: 0.001	Batch size: 32; Epochs:100	Pytorch	ApolloScape: 53 minutes of training, 50 minutes of test sequences
[50] 2021	Coordinates, Image	3 s	LSTM, Dual-Graph Attention	LSTM	MSE loss	-	-	-	-	-	NGSIM I-80 and US-101: training:70%, validation:10%, testing:20%
[138] 2022	Position, Velocity, Orientation, Map	-	GRU,GAT,	LSTM	-	-	LeakyReLU	-	-	-	INTERACTION: train:425,192, validation: 104,627 data pieces; NGSIM I-80 and US-101:63,176 data pieces,training:53,176, validation: 10,000
[126] 2022	Coordinates, Map	2 s	Transformer layer, Graph isomorphism network	Three-layer MLP	Smooth L1 loss, Hinge loss	Adam optimizer	-	Initial learning rate:0.001	Epochs:150; Batch size:128	-	Argoverse
[221] 2022	Coordinates, shape	NGSIM: 3 s	ConvGRU, GAT	ConvGRU	Masked mean squared error	Adam optimizer	-	Learning rate:0.001	-	PyTorch	Argoverse; NGSIM I-80 and US-101; ApolloScape
[131] 2023	Coordinates, Velocity, Acceleration, Yaw Rate, map	2 s	LSTM, GRU, Graph attention network	MLP	L1 norm, NLL	Adam optimizer	LeakyReLU, softmax function	Learning rate:0.0001	-	PyTorch	NuScenes:1000 scenes, 20 s long

"- This information is not available/applicable for work".

in a scene. A directed edge-featured heterogeneous graph was used to represent inter-agent interactions in traffic, and a gate mechanism was added for selective map sharing among target agents. Liu et al. [126] proposed a multi-agent, multi-modal trajectory prediction framework using Graph Attention Isomorphism Networks (GAIN), which consisted of three attention blocks. AI-TP was introduced in [221] to forecast multiple SV trajectories using GAT for interaction information, followed by two convolutional Gated Recurrent Units (GRU) networks. Zhang et al. suggested the Gatformer model in [220] for predicting future movements of nearby traffic agents while considering spatial-temporal connections, using graphs and GAT to capture environmental interactions, and integrating the transformer encoder-decoder. In [131], a two-layer GAT was used for information aggregation and node correlation explanation, with a multi-head attention mechanism to project surrounding states to the graph and explain interactions between vehicles and the traffic flow state. In summary, GATs enable the model to attend to relevant nodes (e.g., vehicles, pedestrians) in the graph, assigning different weights to capture the importance of each node's features for predicting the trajectory of a specific object. However, The performance of GATs heavily depends on the quality and representation of the graph structure. Designing an appropriate graph representation and considering the selection of nodes and edges is crucial for achieving optimal

Table 17
Summary of Graph Convolution Network-based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon, Advantages and Limitation of each work, Summary of each work, and Evaluation Metric (EM).

Work	# trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[120] 2019	1	5 s	The relationships between various traffic agents are described by using fixed graph.	When employed in urban traffic settings, it could see some performance decrease.	Extracts features using several graph convolutional blocks.	RMSE
[119] 2019	1	5 s	The relationships between various traffic agents are described using fixed and dynamic graphs.	Incorporate GRIP++ into a perception and route planning module to further assess overall performance.	Extracts features using several graph convolutional blocks.	RMSE, WSADE, WSFDE
[95] 2020	1	4 s	EGCN-based interaction embedding analyses the inter-vehicle interactions inherently.	Model cannot take into account the road structures.	Model is fully scalable to handle any number of vehicles in driving scenario.	RMSE
[26] 2020	1	5 s	1. The interactions between any two road agents are represented by the adjacency matrix; 2. There is no assumptions regarding the dimensions and geometry of the road-agents.	Training is slow for computing the traffic-graphs.	A two-layer GNN-LSTM structure is used to resolve the trajectory prediction task.	ADE,FDE
[231] 2020	1	5 s	To forecast future trajectory, a model can combine the information gathered from both non-Euclidean and Euclidean domains.	When constructing the topology of the communication network, the time information was not taken into account.	GCN network allows vehicles to communicate information to take into account how the surrounding vehicles are changing and adapting to the environment.	RMSE
[164] 2022	1	5 s	Adjacency matrix used to describe the intensities of mutual influence between vehicles.	The impact of neighbouring automobiles from various angles are ignored.	This network uses a GCN to address spatial interactions and a CNN to capture temporal data.	RMSE
[208] 2022	1	5 s	A matrix that may be adjusted and supplemented to make up for the shortcomings of extracting characteristics from fixed topology	Spatial and temporal features are not processed simultaneously.	The stacked GCN module extracts the global spatiotemporal characteristics of the historical vehicle trajectory data.	RMSE,ADE, FDE
[209] 2023	1	5 s	By describing various logical correlations of numerous road agents, a multi-view logical network is proposed.	It is a difficult challenge to confidently estimate the trajectory based on the missing data and noisy samples.	The GCN module mines the logical-physical properties at the micro level.	WSADE, WFADE

Table 18
Summary of Graph Attention Network-based methods: Related Work & Year, No. of outputs, Prediction Horizon (PH), Advantages and Limitation of each work, Summary of each work, and Evaluation Metric (EM).

Work	# trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[139] 2020	1	5 s	Simulate the diverse interactions between infrastructures and vehicles.	A more detailed HD map not included rather than a picture-based one for interaction with the environment.	A heterogeneous digraph is used by GAT to derive spatial interaction information.	ADE,FDE
[50] 2021	1	5 s	On the scene graph, two types of relation edges are created separately to describe various affects that cars and places have on one another.	The model is not suitable for problems with unbalanced data learning.	A dual GAT is used to simultaneously describe the space attraction and vehicle-wise repulsion.	RMSE
[138] 2022	1	8 s	A gate mechanism allowing for the selective sharing of maps among all target agents.	Hardware configurations in practical applications will have an impact on the data's quality and availability.	Inter-agent interaction modelling with a heterogeneous edge-enhanced GAN.	RMSE,ADE, FDE
[126] 2022	1,5	3 s	1. adjacency matrix outlining agent connection; 2.vector representation method for the map information.	1. dynamic features are excluded; 2. Prediction accuracy can be further improved.	Graph Isomorphism Network has been used to update the node feature and include information from nearby nodes.	minADE, minFDE
[221] 2022	1	5 s	1. Forecasts agent speeds rather than spatial coordinates; 2. Asymmetrical adjacency matrix depicts the relationships between traffic agents.	There are no rasterized maps or LiDAR data included.	These models display traffic scenes as graphs and GAT to explicitly record the interactions from the environment.	RMSE, WSADE, WFADE
[220] 2022	1	5 s	Accurately simulating spatial-temporal interactions from the environment.	Model fixed the maximum number of traffic agents that it could handle.	With the use of attention mechanisms, the GAT block is used to represent the interactions between agents and infrastructure.	ADE,FDE
[131] 2023	1,6,10	6 s	The introduction of an ice and snow mask system that simulates situations where lane lines are covered.	It is crucial to the accuracy of the absolute forecast that the proportions of different input data be considered.	Information gathering is made possible via a two-layer GAT, which also explains node correlation.	minADE, minFDE, MR

results. Table 18 summarizes the GAN-based approaches for trajectory prediction, highlighting the number of trajectories predicted and the prediction horizon measured in seconds (s). The table also provides insights into the strengths and weaknesses of each study, along with the evaluation metrics used for training and testing.

4.2.2.3. Other graph neural network High Definition (HD) maps play a crucial role in trajectory prediction for autonomous vehicles. HD maps provide detailed information about the road network, including lane markings, traffic signals, and road boundaries, which can help predict the future trajectory of a vehicle or pedestrian more accurately. Initially, Benz et al. [236] utilized HD maps for predicting trajectories, followed by determining the vehicle's future trajectory along the lane based on map topology using related lane information. However, this method did not consider interaction-related factors. To improve trajectory prediction accuracy, researchers have incorporated GNN to

Table 19
Summary of other Graph Neural Network-based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon (PH), Advantages and Limitation of each work, Summary of each work, and Evaluation Metric (EM).

Ref.	# Trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[58] 2020	1	3 s	Vectorized representation of the HD map and agent dynamics.	Recalculating the VectorNet features for every target would increase the computational cost with the number of targets being predicted.	Hierarchical GNN, where the first level takes advantage of the spatial proximity of certain road elements and the second level simulates the high-order interactions between all elements.	ADE
[121] 2021	1,6	3 s	1. Create a lane graph using the raw map data; 2. exploit all four different lane-agent interactions.	Case of extreme acceleration does not captures well by the model.	1D CNN to handle the input trajectory data and uses along-lane dilation and numerous adjacency matrices to extend graph convolutions to provide the map features.	minADE, minFDE, MR
[227] 2021	1,6	3 s	Based on the provided scene context, TNT can model the scene context using any acceptable context encoder.	Forecasting over a medium time horizon.	There are three stages to TNT: target prediction, target-conditioned motion estimation and scoring and selection.	minADE, minFDE,MR
[69] 2021	1,6	3 s	Provide an offline optimisation-based method to supply our final online model with several future pseudo-labels.	The model is trained in urban datasets only.	Estimates dense target candidate probabilities without using heuristic anchoring.	minADE, minFDE,MR
[217] 2021	1,6	3 s	The actor-to-actor and actor-to-map relations are distributedly and map-awarely captured by LaneRCNN.	Predict only single vehicle future trajectories.	To encode each actor's previous motions and the topology of the local map, learn a local lane graph representation for each actor.	minADE, minFDE,MR

capture interaction features between vehicles and maps as well as between vehicles, following the introduction of the Argoverse dataset [27] with vector maps. Gao et al. [58] proposed VectorNet, a GNN-based system that employs nodes to represent both the vector maps and vehicles in the scene for trajectory prediction. Liang et al. [121] integrated CNN-extracted vehicle features and GCN-extracted lane features from vector maps for trajectory prediction. Zhao et al. [227] presented a target-driven method called target-driven trajectory prediction (TNT) that selects sparse goal anchors and the optimal route to the target using VectorNet-extracted map features. DenseTNT [69] outperforms TNT in performance by evaluating dense goal candidates. Zeng et al. [217] utilized LaneRCNN to represent local lane maps and interaction modules to account for interaction factors between participants' historical trajectories and local map topology. Researchers are exploring ways to integrate multiple sources of information, including HD maps, sensor data, and machine learning algorithms, to improve the accuracy and robustness of trajectory prediction for autonomous vehicles. Table 19 provides a summary of other graph neural network-based approaches for trajectory prediction, focusing on the number of trajectories predicted and the prediction horizon measured in seconds (s). The table also provides the strengths and weaknesses of each study, along with the evaluation metrics used for training and testing.

4.3. Combination of CNNs and RNNs

Several researchers have proposed models that use a combination of RNN and CNN to handle temporal and spatial information for trajectory prediction. For instance, Deo et al. [43] use an LSTM encoder to extract temporal data from nearby vehicles, which is then fed into a social pooling layer that collects interaction-related parameters between vehicles. A social tensor is created and fed into a collection of CNNs to determine the spatial correlation of vehicles. MATF [230] introduces a fully convolutional network that resembles a U-net [233] for Multi-Agent Tensor Fusion (MATF) encoding and decoding. The fused vectors of each vehicle are taken from the output layer of the U-net [233]-like network, added to the LSTM-encoded vectors of the vehicles' dynamics, and then supplied to LSTM decoders. Schreiber et al. [159] use a CNN on condensed BEV images and an Encoder-Decoder LSTM to learn the temporal dynamics of the input data. TraPHic [25] uses a CNN-LSTM hybrid network to derive features from the state and nearby objects of the primary vehicle. Xie et al. [202] use a "box" to find and remove outliers in the vehicle's trajectory and extract interaction-aware features by feeding them into a convolutional layer and a maximum pooling layer. Xu et al. [210] propose a model that uses a convolutional network and a graph operation layer to capture spatiotemporal features and an LSTM encoder-decoder to forecast the traffic-related future trajectories of multiple vehicles. Table 20 presents a summary of CNN+RNN-based approaches for trajectory prediction, emphasizing on the number of trajectories predicted and the prediction horizon measured in seconds (s). The table also highlights the strengths and weaknesses of each study including evaluation metrics used for training and testing. Beyond evaluating input modalities and history length, Table 21 compares encoding and decoding procedures in the CNN+RNN-based methods. It differentiates methods based on optimisation approaches, loss functions, activation functions, learning rates, additional training parameters, and training platforms. Moreover, the table provides sample numbers for each method in the training, validation, and test datasets.

4.4. Generative model

Predicting multi-modal trajectories presents challenges and uncertainties due to the potential diversity of outcomes. To address this issue, some researchers have turned to generative models to create multi-modal trajectories that can capture the underlying diversity. However, in order for a multi-modal trajectory prediction model to be effective, its output distribution must meet certain requirements, including diversity, social acceptability, and controllability. Achieving an optimal distribution using only one ground truth can be difficult and may lead to less diverse and unacceptable predictions. To overcome this challenge, Generative Adversarial Networks (GANs) and Variational Auto Encoders (VAEs) have been proposed as solutions. Fig. 12 illustrates the involvement of research papers, depict in percentages, of both generative models in assisting Autonomous Vehicles (AVs) with the task of trajectory prediction. Both models contribute approximately equally to the prediction process, showcasing their shared responsibility in generating accurate trajectory predictions.

4.4.1. Generative adversarial network

In trajectory prediction tasks, Generative Adversarial Networks (GANs) are used to generate realistic trajectories based on the input data. The generator takes in the historical trajectory data as input and generates a future trajectory, while the discriminator evaluates the generated trajectory for realism. The generator is trained to improve the realism of the generated trajectories by fooling the discriminator into

Table 20
Summary of CNN+RNN based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon (PH), Advantages and Limitation of each work, Summary of each work, and Evaluation Metric (EM) - Predicting any number of trajectories for the work.

Ref.	# Trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[43] 2018	-	5 s	For effectively learning interdependencies in vehicle motion, convolutional social pooling is an enhancement over social pooling layers.	It depends on information from vehicle tracks and ignores visual and map-based cues for predicting manoeuvre classes and future trajectories.	On each vehicle trajectory, LSTM is used. The outcome is displayed in a BEV grid structure before being fed to a CNN. A total of six LSTM decoders receive the output.	RMS, NLL
[230] 2019	1	5 s	1. The use of convolutional fusion enables the modelling of interactions among multiple agents; 2. The incorporation of adversarial loss facilitates stochastic prediction learning.	Despite the limited agent-scene interactions and straight road lanes in the dataset, our model did not surpass NGSIM.	A concatenated vector comprising an agent's movement and a static scene encoded by a CNN and LSTM.	RMSE
[159] 2019	-	2 s	It created a recurrent skip architecture to handle missing input data.	Only a stationary sensor records the data.	The spatial features are first extracted from the input image using a convolutional network. These features are then fed as input to the encoder-decoder LSTM.	F1-scores
[25] 2019	-	5 s	Forecasting the trajectories of road agents in busy traffic footage.	1. Dense diverse traffic serves as a model design inspiration; 2. Understanding the diverse sizes and shapes of road agents is necessary for simulating heterogeneous constraints.	A hybrid LSTM-CNN network is used to model the horizon-based and heterogeneous-based weighted interactions between road agents.	RMSE, ADE, FDE
[202] 2020	1	30 s	1. The box plot is used to identify and get rid of anomalous vehicle trajectory values; 2. The model's hyper-parameters are optimised using a grid search approach.	Consider the historical information of surrounding vehicles.	Convolutional and maximum pooling layers in the CNN-LSTM framework extract interaction-aware features, while an LSTM and a fully connected layer is used for prediction.	RMSE, MAE, and deviation
[210] 2022	1	5 s	The interaction of cars is depicted using a traffic graph; 2. The prediction mechanism takes into account road and speed parameters.	Future trajectories are not thought to be affected by the diversity of traffic agents, traffic regulations, or climate change.	The temporal characteristics of a vehicle are captured using a convolutional layer, spatial features are captured using graph operation layer and LSTM encoder-decoder to predict the future trajectories.	ADE,FDE

Table 21
Summary of CNN+RNN based methods based on design and implementation parameters.

Work	Input	Perceive (second(s))	Encoding	Decoding	Loss function	Optimiser	Activation function	Learning rate	Other parameters	Platform	Data
[230] 2019	Coordinates, Map	3 s	LSTM, CNN, Fully convolutional layers	LSTM, MLP	L2 or L1 loss, Adversarial Loss;	Adam optimiser	ReLU activations, Leaky ReLU, Sigmoid activation	Initial learning rate: 0.001	Dropout ratio:0.3(training); Batch size of 32	PyTorch	NGSIM US-101; contains around 6k vehicles in total; train, validation
[159] 2019	Occupancy, Velocity, Map	2 s	ConvLSTMs	ConvLSTMs	Static loss, Dynamic loss	ADAM optimiser	-	Learning rate: 0.0001	Exponential decay rates $\beta 1$: 0.9 and $\beta 2$: 0.999; training process is stopped after one epoch	-	Data is only recorded by a stationary sensor in urban intersection; training set:80%, test set: 20%
[25] 2019	Coordinates, Map	3 s	Single-layer LSTMs: hidden state dimensions of 64, CNN	Single-layer LSTMs with hidden state dimensions:128	Negative log-likelihood loss	Adam optimizer	ELU non-linearity	Learning rate:0.001	Batch size:128; 16 epochs; use radius of 2 meters to define the neighbourhood	Pytorch: single TiTan Xp GPU	NGSIM, KITTI
[202] 2020	Speed, Distance, Offset	-	CNN-LSTM		MAE loss	Adam optimiser	Sigmoid activation	-	Grid search algorithm, Batch size:100	-	NGSIM US-101; train, test
[210] 2022	Coordinates, Speed, Acceleration, Class	3 s	LSTM, Graph convolution	LSTM	Adaptive weight loss function	Adam optimiser	LeakyReLU activation	Learning rate:0.0001	Dropout:0.3	Python 3.8, PyTorch 1.8	NGSIM I-80: test:166,400 data, US101; HighD: test:150,016 data

"- This information is not available/applicable for work".

Table 22
Summary of Generative Adversarial Network-based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon (PH), Advantages and Limitation of each work, Summary of each work, and Evaluation Metric (EM) - Predicting any number of trajectories for the work / This information is not available for work.

Ref.	# Trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[77] 2020	-	4.8 s	1. The vehicle interactions are handled by the pooling mechanism; 2. based on the vehicle's past performance to forecast its upcoming course.	The model requires processing of the images before sending them to the GAN, so it cannot operate in real-time.	Future paths are generated by the generator, which is composed of an encoder and decoder network and a pooling module. The discriminator, which consists of an encoder, can classify the trajectories more precisely as genuine or fake.	ADE,FDE
[229] 2020	1	30 s	1. The conversion of each vehicle's position coordinates into normalised coordinates; 2. Based on the psychology of the driver, the vehicle turning model can improve the driving path.	1. Does not include the interaction between the vehicles and road information; 2. Consider only the urban road scenarios.	Using past trajectory data, GAN is used to train and understand the driver's behaviour.	MAE, RMSE, average accuracy
[118] 2021	1	6 s	Prediction models that use rules as inductive biases.	Fail to take trajectory uncertainty and rule prioritisation into account.	Signal Temporal Logic is viewed as a collection of discriminator features and a generator auxiliary loss.	ADE,FDE, MaxDist
[71] 2023	1,6	3 s	To allow the global map to be reused, a graph query mechanism is presented.	The potential of HD maps for trajectory prediction tasks has to be investigated.	The generator side, the proposed model creates contextual features by fusing vehicle motion with high-definition maps. The addition of the map enhances the discriminator's basis for making judgments about the resulting trajectories.	MinADE, MinFDE, MR,DAC
[189] 2020	-	5 s	1. To record multi-vehicle interactions in both spatial and temporal dimensions, two parallel fusion modules have been constructed; 2. Under various conditions, display the effects of nearby vehicles on trajectory prediction.	1. Separately consider the spatial and temporal features for prediction; 2. Consider only the historical information of multi-vehicles.	The generative adversarial network is used to handle the agent motion behaviour's innate multi-modal properties.	RMSE
[186] 2021	-	5 s	1. Considering the precise spatial distributions of agents during movement; 2. Multiple trajectory sequences are used to capture socially temporal relationships; 3. To formally describe the temporal correlations between interactions, use the social recurrent mechanism.	Only historical trajectories for the scenario's observed agents are taken; scene information for future trajectories is not included.	GAN to produce multi-modal trajectory distribution.	ADE,FDE

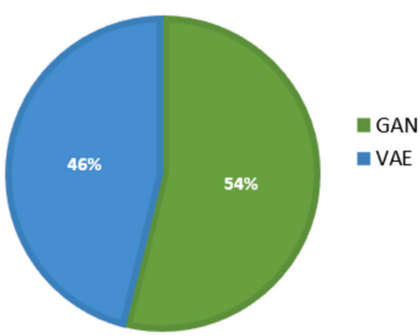

Fig. 12. Participation of Research articles in trajectory prediction task using Generative models.

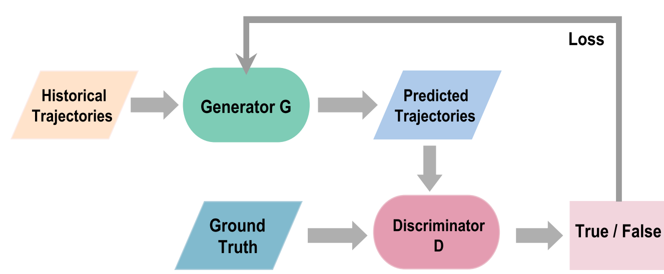

Fig. 13. Depiction of Generative Adversarial network for trajectory prediction task.

believing they are real. This methodology is shown in Fig. 13. GAN was introduced by Ian Goodfellow in 2014 [183]. When GANs are used for trajectory prediction, the discriminator assesses the accuracy of the predicted trajectory while the generator constructs it. In [77], Hegde et al. forecast vehicle trajectories using the vehicle's coordinate information. The TS-GAN model presented by Wang et al. [189] utilizes a self-developed convolutional social mechanism and a recurrent social mechanism to extract vehicle spatial and temporal information from the GAN network. To create model-based multi-modal trajectories, Song et al. [168] employ vector maps and vehicle status information and apply a learning-based discriminator to extract information about vehicle interactions for providing the best trajectories. In [229], the GAN-VEEP model is proposed for short-term vehicle trajectory prediction, utilizing a vehicle coordinate normalisation model to convert position coordinates into normalized coordinates. In [118], two strategies for incorporating Signal Temporal Logic (STL) rules into a GAN-style trajectory predictor are presented. In [186], the STSF-Net framework is proposed, which utilizes a GAN for multi-modal trajectory distribution, with a generator that has an LSTM encode-decoder framework with a 3D CNN network for temporal correlations modelling and a discriminator that uses a Multi-Layer Perceptron (MLP) to identify the true trajectory. Additionally, Guo et al. [71] suggests using a map-enhanced GAN for trajectory prediction by fusing vehicle motion with high-definition (HD) maps to create contextual features. Table 22 presents a summary of GAN-based approaches for trajectory prediction, focusing on the number of trajectories predicted and the prediction horizon measured in seconds (s). The table also provides the strengths and weaknesses of each study and highlights the evaluation metrics used for training and

Table 23
Summary of Generative Adversarial Network-based methods with design and implementation parameters.

Work	Input	Perceive (second(s))	Generator	Discriminator	Loss function	Optimiser	Activation function/learning rate	Other parameters	Experimental platform	Data
[77] 2020	Coordinates	-	LSTM encoder, LSTM decoder	LSTM encoder, MLP		-	Learning rate 0.001.	Batch size:64; Epoch:200	Python 3.6, TensorFlow,PyTorch	VisDrone:6471 for training, 548 for validation and 1580 for testing
[229] 2020	Coordinate, Speed, Future-goal	-	Depth neural network	Two depth neural network with three hidden layers	Maxlog(D)	ADADELTA; policy gradient	Learning rate of the generative model: 0.02, Discrimination model: 0.01.		Intel Core i5-4210, 2.40-GHz CPUs; Python 3.5, TensorFlow 1.12	SUMO simulator; train:80%, test:20%
[189] 2020	Coordinates	3 s	LSTM encoder; LSTM decoder; Auto-encoder social convolution module; Recurrent social module	LSTM decoder	Min-max loss, L2 loss, Reconstruction loss	Adam optimizer	Learning rate:0.001; ReLU activation	Batch size:64; epochs: 50	-	NGSIM US-101,I-100: 5922867 training,859769 validate, and 1505756 test data
[118] 2021	Coordinates, Heading, Noise term	2 s	Multi-modal trajectory predictor (MTP)	LSTM encoder, CNN	Auxilary loss; standard min-max loss	Stochastic gradient optimizers	-	Batch size:32; Epochs:20	4 NVidia Tesla V100 GPUs	NuScenes; train:20000 trajectories, validation:4000 trajectories.
[186] 2021	Coordinates	3 s	LSTM encoder; 2D-CNN; 3D-CNN; LSTM decoder	LSTM encoder	Min-max loss,MSE loss	Adam optimizer	Initial learning rate:0.001;ReLU nonlinearity	Epochs:100; Batch size:64	Nvidia Titan V GPU; pytorch	NGSIM US-101, I-80
[71] 2023	Coordinates, Map	2 s	Temporal encoding modules:LSTM,GRU; embedding modules:linear,convolutional layers; map extraction: GNN, attention mechanisms	LSTM, MLP	-	Adam optimiser	ReLU activation; Learning rate: 0.0001	-	Six Xeon Gold 6142 CPUs and an NVIDIA RTX3090 GPU; PyTorch	Argoverse: 324,557 trajectory; NuScenes

"- This information is not available/applicable for work".

testing. Table 23 assesses input modalities and their observed history length (in second(s)). It compares different methods based on generator and discriminator techniques in the model. The table also contrasts various factors such as loss functions, optimisation methods, activation functions/learning rates, and other parameters used during model training, considering the training platform. Additionally, it lists the number of samples in the train, validation, and test datasets for each method.

4.4.2. Variational autoencoder

The Auto Encoder (AE) compresses data using an Encoder and decodes it with a Decoder to produce a reconstructed output with minimal reconstruction errors. However, AE has been criticized for merely "memorizing" data and having limited data generation capacity. In contrast, the Variational Autoencoder (VAE) has a generative capability that spans the entire space, and it addresses the issue of non-regularized latent space in autoencoders. VAE aims to minimize both reconstruction loss and similarity loss. Bhattacharyya et al. [20] proposes the use of a Conditional Variational Autoencoder (CVAE) for structured prediction tasks. Cho et al. [34] suggested using CVAE and LSTM to estimate possible future positions of vehicles. To ensure compliance with traffic laws and social navigation principles, they also utilized Signal Temporal Logic (STL) to eliminate irrational scenarios. Hu et al. [88] proposed a multi-modal trajectory prediction framework based on CVAE, but it only considered situations where two vehicles were involved. Zhang et al. [219] proposed using Stacked Sparse AutoEncoders (SSAE) to handle a high-dimensional input vector with motion and interaction data in a multi-modal scenario. Sriram et al. [170] presented an architecture that predicts the multi-modal trajectory of all traffic participants simultaneously using Convolutional LSTM and CVAE for scene context feature extraction and trajectory prediction, respectively. Dulian and Murray [53] utilized CNN networks to extract spatial information from Bird's Eye View (BEV) images of an HD-Map and used a CVAE to predict future trajectories, sampling the conditional variable from a prior distribution during the testing phase. Liu et al. [125] developed a CVAE-based model to generate potential trajectories while considering motion uncertainty and then created a driving risk map. They also developed a probability model based on the trajectory risk value and used a random selection method to produce a unifying rendering of the scene's traffic agents' interactions. Based on the findings of these studies, CVAEs can take into account various conditions such as the current state of the vehicle, surrounding traffic, road conditions, or any other relevant contextual information. These conditions can be encoded as additional inputs to the CVAE model, which then learns to generate future trajectories conditioned on these inputs. Additionally, the performance of CVAEs heavily relies on the effectiveness of the chosen conditioning inputs. Table 24 summarizes the Variational Autoencoder-based approaches for trajectory prediction, highlighting the number of trajectories predicted and the prediction horizon measured in seconds (s). The table also provides insights into the strengths and weaknesses of each study, along with the evaluation metrics used for training and testing. Furthermore, Table 25 assesses input modalities and their observed history length. It compares different methods based on generator and discriminator techniques in the model. Additionally, the table contrasts various factors such as loss functions, optimisation methods, activation functions/learning rates, and other parameters used during model training, considering the training platform. It also presents the number of samples in the train, validation, and test datasets for each method.

Table 24
Summary of Variational Auto Encoder-based methods: Related Work & Year, No. of Predicted Trajectories for each vehicle, Prediction Horizon (PH), Advantages and Limitation of each work, Summary of each work, and Evaluation Metric (EM) - Predicting any number of trajectories for the work / This information is not available for work.

Ref.	# Trajectories	PH	Advantages	Limitation	Summary of Prediction Method	EM
[34] 2019	-	5 s	Jointly reason about future vehicle trajectories as well as the degree to which each rule is satisfied.	Supposing that the control of the ego vehicle is impacted by six nearby vehicles in adjacent lanes.	Using the prior trajectory and feature representation made up of lane deviation distance and distances to nearby vehicles, CVAE uses these to learn a distribution of future trajectories.	ADE
[88] 2019	1,3	-	The ability to connect the model to the underlying motion pattern makes it interpretable.	The prediction system is tuned for roundabout situations.	Using historical scene details and driving intentions as conditional inputs in the model structure, the prediction of joint trajectories of two cars.	MSE,NLL
[219] 2020	1,10	5 s	The spatial interactions of vehicles are modelled by dilated convolutional social pooling.	Vehicle statuses are the only components of the multimodal information input.	SSAE can automatically extract deep and high-level features from input data, which reflects the interaction between different states of vehicles.	RMSE
[170] 2020	1,5	5 s	Taking into account scene semantics and inter-agent interactions, with constant-time inference regardless of the agent count.		Using CVAE to predict data diversity for each type of agent.	ADE, FDE,NLL
[53] 2021	-	6 s	1. Instead of being limited to generating a finite number of deterministic trajectories, generate an endless number of varied motion samples; 2. show benefits of using the Minimum over N (MoN) cost function.	On-road participants' social interactions are excluded.	Constrained by an agent's previous mobility and a rasterized scene context encoded with the Capsule Network.	minADE, minFDE
[125] 2022	1	5 s	Considering the drivers' unknown trajectory intentions given the driving risk map.	Predict the future path for vehicles only.	Using ground truth and historical vehicle trajectories, CVAE based on GRU will produce candidate trajectories.	ADE,FDE

Table 25
Summary of Variational Auto Encoder-based methods with design and implementation parameters.

Work	Input	Perceive (second(s))	Encoding function	Decoding function	Loss function	Optimiser	Activation function/learning rate	Other parameters	Experimental platform	Data
[219] 2020	Coordinates, Speed, Acceleration, Heading angle, Lane ID, Type, Size	3 s	Stacked sparse auto-Encoder; Dilated convolutional social pooling layer	LSTM	MSE loss, Negative log-likelihood loss	Adam optimiser	Learning rate:0.001; LeakyReLU	-	PyTorch	NGSIM US-101 and I-80: train:80%, test:20%
[170] 2020	Coordinates, Map	5 s	LSTM,MLP	ConvLSTM layer	KL Divergence Loss, Mean absolute loss	Adam optimize	Learning rate:0.008	10 epochs	NVIDIA RTX 2080Ti GPU;Tensorflow 2.0	Argoverse: 333,441 trajectories
[53] 2021	-	2 s	MLP	MLP	Reconstruction loss, KL-divergence loss	SPS optimizer	Adaptive learning rate; Leaky ReLU, Squashing non-linearity	360 epochs; Batch size:64	Single Nvidia RTX 2080Ti; PyTorch	NuScenes; train/val/test: 20965/5000/5956
[125] 2022	Coordinates, Velocity, Acceleration, Map	1 s	GRU	GRU	KL-divergences, L2 norm	Stochastic gradient descent optimiser	ReLU activation function	-	Intel (R) Core (TM) I7-8700 CPU @ 3.20 GHz processor and a NVIDIS 1050Ti GPU; C++	HighD; train:80%, test:20%

"- This information is not available/applicable for work".

Majority of researchers utilized deep learning-based methods to tackle spatial and temporal prediction issues, resulting in cutting-edge outcomes. Their strategy accommodates diverse driving scenarios like roundabouts, highways, or intersections, and the algorithm accounts for lower-order interactions among the surrounding vehicles (SVs). Despite their proficiency in producing rational trajectory predictions, many trajectory prediction algorithms are complex and often lack explainability due to their black-box nature. To tackle the complexity and enhance explainability in trajectory prediction algorithms, one solution is to explore Explainable AI models or other techniques like model pruning,

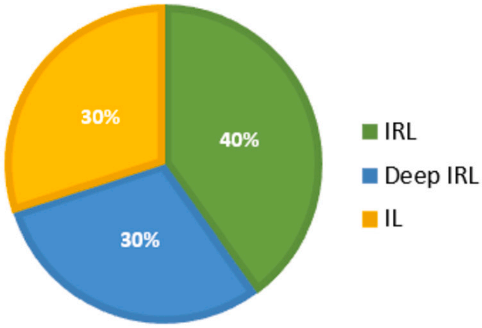

Fig. 14. Participation of Research articles in trajectory prediction task using reinforcement learning-based approaches.

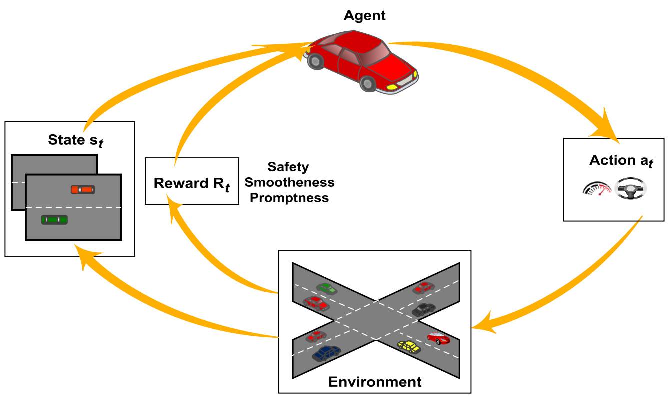

Fig. 15. Depiction of the Reinforcement-learning based method.

quantisation, and distillation. Presently, an escalating number of autonomous vehicle trials are adopting deep learning-based methods to forecast the future trajectories of traffic participants.

5. Reinforcement learning-based methods

Recent years have seen the rapid growth of Reinforcement Learning (RL), which offers a new method for comprehending high-dimensional complex policies [80]. It offers innovative solutions for Autonomous Vehicles (AVs)' challenges involving trajectory prediction [106,55]. The Markov Decision Process (MDP) is typically utilised when RL is used to AVs trajectory prediction to maximise the projected cumulative reward.

RL techniques are utilized to estimate the underlying cost function or directly identify the optimal policy for trajectory prediction. In either approach, it is assumed that the observed agent always seeks to reach its objective by utilizing the optimal policy based on a specific cost function. Fig. 15 illustrates the application of RL methods in AVs. Within the framework of MDP, RL-based methods can be categorized into Inverse Reinforcement Learning (IRL) methods, Imitation Learning (IL) methods, and Deep IRL methods, as explained in the following sections. Fig. 14 illustrates the distribution of research articles, expressed as percentages, for different variants of RL and their involvement in addressing the trajectory prediction task in AVs.

5.1. Inverse reinforcement learning

The main idea behind Inverse Reinforcement Learning (IRL) is to learn the reward function that explains the observed behaviour of the agents. Instead of directly imitating the observed trajectories, IRL aims to understand the underlying motivations or objectives that drive those trajectories. By inferring the reward function, the algorithm can generalize beyond the observed trajectories and make predictions about future trajectories. Manually specifying the weight of the reward function is inappropriate due to the complex nature of driver behaviour, according to Wang et al. [184] and Guan et al. [70]. To address this issue, IRL learns the optimal driving policy by inferring the reward function based on expert demonstrations (trajectories), as depicted in Fig. 16. Liting et al. [67] utilize a spatiotemporal state lattice to describe driver behaviour based on expert demonstrations. The driving manoeuvres create a distribution for upcoming trajectories [172]. Interaction-related elements are considered to achieve probabilistic prediction for AVs. DriveIRL, presented by Tung et al. [148], is the first learning-based planner that uses IRL to control a vehicle in congested urban traffic. They build an architecture divided into ego trajectory generation, checking, and scoring, using simple and reliable techniques to solve the very complex problem of ego trajectory generation.

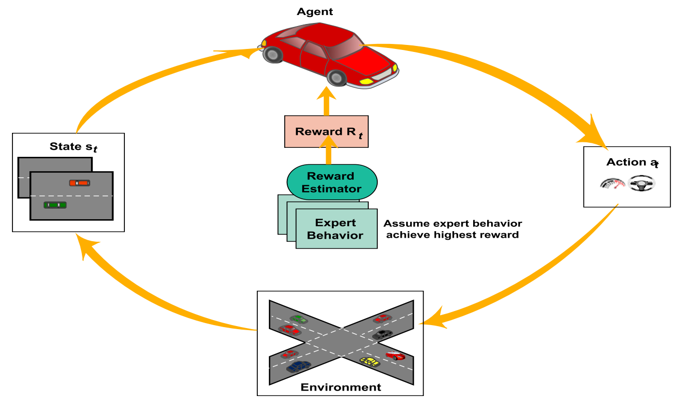

Fig. 16. Depiction of the Inverse Reinforcement-learning based method.

A significant challenge in IRL is that an optimal policy may be ambiguous, since it can result from multiple reward functions [6]. Because of this, a modified algorithm called Maximum Entropy IRL (MaxEntIRL) was developed by Ziebart et al. [235]. The MaxEntIRL algorithm aims to resolve the ambiguity in IRL by maximizing the entropy across the distributions of potential state-action pairs for a learned policy. Some MaxEnt-IRL techniques use sampled trajectories to carry out prediction tasks. Xu et al. [207] sample candidate trajectories with the lowest cost that will be selected as the anticipated trajectory. Wu et al. [198] propose a method for learning reward functions in the continuous domain by estimating the partition function using the speed profile sampler. State sequences from the MaxEnt policy are sampled in [45] and provided to an attention-based trajectory generator to produce valuable future trajectories. To estimate the best policy while reducing computing costs, Xin et al. [204] utilize randomly pre-sampled policies in sub-spaces. Yifei et al. [211] propose an Inverse Optimum Control (IOC) method utilizing Langevin Sampling to determine the cost function of other vehicles in an energy-based generative model. In summary, while IRL has the potential to provide deeper insights and more flexible trajectory predictions, the requirement for expert demonstrations and the challenges associated with their quality and computational complexity should be carefully considered in practical applications. Table 26 offers a detailed summary of models based on parameters like state, action space, reward function, optimisation approach, evaluation metrics, and expert demonstration.

5.2. Deep inverse reinforcement learning

Deep Inverse Reinforcement Learning (Deep IRL) is an extension of Inverse Reinforcement Learning (IRL) that incorporates Deep Neural Networks (DNNs) to learn the reward function from expert demonstrations. The deep IRL framework is introduced in [199] to approximate complex and nonlinear reward functions. To approximate rewards, this article uses a fully Convolutional Neural Network (CCN). You et al. [215] consider driving behaviour and road geometry, constructing the MDP first using RL, learning the best driving strategy using IRL, and

Table 26
Summary of Inverse Reinforcement Learning Methods with design and implementation parameters.

Work	State Space	Action Space	Reward Function	Optimisation approach	Evaluation metrics	Demonstrations available
[211] 2019	Coordinates, Steering, Speed, Map	Steering, Acceleration	Non-Markov cost function	Inverse-dynamics optimisation using gradient descent	RMSE	The number of expert trajectories are 44
[207] 2020	Coordinates, Velocity, Acceleration	Human driving trajectory	Lane Incentive Cost	L-BFGS algorithm	Distance metric	Yes
[198] 2020	Speed, Acceleration, Jerk	Distance, Interaction distance	Linear-structured reward function	Gradient decent	Feature deviation; mean euclidean distance; Probabilistic Metrics	Yes
[45] 2020	Path states, Goal state	Up, down, left, right, end	Path reward, Goal reward function	MaxEnt policy	MinADE	Yes
[148] 2022	Speed, Acceleration, Steering, Map, Route	Coordinates, Heading, Speed	-	Adam optimizer with an initial learning rate of 0.001	computation, categories	Human driving vehicle in Las Vegas

approximating the reward function using DNN. In [237], driving behaviour is represented by Deep IRL utilizing camera images, while CNN extracts the corresponding state information. Zhu et al. [234] encode the vehicle's kinematics using RL ConvNet and State Visiting Frequency (SVF) ConvNet by back-propagating the loss gradient [200] between expert SVF from expert demonstration and policy SVF from LiDAR data. Jung et al. [98] using neural LSTM to extract the feature map from the LiDAR and trajectory data, which will then be merged into the output reward map to forecast the traversability map. In [33], a fused dilated convolution module is proposed to improve the extraction of raster features. Subsequently, a reward update policy with inferred goals is enhanced by learning the state rewards of goals and pathways individually instead of the original complex rewards, which can reduce the need for preset goal states. In summary, Deep IRL offers the potential for more powerful and adaptive trajectory prediction models by leveraging deep neural networks. However, challenges related to data requirements, computational complexity, interpretability, and overfitting need to be carefully addressed for successful application in trajectory prediction for autonomous driving. Table 27 summarizes models based on design and implementation parameters.

5.3. Imitation learning

One disadvantage of Inverse Reinforcement Learning (IRL) algorithms is their difficulty in training with scenarios where there are few rewards or no direct reward function. To address this issue, Imitational Learning (IL) has been suggested as a solution. IL aims to quickly determine a policy based on an expert's observation without requiring a cost function. One of the pioneering methods in imitation learning for autonomous driving is ALVINN, developed by Pomerleau [151]. Another notable approach by Anthony et al. [85] introduces a novel model-based architecture that leverages 3D geometry as an inductive bias. This method is trained solely on an offline dataset of expert driving data, eliminating the need for reward signals or online interaction. This approach shows great promise for real-world applications.

The author utilizes behaviour cloning (BC) [46], a technique that focuses solely on imitating the expert's policy. BC is straightforward and effective, but it struggles with unknown states, requiring a substantial amount of data. To address this limitation and produce a policy instead of a cost function, Generative Adversarial Imitation Learning (GAIL), proposed by Ho et al. [81], uses the Generative Adversarial Network (GAN) approach for imitation learning in RL. GAIL extracts policies directly from data rather than relying on expert demonstrations. GAIL, similar to GAN, is based on the fundamental concept of a generator and discriminator. The generator in GAIL produces trajectories that resemble those of an expert as closely as possible, while the discriminator determines whether the generated trajectories are from the expert or not, as shown in Fig. 17. To address GAIL's limitations in only using the current state to model the subsequent state, Choi et al. in [35] propose a method that incorporates a Partially Observable Markov Decision Process (POMDP) within the GAIL framework and uses the reward function from the discriminator to train the model. The high-dimensional solution space of a POMDP makes complex scenario modelling computationally expensive. Additionally, the ambiguity of state observations makes it difficult to differentiate state-action pairs. However, notable advancements have been made in online POMDPs, as demonstrated in [167] and [166]. Bronstein et al. [23] modify the default model-based GAIL with a hierarchical model to enable generalisation to any goal pathways and evaluate performance with simulated interactive agents in a closed-loop evaluation framework. Kuefler et al. [108] employ GAIL to model human driving behaviour on highways and propose an RNN integrated into the GAIL architecture. Bansal et al. [15]'s ChaffeurNet utilizes IL to train a robust policy while penalizing implausible events and introduces an explicit loss to prevent the algorithm from solely imitating such undesirable behaviour. In summary, IL and GAIL are promising approaches to address the challenges of training RL algorithms in scenarios with limited rewards or no direct reward function. Their success in modelling human driving behaviour and generating realistic predictions opens up possibilities for their application in other real-world scenarios. Table 28 presents an overview of the IL models, detailing their design and implementation specifics. Additionally, Table 29 offers a concise summary of reinforcement learning-based approaches including IRL, deep IRL and IL for trajectory prediction, outlining the strengths and weaknesses of each study.

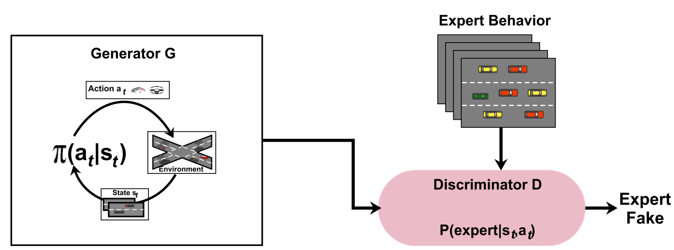

Fig. 17. Depiction of the Generative Adversarial Imitation learning based method.

Integrated with deep learning networks, these approaches effectively extract insights from expert demonstrations and consider diverse factors, including the direct interaction between objects. They leverage diverse features, such as semantic information from road maps, and demonstrate the capability for long-term prediction. Nonetheless, their computational intensity and lengthy training periods pose challenges. A solution lies in exploring optimisation techniques, including algorithmic

Table 27
Summary of Deep Inverse Reinforcement Learning methods with design and implementation parameters.

Work	State Space	Action Space	Reward Function	Optimisation/ Learning algorithm	Evaluation metrics	Expert Data	Deep neural network	Platform	Activation function
[237] 2018	Steering Angle, Throttle, Brake, Speed	Do not turn to the straight line, turn left slightly, turn right slightly, turn left vigorously, turn right vigorously	State feature; maximum margin planning algorithm	SARSA algorithm	L2 error	Driving trajectory of steady driving driver	Convolutional neural network;three-layer neural network for feature	-	Sigmoid function
[215] 2019	9-cell internal-lane state, 6-cell left-boundary state, 6-cell right-boundary state	Maintain, accelerate, brake, left-turn, right-turn	Linear combination of “Action features, Position of the HV, Overtaking strategy, Tailgating, Collision incident”	Q-learning	Prediction error	Past states and actions	DNN for reward function	-	Tanh activation
[234] 2020	2D position of the vehicle	Combinations of steering angle, driving forward or backward	-	Adam optimiser	Hausdorff Distance	Normal driving behaviour, avoid the negative obstacles, cross negative obstacles if they block the way, cross all negative obstacles on the road	Five-layer fully convolutional neural network for feature	Intel Xeon E5 CPU and an NVIDIA TiTanX GPU	
[98] 2021	Cell in the grid map	Stay, east, west, north, northwest, and northeast	based on spatiotemporal feature	ADAM optimizer	Hausdorff distance, Negative log-likelihood	12000 demonstrations: vehicle positions for the past 3 s and 3 s from the present time	ConvLSTM-based DNN as the reward function	Pytorch; deep learning framework, NVIDIA DGX station, 2080Ti GPU with i9 CPU	ReLU activation

“- This information is not available/applicable for work”.

enhancements like experience replay or prioritized experience replay, which can significantly improve efficiency and reduce overall training time.

6. Training and evaluation

Various standard datasets are used to test prediction algorithms, and appropriate metrics are used to assess their performance. This section offers a concise overview of the simulators subjected to analysis and comparison.

6.1. Datasets

To evaluate the accuracy of a trajectory prediction model, the projected trajectory and ground truth trajectory are usually compared. These trajectories are obtained from multiple datasets that are collected using sensors such as LiDAR, cameras, radar etc. The vehicle movements in these datasets are either automatically generated or manually annotated. Modern benchmarks have made significant progress in the AVs prediction field, overcoming the limitations of older datasets which were constrained in terms of environments and agent categories. The NGSIM-180 [38] and highD [107] dataset are examples of such benchmarks that utilized drones and surveillance cameras to capture cars on highways. These datasets focused on a single type of agent with a limited set of possible actions, which included moving left or right and maintaining a straight path. The KITTI [61] dataset, introduced by Geiger et al. in 2013, was among the earliest multimodality datasets that included LiDAR point clouds in addition to camera frames for input scenes. This development has generated a recent interest in object detection using 3D bounding boxes [30]. Moreover, KITTI [61] offers annotations for both cars and pedestrians.

As the depth of AI models increases, more images are required for efficient generalisation. Recent datasets such as Lyft [84], Waymo [54], nuScenes [24], and Argoverse [193] have significantly increased the number of annotated frames, thereby facilitating the training of deep models. These datasets not only include camera and LiDAR data but also provide High Definition (HD) maps that capture the road’s topology [162]. The addition of HD maps has made it possible to investigate global navigation abilities, thus enabling the training of models for longer prediction horizons. Unlike previous datasets, the aforementioned datasets cover more classes, record ego-vehicle odometry data, encompass various cities, different weather and lighting conditions (including rain and night), and provide labels for other agents such as traffic lights and road rules. However, they still lack labels related to intention prediction.

To summarize, modern datasets have effectively addressed many of the challenges associated with prediction by providing a vast amount of diverse, multi-agent, multi-modal data. This data can be used to train models capable of predicting the behaviour of various interacting agents in diverse weather conditions. Furthermore, these datasets offer annotations that are useful for high-level comprehension of the driving scene, including information on location, action, and events. Table 30 presents

Table 28
Summary of Imitation Learning methods with design and implementation parameters.

Work	Input	Loss function	Deep neural network	Learning rate	Activation function/ Optimiser	Evaluation metrics	Training/ Platform	Expert data
[15] 2018	Coordinates, Heading, Speed, Images	Collision Loss, On Road Loss, Geometry Loss, Auxiliary Losses, Agent Position, Heading and Box Prediction loss, Agent Meta Prediction loss	Convolutional neural network, Recurrent neural network	-	Sigmoid activation	L2 distance error	NVidia Tesla P100 GPU	30 million real-world expert driving examples
[85] 2022	Image, route, speed, acceleration, steering	Kullback-Leibler (KL) divergence, mean-squared error, L1 and cross-entropy loss	Recurrent neural network	Learning rate 0.0001, weight decay 0.01	AdamW optimizer	Route completion, infraction penalty, and driving score	50, 000 iterations on a batch size of 64 on 8 V100 GPUs	high-resolution videos
[35] 2021	ID of links in the road network and two virtual tokens representing the start and the end of a trip (Start,End); Straight,Left,Right, Terminate	mean-squared error, cross-entropy loss	Long Short Term Memory, multi-layer perception, gated recurrent cell	Learning rate 0.00005		BLEU score, METEOR score	Number of iterations 20,000, Number of samples 20,000; i9-10900 KF CPU, 64 GB RAM, and Nvidia Geforce RTX 3080	Real vehicle trajectory dataset
[23] 2022	Pose, Other actors' bounding boxes, Traffic light state	Discriminator loss, MGAIL policy loss, and BC loss	Gated recurrent unit	Learning rate of 0.0003	Adam optimiser	Road-Route Failure, Collision, Off-road	-	Real vehicles driving over 100,000 miles in San Francisco

"- This information is not available/applicable for work".

an overview of the popular datasets commonly utilized in trajectory prediction tasks. The table includes information about the sensors used, scene descriptions, applications, and the research articles that have utilized these datasets. The majority of the techniques described in this paper employ trajectories as input, however, some also make use of vehicle states or map data.

6.2. Evaluation metrics

Evaluation Metrics (EMs) are crucial for assessing the effectiveness of vehicle trajectory prediction models. One common metric used for evaluating model output is the *Average Displacement Error* (ADE), which measures the mean l_2 distance between the predicted trajectory's locations and the corresponding ground truth. Another metric, the *Final Displacement Error* (FDE), calculates the same distance but only for the final predicted location and its ground truth at the prediction horizon. Probabilistic generative models that produce multi-modal predictions require additional metrics. The *Best of N* metric calculates ADE and FDE for the best N samples out of all generated trajectories. When N equals 1, the method is called minADE and minFDE, respectively, and only the generated trajectory which is closest to the ground truth is selected. Other metrics for the multi-modal distribution include various versions of *Negative Log Likelihood* (NLL), which compares the distribution of generated trajectories against the ground truth. To evaluate the performance of the model on the ApolloScape [91] trajectory dataset in the literature, two metrics, the *Weighted Sum of Average Displacement Errors* (WSADE) and the *Weighted Sum of Final Displacement Errors* (WSFDE), are frequently used. Table 31 highlights the commonly used Evaluation metrics for trajectory prediction tasks with their formula and description.

6.3. Simulator

MATLAB/Simulink [3] excels in simple scenarios, boasting efficient computation and plot functions. Co-simulation with software like CarSim [5] facilitates diverse vehicle model creation, with users often combining CarSim's vehicle models and MATLAB/Simulink [3] for control algorithm co-simulation. However, MATLAB/Simulink [3] has limitations in realistic visualization of traffic scenarios. In contrast, PreScan [4] offers robust simulation capabilities, including realistic weather conditions not achievable with MATLAB/Simulink [3] and CarSim [5]. Its interface with MATLAB/Simulink [3] enhances modelling efficiency. Gazebo [57] offers flexibility and integrates well with ROS, allowing comprehensive simulation control but requiring manual effort for world creation. Unlike CARLA [52] and LGSVL [156], Gazebo [57] lacks automated processes for defining simulation worlds, relying on users to manually specify 3D models and physics in XML files. The comparison then shifts to CARLA [52] and LGSVL [156] simulators. CARLA [52] and LGSVL [156] require GPU units for optimal performance. They offer flexible APIs with different facilities; CARLA [52] has a built-in recorder, while LGSVL [156] relies on Nvidia drivers for video recording. Both provide common sensors, and users can create custom sensors. CARLA [52] uses Unreal Engine for automated map generation based on OpenDRIVE, whereas LGSVL [156], using Unity, involves manual component import. Software architecture differs; LGSVL [156] connects to AD stacks via bridges, while CARLA [52], with most facilities built-in, allows connection to ROS1, ROS2, and Autoware through bridges. Table 32 presents a comparative summary encompassing all six simulators discussed in this paper.

7. Discussion

In this section, a fair evaluation of the proposed models is presented through a comparison of representative models. The selected criteria encompass different factors that pertain to the task of trajectory prediction, as well as the overall prerequisites for utilizing the models in the field. Nonetheless, the comparison reveals prevailing patterns and provides an understanding of particular characteristics and scenarios of use. Deep Learning-based models and Reinforcement learning-based approaches shall be compared. The comparison results are summarized in Table 33.

Table 29
Summary of Reinforcement Learning based methods: Related Work & Year, Major Techniques, Scenarios, and their applications, Environment, Advantages, and Limitation of each work.

Work	Techniques	Scenarios	Application	Environment	Advantages	Limitation
[184] 2018	Deep Q-learning	Highway segment with three lanes in one direction	Lane change, Vehicle Control.	Simulator	The ability of the vehicle agent to learn a safe and effective lane change driving strategy.	The algorithm's performance degrades in various traffic flow scenarios and road layouts.
[70] 2018	Markov decision process, Dynamic programming,	Two-lane highway scenario.	Decision-making, overtaking and avoiding collision.	Simulator	This approach might be used in common situations like highway and park driving, eliminating the need to manually model rules and compare them to rule-based approaches.	Due to the high computational complexity, the time required increases dramatically as the dimension of the state space grows.
[237] 2018	Inverse Reinforcement Learning (IRL), Markov decision process, CNN	Driving curve scenario.	Left turn and right turn, driver behaviour modelling	Simulator	With many unvisited states in the new curve scenario, this technique exhibits excellent generalisation efficiency.	In the demonstration trajectory state, the agent is unable to make parallel driving decisions, but it can still perform the decision-making tasks in an unvisited state.
[172] 2018	Hierarchical IRL	Ramp-merging driving scenario.	The discrete driving decisions such as yield or pass as well as the continuous trajectories.	NGSIM	Driving choice influences are explicitly modelled for both discrete and continuous driving situations.	
[67] 2018	Maximum Entropy IRL, Semi-Markov Decision Process (SMDP).	Merges to the right, overtakes, becomes snarled in sluggish traffic, and drives carelessly.	Two-lane highway.	Self-gathered data	The suggested method successfully models the problem of highway driving from inefficient driving demonstrations captured using an instrumented vehicle.	This does not extend the original spatiotemporal state lattice concept to handle the stop-and-go scenarios.
[204] 2019	Accelerated IRL, maximum entropy	Ego vehicle and one front car in the scenarios.	Lane change, Lane keeping	Simulator	It substitutes choosing the ideal trajectories from the candidate trajectory library produced by random sampling policies for invoking RL to generate the best trajectory during each iteration.	This model is specific to lane changes and lane keeping and does not generalise to other types of driving conditions, such as intersections and unpaved roads; 2. There is a need to generalize the proposed method under a distributed learning framework to address the stochastic problem.
[211] 2019	Inverse Optimal Control (IOC), Langevin Sampling, Monte Carlo neural network	Greater length of the road segment, more lane curvature, and more highway entrances and exits.	Lane-following, avoiding collisions, and passing, Optimal Control.	NGSIM	1. Make more stable predictions that are better; 2. Langevin Sampling and the energy-based approach can handle complex cost functions.	This does not predict the joint trajectory distribution for all moving agents; it just predicts individual trajectories.
[207] 2020	Numerical Optimisation algorithms	With a length of 65.3 km and an intended top speed of 80 km/h, the multi-lane highway is devoid of traffic signals	Lane change and car followings motion planning	Self-collected	Heuristic and learning-based lane incentive costs that are suggested and put into practice.	1. Actors' interaction behaviours are not taken into account; 2. Because the large size vehicles are not included in the dataset, the shape, size, and heading of the vehicles are not adequately addressed.
[198] 2020	Maximum-entropy IRL	Settings for both interactive and non-interactive driving.		INTERACTION	The suggested algorithm is more generalizable and converges much more quickly.	The proposed technology does not extend to generic robotic systems and was created primarily for ground vehicles and other mobile robot systems.
[45] 2020	Maximum-entropy IRL, attention mechanism	Unknown environments.	Multimodal trajectories	Stanford drone, NuScenes datasets	Using MaxEnt IRL to seek a strategy that can jointly predict agents' intentions and paths on a rough 2-D grid defined across the scene.	It does not consider the interaction factors of surrounding agents for future trajectory prediction.

(continued on next page)

Table 29 (*continued*)

Work	Techniques	Scenarios	Application	Environ-ment	Advantages	Limitation
[215] 2019	Markov decision process, IRL	Segments of highway and bend roads, each segment having five lanes.	Lane switching, maintaining lanes and speeds, accelerating and braking, overtaking, and tailgating.	Simulator	The traffic model is easily expandable to accommodate more vehicles and lanes.	unable to distinguish between different car kinds in the simulator.
[234] 2020	Maximum Entropy Inverse Reinforcement Learning, CNN	Straight and flat road scenes and negative obstacles scenes.	Normal driving behaviour and avoid or cross the negative obstacles.	Self collected	1. To facilitate effective forward reinforcement learning, two new CNN are proposed. This addresses the issue of Complexity of state-space grows exponentially; 2. Different cost functions of traversability analysis are learned to guide the trajectory planning of different behaviours	Extensive experimental research will only be performed in certain circumstances.
[98] 2021	Deep IRL, ConvLSTM	Keeping to the lane and moving through intersections whether or not there are other agents.	Traversability map.	Self-collected	1. It takes into account several circumstances at once for an autonomous vehicle's social navigation; 2. It is independent of expensive prior environmental knowledge.	1. The model did not take into account traffic signals and stop signs; 2. They did not take into account the dynamic heterogeneous agents in relation to the nearby automobiles.
[35] 2021	GAIL, imitation learning, POMDP	Four-way intersections	Left, right, terminate	Simulator, self-collected	1. Enables trajectory generation that can scale to large road network environments; 2. Describing the underlying route distribution of a traffic network in synthetic data generation problems.	It does not take into account the effects of traffic conditions or interactions with other vehicles.
[33] 2022	Imitation learning	Four training cities, each with four weather conditions	Distribution of diverse futures states and actions.	Simulator	This strategy uses a camera-only methodology to simulate static and dynamic scenes as well as ego behaviour when driving in urban areas; 2. Hat jointly learns a driving policy and a world model from offline expert demonstrations alone.	They are not driving reward function from expert data.
[85] 2022	Maximum entropy IRL	Cut-ins, sudden stops, and crowded hotel pickup/drop-off areas.	Multi modal	Self collected	1. This system calculates, evaluates, and scores vehicle's trajectory; 2. A straightforward design, easily understandable features, and potent real-world performance.	Evaluation metrics are not good enough to measure the performance of the model.
[148] 2022	GAIL	Hill-climbing.	Trajectory planning	Self-gathered	Underline the value of closed-loop evaluation and training using interactive agents.	1. The ability of the policy may be enhanced by directly teaching the planning agent alongside already taught interactive agents; 2. Generalisation to innovative routes is not addressed.
[23] 2022	Deep IRL, maximum entropy IRL	Mixed-driving scenario.	Multi modal	Nuscene	1. This uses scene rasterisation to provide the neural network's input scene data; 2. The trajectory generator module now includes a correction factor, by disfavouring trajectories with little difference, can produce more varied trajectories.	Scene rasterisation could be hampered by ineffective coding, a lengthy learning curve, and a loss of connection information from occlusion.

Deep learning-based models have demonstrated their ability to produce accurate predictions over an extended period, as they can conduct long-term predictions of up to 8 seconds. However, these models are typically comprised of neural networks and are therefore considered black-box models, which reduces their explainability and could pose challenges in terms of validation and approval. Despite this, these models have the advantage of being holistic since they can integrate various features from multiple sources, including object interaction and semantic data, into the neural network. However, to achieve good prediction performance, it is crucial to carefully select valid features. The use of spatial features and corresponding representation enables the consideration of the interaction between agents, which makes interaction awareness possible. Deep learning-based models have the capability to describe complex processes at varying levels of abstraction, with the ability to output trajectories as prediction results. However, these models require valid training data that reflects the specific field of application to enable comprehensive and robust predictions. As a result, these models are highly data-dependent. Additionally, the adaptivity of

Table 30
Datasets for AVs which are utilized in trajectory prediction.

Dataset	LiDAR	Camera	Radar	Drone	Scene Description	Applications	Used by
Nuscenes [24] 2020	✓	✓	✓		Each scene in nuScenes is 20 seconds long, completely annotated, and has 3D bounding boxes for 8 attributes and 23 classes.	Object identification, tracking, and segmentation	Lapred [103], MHA-JAM [134], Covernet [147], GNN+GAT [131], STL-GAN [118], ME-GAN [71], CVAE and CapsNet [53]
Lyft [84] 2021	✓	✓			During the course of four months, this was gathered by a fleet of 20 autonomous vehicles travelling along a predetermined path in Palo Alto, California. There are 170,000 scenes total, and each scene lasts for 25 seconds.	Motion forecasting and planning and simulation.	two-stage safety-balanced driving-style-aware trajectory planner [185], graph-LSTMs [26], Gatformer [220]
Waymo Open datasets [54] 2021	✓	✓			Presently contain 1,950 segments of 20 seconds each, accumulated over 390,000 frames in various environments.	Object detection	Scene transformer [142], Densetnt [69]
Argoverse [193] 2023	✓	✓			This dataset consists of 360-degree pictures captured by seven cameras with overlapping fields of view, 3D point clouds generated by long-range LiDAR, 6-DOF posture annotations, and 3D track annotations.	Tracking and predicting movements of 3D objects.	LAMP-Net [102], MFP [174], CNN+TCN [171], Lapred [103], Baseline Network+Trajectory Proposal Attention [105], mmTransformer [127], multimodal attention Transformer [93], Scene transformer [142], Lane transformer [190], graph-lstms [26], GAIN [126], Ai-tp [221], Vectornet [58], Urtasun [121], Tnt [227], Densetnt [69], Lanercnn [217], ME-GAN [71], Smart [170]
INTERACTION [218] 2019		✓		✓	The interactive driving scenarios cover a wide range of situations, such as merging and lane changes in urban, highway, and ramp environments, roundabouts with yield and stop signs, signalised intersections, etc.	Decision-making, planning, and imitation learning	Hsta [196], Transformer+Augmented Information [153], HEAT [138], Recog [139], Tnt [227]
HighD [107] 2018				✓	In six different places, traffic was recorded, and more than 110 500 vehicles were present. The trajectory of each vehicle, including its type, size, and manoeuvres, is automatically retrieved.	Traffic pattern analysis or driver model parameterisation	TCN-MLP [115], LSTM encoder–decoder [212], Hsta [196], multi-head attention [104], multi-head attention [135], transformer+augmented information [153], SAL+TAL+CAL [31], dual Transformer [59], DRM-DL [125]
Apolloscape [91] 2018	✓	✓			Amazing collection of more than 140,000 video frames from several sites in China were collected during varied weather situations	Scene parsing, lane segmentation, object detection and tracking in three dimensions, and self-localisation	GRIP++ [119], graph-LSTMS [26], Mvhgn [209], Ai-tp [221], TPA+GCN [210]
Kitti [61] 2013	✓	✓			This dataset are gathered while travelling through rural areas, on highways, and around Karlsruhe, a medium-sized city. Per shot, up to 15 cars and 30 pedestrians can be seen.	3D object detection and 3D tracking	Mantra [130]
NGSIM [38] 2017	✓				Road segment lengths of 640 and 500 meters were recorded for US Route 101 and Interstate 80	Trajectory prediction	Dual LSTM [205], ST-LSTM [40], VBIN [48], LSTM [44], MFP [174], Sense–Learn–Reason–Predict [203], Deeptrack [101], GRU+attention [74], LSTM+MDN+attention [56], STA-LSTM [123], DSCAN [216], LSTM+attention [132], Hsta [196], LCTP [213], LSTM+multi-head attention [135], TP2Net [87], Mals-net [75], Spatial-Channel Transformer Network [228], SAL+TAL+CAL [31], Structural Transformer [82], Dual Transformer [59], Convolutional Social Pooling [43],GRIP [120], GRIP++ [119], GSTCN [164], GCM+GRU [208], Gisnet [231], Scale-net [95], graph-lstms [26], Ra-gat [50], HEAT [138], Ai-tp [221], CNN-LSTM [202], MATF [230], TPN+GCN [210], TS-GAN [189], STSF-Net [186], STL+MPC [34], VD+DCS-LSTM [219]

Table 31
Evaluation metrics applied in the literature to evaluate the performance of trajectory prediction.

Evaluation Metrics	Formula	Description
Mean Absolute Error (MAE)/Root Mean Square Error (RMSE)	RMSE= $\sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$ MAE= $\frac{1}{n}\sum_{t=1}^{n} \lvert e_t \rvert$	The MAE calculates the average of the prediction error or displacement error, represented by e_t. On the other hand, RMSE computes the square root of the average of e_t^2, considering a defined time window (t) on the prediction horizon, and n represents the number of samples in the prediction horizon. The value of e_t represents the difference between the actual and predicted data values and is utilized for the creation of a regression-based trajectory method.
Final Displacement Error (FDE)	FDE $= \lvert \hat{Y}_{end} Y_{end} \rvert$	The FDE measures the discrepancy between the predicted final location $\hat{Y}_{end}$ and the actual final location Y_{end} at the conclusion of the prediction horizon. It solely considers the forecast errors that occur in the last time step of the prediction horizon and disregards any previous errors. When dealing with multimodal predictions, the Minimum FDE (mFDE) is used to refer to the smallest FDE value among K predictions.
Average Displacement Error (ADE)	ADE $= \frac{1}{T}\sum_{t=1}^{T} \lvert\lvert \hat{Y}_t - Y_t \rvert\rvert_2$	The ADE is the distance between the predicted location $\hat{Y}_t$ and the actual location Y_t throughout the prediction horizon, which is defined as T. When dealing with multimodal predictions, the Minimum ADE (mADE) is used to represent the smallest ADE value among K predictions.
Negative Log Likelihood (NLL)	$H(q,p) = E_{x\sim p} - log((p(x))$	The trajectory distribution of the model is represented by p, whereas q denotes the distribution of the ground truth data.
WSADE, WSFDE [119], [209], [221]	$WSADE = D_V.ADE_V + D_P.ADE_P + D_C.ADE_C$ $WSFDE = D_V.FDE_V + D_P.FDE_P + D_C.FDE_C$	Given the dissimilar characteristics of vehicle, cyclist, and pedestrian trajectories, the variables D_V, D_P, and D_C are inversely proportional to the mean speeds of vehicles, pedestrians, and cyclists, respectively, in the dataset. The evaluation metrics used for vehicles include ADE_V and FDE_V, for pedestrians ADE_P and FDE_P, and for cyclists ADE_C and FDE_C.
Miss Rate (MR)		To compute the ratio of predicted trajectories that differ by more than 2.0 meters from the ground truth, the Euclidean distance between their final positions is calculated. In scenarios where the prediction outcomes are multimodal, K feasible future trajectories are taken into account, and the optimal future trajectory is used to evaluate ADE, FDE, and MR, which will be indicated as ADE_K, FDE_K, and MR_K, respectively.
Computation Time	Based on the hardware configuration or specifications.	Computation time is a critical factor in determining the on-board performance of the method. Although autonomous vehicles have limited computing capabilities, trajectory prediction models are often complex and require substantial computational resources. As the level of autonomous driving increases, it becomes imperative for each module to execute computations at a faster rate to reduce any potential delays. Therefore, the model's real-time performance or computational cost is of utmost importance.
Prediction Horizon	Based on the specific use case or application.	The time steps into the future that the model can predict are referred to as the prediction horizon. In a dynamic and, at times, unpredictable driving environment, the accuracy of trajectory prediction models typically declines as the prediction horizon increases. Nonetheless, to fulfil the requirements of the planning and control system, the forecast time should not be excessively brief, and it should be consistent with other modules, even in a dynamic and stochastic environment.

Table 32
Comparison of Various Simulators.

Requirements	MATLAB (Simulink) [3]	CarSim [5]	PreScan [4]	CARLA [52]	Gazebo [57]	LGSVL [156]
Sensor Models	✓	✓	✓	✓	✓	✓
Weather Conditions	✗	✗	✓	✓	✗	✓
Camera Calibration	✓	✗	✓	✓	✗	✗
Path Planning	✓	✓	✓	✓	✓	✓
Vehicle Dynamics	✓	✓	✓	✓	✓	✓
3D Virtual Environment	Unknown	✓	✓	✓, Outdoor (Urban)	✓, Indoor and Outdoor	✓, Outdoor (Urban)
Traffic Infrastructure	✓, with lights model	✓	✓	✓, Traffic lights, Intersections, Stop signs, lanes	✓, allows manual model building	✓
Dynamic Objects	✓	✓	✓	✓	✗	✓
2D/3D Ground Truth	✓	✗	✗	✓	Unknown	✓
Interfaces	✓, Carsim, Prescan, ROS	✓, Matlab(Simulink)	✓, MATLAB(Simulink)	✓, ROS, Autoware	✓, ROS	✓, Autoware, Apollo, ROS
Scalability	Unknown	Unknown	Unknown	✓	✓	✓
Open Source	✗	✗	✗	✓	✓	✓
Maintenance/Stability	✓	✓	✓	✓	✓	✓
Portability	✓	✓	✓	✓, Windows, Linux	✓, Windows, Linux	✓, Windows, Linux
Flexible API	✓	✓	Unknown	✓	✓	✓

Table 33
Comparison of prediction models.

	Prediction Horizon	Explainability	Holism	Complexity	Data Dependency	Adaptivity	Computational Time	Accuracy
Deep learning-based methods	Long-term(5 s-8 s)	Low	High	High	High	Medium	High	High
Reinforcement learning-based methods	Long-term(5 s-8 s)	Medium(IRL), Low(IL)	High	High	Very High	High, if reward function is learned (IRL)	High	Medium

Holism: Adopting object interaction and semantic data, Adaptivity: Robust application in unknowable situations.

these models is limited to scenarios that fall within the data the model has been trained on. Due to their holistic approach, Deep learning-based models are typically associated with high computational costs, which are strongly influenced by the size of the neural networks used. Nevertheless, in the current state of the art, Deep learning-based models offer the highest prediction accuracy.

Reinforcement learning-based methods are also capable of conducting long-term predictions. However, the degree of explainability varies depending on the specific approach used. Indirect models generate a cost function that is mapped to state-action tuples, which can be used to interpret the proposed output of a policy. Nevertheless, it is challenging to explain how the cost function is determined from an expert's demonstration. Direct models that output a policy do not explicitly derive a cost function from demonstration, making them less explainable. These models can directly consider the interaction between multiple objects as an input feature. Additionally, a wide range of features, including semantical information from road maps, can be used as input, making these models holistic.

Reinforcement learning-based models have the ability to describe complex manoeuvres by utilizing the underlying policy. However, the model's output typically consists of discrete manoeuvres because policies comprise state-action tuples that objects may execute. Although explicit trajectories can be derived from subsequent modules, such as a Recurrent Neural Network (RNN) demonstrated in [15], these models heavily rely on diverse data, including demonstrations, for training. Extracting comprehensive cost functions or robust policies is particularly challenging as it strongly relies on expert behaviour observations, making it difficult to train correctly. Reinforcement learning-based models are designed to reason about an object's motion, allowing them to adapt well to unknown scenarios. However, similar to Deep learning-based models, holistic models based on the reinforcement learning approach have high computational costs. Moreover, the complexity of learning a robust policy negatively affects prediction accuracy.

8. Challenges and future research directions

Based on the above survey, this section highlights the research challenges and future research directions in the domain of Autonomous Vehicles (AVs) trajectory predictions.

8.1. Challenges

Trajectory prediction is a critical component of AV systems, as it enables them to anticipate the future motion of traffic agents such as vehicles, pedestrians, and cyclists in their environment. However, there are several challenges specific to the domain of AVs that make trajectory prediction exceptionally challenging:

1. *Uncertainty:* The future trajectory of traffic agents is inherently uncertain, and it is impossible to predict it with 100 percent accuracy. Various factors such as noise in sensor measurements, unpredictable environmental changes, and unknown intentions of other traffic agents can contribute to this uncertainty.
2. *Complex dynamics:* The motion of traffic agents can be affected by various physical laws, including gravity, friction, and aerodynamic forces. These dynamics can be highly complex and nonlinear, making it difficult to model accurately.
3. *Limited sensor coverage:* Autonomous vehicles rely on a suite of sensors, including cameras, LiDAR, and radar, to perceive their environment. However, the coverage of these sensors is limited, as depicted in Fig. 18, and can be affected by occlusions, weather conditions, and other factors that can make it difficult to accurately track the motion of other traffic agents.
4. *Limited data:* In some cases, there may be limited or incomplete data available for trajectory prediction. This can occur when sensors are malfunctioning, or when the historical data is missing or corrupted.
5. *Long-term prediction:* Predicting trajectories over a long time horizon (no less than 3 seconds) can be challenging, as small errors in the initial prediction can compound and result in significant deviations from the true trajectory.
6. *Complex road environments:* Autonomous vehicles operate in complex and dynamic road environments, which can include intersections, roundabouts, and crowded urban areas. Predicting trajectories in these environments requires models that can handle complex interactions between multiple agents, including other vehicles, pedestrians, and cyclists.
7. *Multimodal Outputs:* In autonomous driving, agents' behaviours exhibit multimodality, where a single past trajectory can have multiple potential future trajectories, as depicted in Fig. 19.
8. *Sparse and noisy data:* The data from sensors can be sparse and noisy, particularly in urban areas where buildings and other structures can obstruct the line of sight between the sensors and the objects being tracked. This can make it difficult to accurately model the motion of other traffic agents over time.
9. *Multi-agent interactions:* In many real-world scenarios, multiple agents interact with each other, and their trajectories are interde-

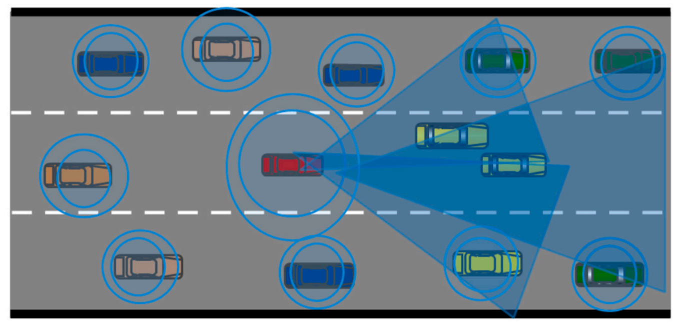

Fig. 18. Depiction of limited visibility range of in-car sensors in the context of driving on a three-lane highway. The red vehicle fully covers the Yellow vehicle and partially covers the green vehicle.

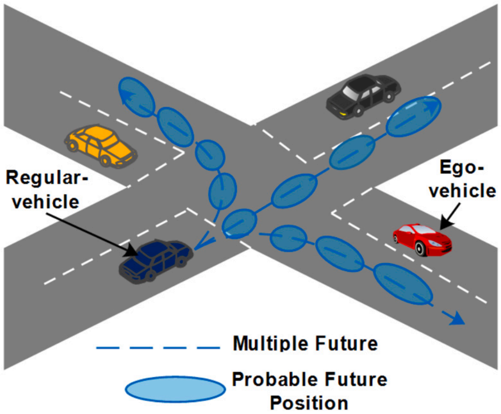

Fig. 19. Depiction of the presence of multimodal nature of vehicle and uncertainties in the urban street setting. A common scenario where the self-driving car must decide on its next move while facing various uncertainties related to the anticipated movement of other regular vehicles.

Fig. 20. Depiction of the intersection scenarios: The Ego-Vehicle, represented by the Red vehicle, is about to change lanes to the target lane. To ensure socially acceptable driving behaviour, the Ego-Vehicle must anticipate the future movements of the yellow vehicles. This prediction also includes taking into account the presence of green vehicles.

pendent. Predicting the trajectory of one agent may depend on the actions of other agents, as depicted in Fig. 20, making the problem even more challenging.

10. *Heterogeneous environment:* A Heterogeneous environment refers to an environment that contains a diverse range of elements, such as various types of vehicles, pedestrians, cyclists, different road types, and complex interactions among them. In order to effectively predict trajectories in such environments, prediction models need to account for the different types of agents, incorporate contextual information, fuse sensor data, model interactions among multiple agents, estimate uncertainty, and enable adaptability.
11. *Safety-critical applications:* Autonomous vehicles are safety-critical systems, and errors in trajectory prediction can have serious consequences, including accidents and injuries. As a result, trajectory prediction algorithms need to be highly accurate and reliable, with well-defined safety margins.
12. *Real-time constraints:* Autonomous vehicles operate in real-time environments, and trajectory prediction algorithms need to be able to process data and generate predictions in real-time. This requires efficient algorithms and hardware architectures that can handle the large amounts of data generated by the sensors.

8.2. Future research directions

The field of trajectory prediction is undergoing rapid evolution in research, offering numerous opportunities for future investigations, particularly within the realm of autonomous vehicles. Several potential research directions can shape the trajectory prediction landscape. Here, we outline the identified futuristic research directions as follows:

1. *Incorporating context and intention:* One limitation of current trajectory prediction methods is that they often focus solely on the motion of other vehicles, without considering the context or intention behind that motion. Future research could explore how to incorporate contextual information, such as road layout and traffic rules, as well as the intention of other drivers, to improve trajectory prediction accuracy.
2. *Integration of multiple sensors:* Autonomous vehicles rely on a suite of sensors to perceive their environment, and future research could explore how to integrate data from multiple sensors to improve trajectory prediction accuracy. This could involve developing new algorithms for fusing data from cameras, LiDAR, radar, and other sensors, as well as exploring new sensor modalities such as acoustic or thermal sensors.
3. *Uncertainty modelling:* Trajectory prediction is inherently uncertain, and future research could explore how to model and propagate uncertainty through the prediction pipeline. This could involve developing new probabilistic models, such as Bayesian neural networks, or exploring new techniques for uncertainty quantification and propagation [180].
4. *Human-aware trajectory prediction:* Autonomous vehicles operate in environments that include not only other vehicles but also pedestrians and cyclists. Future research could explore how to develop trajectory prediction methods that are aware of human behaviour and can accurately predict the motion of pedestrians and cyclists in crowded urban environments.
5. *Real-time implementation & Hardware acceleration:* Autonomous vehicles operate in real-time environments, and trajectory prediction algorithms need to be able to process data and generate predictions in real-time. Future research could explore how to optimize trajectory prediction algorithms for real-time performance, as well as developing new hardware architectures for efficient computation [97] [110].
6. *Ensuring safety and robustness:* Safety is of paramount importance in autonomous driving systems. Future research should aim to develop trajectory prediction methods that prioritize safety and robustness. This includes investigating techniques for handling rare or anomalous events, improving prediction accuracy in challenging weather conditions, and considering ethical aspects in trajectory prediction algorithms [11].
7. *Relative trajectory prediction:* Relative trajectory prediction refers to the task of predicting the future motion or path of surrounding objects or agents relative to the ego vehicle or coordinate system. Future research should focus on estimating the relative displacement, velocities, and trajectories of other vehicles, pedestrians, and cyclists with respect to the ego vehicle [128].
8. *Random obstacle aware trajectory prediction:* This approach refers to predicting the future trajectories of a vehicle while considering the presence of unexpected or random obstacles in the surrounding environment. These obstacles can be animals or objects in between roads, the sudden arrival of pedestrians, and road accidents that lead to an uncertain obstacle in between roads. Future research should focus on incorporating rare events into the prediction models and collecting and analyzing data related to these rare events to develop more comprehensive and robust prediction models.
9. *Challenging Weather condition:* Adverse weather conditions, such as heavy rain, snow, fog, or low visibility, can affect the performance of sensors and limit the availability of critical data for trajectory prediction. Future research should focus on involves incorporating techniques such as sensor fusion, adaptive filtering, probabilistic modelling, and machine learning to improve the reliability and accuracy of trajectory predictions under adverse weather conditions.

10. *Vehicle-to-Vehicle (V2V) communication and Vehicle-to-Everything (V2X) communication strategies:*
V2V communication refers to the exchange of information directly between vehicles. V2X communication expands beyond V2V and includes communication with other entities such as infrastructure, pedestrians, cyclists, and traffic management systems. By sharing real-time data such as position, speed, acceleration, and intentions, vehicles can collaborate and cooperate to enhance trajectory prediction [191], [109].
11. *Hybridisation of several approaches:*
Multiple strategies are suggested in Sections 3, 4, and 5 for solving the task of trajectory prediction. Hybridisation can take different forms depending on the specific context and requirements. This can lead to more accurate and robust trajectory predictions

9. Conclusion

This paper provides an extensive survey of the current state-of-the-art Machine Learning (ML)-based trajectory prediction methods for Autonomous Vehicles (AVs). These ML-based approaches have demonstrated significant promise in accurately predicting trajectories, employing several techniques like deep learning-based methods and reinforcement learning-based methods. Deep learning-based methods including sequential models, vision-based models, and generative models are thoroughly explored, highlighting their respective strengths and weaknesses in trajectory prediction tasks. Furthermore, the review focuses on the discussion of reinforcement learning methods, including Inverse reinforcement learning, deep inverse reinforcement learning, and imitation learning techniques. Multiple informative tables and figures are provided to facilitate a comprehensive comparative study of various approaches used to address trajectory prediction tasks. The review paper includes an analysis of multiple datasets and evaluation metrics used to assess the accuracy of trajectory prediction tasks. This conducts a comparative analysis between deep learning-based methods and reinforcement learning methods across various characteristics. Recent advances in trajectory prediction for AVs show promise, but there are still several challenges that need to be addressed. The paper outlines potential research directions, emphasizing the need for more robust and interpretable models and the exploration of new sensor modalities. The survey aims to provide a valuable reference for researchers and practitioners in this field and guide future advancements in the trajectory prediction domain.

CRediT authorship contribution statement

Vibha Bharilya: Investigation, Methodology, Visualization, Writing – original draft. **Neetesh Kumar:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Global status report on road safety 2018: Summary, world health org, Geneva, Switzerland, 2018.
[2] Road trauma Australia—annual summaries, Australia, 2022.
[3] https://in.mathworks.com/products/automated-driving.html.
[4] https://in.mathworks.com/products/connections/product_detail/prescan.html.
[5] https://www.carsim.com/.
[6] P. Abbeel, A.Y. Ng, Apprenticeship learning via inverse reinforcement learning, in: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, p. 1.
[7] S. Albeaik, A. Bayen, M.T. Chiri, X. Gong, A. Hayat, N. Kardous, A. Keimer, S.T. McQuade, B. Piccoli, Y. You, Limitations and improvements of the intelligent driver model (idm), SIAM J. Appl. Dyn. Syst. 21 (3) (2022) 1862–1892.
[8] F. Altché, A. de La Fortelle, An lstm network for highway trajectory prediction, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2017, pp. 353–359.
[9] S. Ammoun, F. Nashashibi, Real time trajectory prediction for collision risk estimation between vehicles, in: 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing, IEEE, 2009, pp. 417–422.
[10] D. Augustin, M. Hofmann, U. Konigorski, Prediction of highway lane changes based on prototype trajectories, Forsch. Ingenieurwes. 83 (2) (2019) 149–161.
[11] G.S. Aujla, R. Chaudhary, N. Kumar, J.J. Rodrigues, A. Vinel, Data offloading in 5g-enabled software-defined vehicular networks: a Stackelberg-game-based approach, IEEE Commun. Mag. 55 (8) (2017) 100–108.
[12] M.N. Azadani, A. Boukerche, A novel multimodal vehicle path prediction method based on temporal convolutional networks, IEEE Trans. Intell. Transp. Syst. 23 (12) (2022) 25384–25395.
[13] M. Bahram, A. Lawitzky, J. Friedrichs, M. Aeberhard, D. Wollherr, A game-theoretic approach to replanning-aware interactive scene prediction and planning, IEEE Trans. Veh. Technol. 65 (6) (2015) 3981–3992.
[14] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint, arXiv:1803.01271, 2018.
[15] M. Bansal, A. Krizhevsky, A. Ogale, Chauffeurnet: learning to drive by imitating the best and synthesizing the worst, arXiv preprint, arXiv:1812.03079, 2018.
[16] A. Barth, U. Franke, Where will the oncoming vehicle be the next second?, in: 2008 IEEE Intelligent Vehicles Symposium, IEEE, 2008, pp. 1068–1073.
[17] T. Batz, K. Watson, J. Beyerer, Recognition of dangerous situations within a cooperative group of vehicles, in: 2009 IEEE Intelligent Vehicles Symposium, IEEE, 2009, pp. 907–912.
[18] D.E. Benrachou, S. Glaser, M. Elhenawy, A. Rakotonirainy, Use of social interaction and intention to improve motion prediction within automated vehicle framework: a review, IEEE Trans. Intell. Transp. Syst. (2022).
[19] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, H. Ghayvat, Cnn variants for computer vision: history, architecture, application, challenges and future scope, Electronics 10 (20) (2021) 2470.
[20] A. Bhattacharyya, M. Hanselmann, M. Fritz, B. Schiele, C.-N. Straehle, Conditional flow variational autoencoders for structured sequence prediction, arXiv preprint, arXiv:1908.09008, 2019.
[21] M. Brännström, E. Coelingh, J. Sjöberg, Model-based threat assessment for avoiding arbitrary vehicle collisions, IEEE Trans. Intell. Transp. Syst. 11 (3) (2010) 658–669.
[22] A.M. Braşoveanu, R. Andonie, Visualizing transformers for nlp: a brief survey, in: 2020 24th International Conference Information Visualisation (IV), IEEE, 2020, pp. 270–279.
[23] E. Bronstein, M. Palatucci, D. Notz, B. White, A. Kuefler, Y. Lu, S. Paul, P. Nikdel, P. Mougin, H. Chen, et al., Hierarchical model-based imitation learning for planning in autonomous driving, in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2022, pp. 8652–8659.
[24] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, Nuscenes: a multimodal dataset for autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11621–11631.
[25] R. Chandra, U. Bhattacharya, A. Bera, D. Manocha, Traphic: trajectory prediction in dense and heterogeneous traffic using weighted interactions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8483–8492.
[26] R. Chandra, T. Guan, S. Panuganti, T. Mittal, U. Bhattacharya, A. Bera, D. Manocha, Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms, IEEE Robot. Autom. Lett. 5 (3) (2020) 4882–4890.
[27] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al., Argoverse: 3d tracking and forecasting with rich maps, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8748–8757.
[28] N.S. Chauhan, N. Kumar, Traffic flow forecasting using attention enabled bi-lstm and gru hybrid model, in: International Conference on Neural Information Processing, Springer, 2022, pp. 505–517.
[29] W. Chen, F. Wang, H. Sun, S2tnet: spatio-temporal transformer networks for trajectory prediction in autonomous driving, in: Asian Conference on Machine Learning, PMLR, 2021, pp. 454–469.
[30] X. Chen, H. Ma, J. Wan, B. Li, T. Xia, Multi-view 3d object detection network for autonomous driving, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1907–1915.
[31] X. Chen, H. Zhang, F. Zhao, Y. Cai, H. Wang, Q. Ye, Vehicle trajectory prediction based on intention-aware non-autoregressive transformer with multi-attention learning for Internet of vehicles, IEEE Trans. Instrum. Meas. 71 (2022) 1–12.

[32] B. Cheng, I. Misra, A.G. Schwing, A. Kirillov, R. Girdhar, Masked-attention mask transformer for universal image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1290–1299.
[33] J. Cheng, R. Xin, S. Wang, M. Liu, Mpnp: multi-policy neural planner for urban driving, in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2022, pp. 10549–10554.
[34] K. Cho, T. Ha, G. Lee, S. Oh, Deep predictive autonomous driving using multi-agent joint trajectory prediction and traffic rules, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2019, pp. 2076–2081.
[35] S. Choi, J. Kim, H. Yeo, Trajgail: generating urban vehicle trajectories using generative adversarial imitation learning, Transp. Res., Part C, Emerg. Technol. 128 (2021) 103091.
[36] J.K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, Attention-based models for speech recognition, Adv. Neural Inf. Process. Syst. 28 (2015).
[37] F.-C. Chou, T.-H. Lin, H. Cui, V. Radosavljevic, T. Nguyen, T.-K. Huang, M. Niedoba, J. Schneider, N. Djuric, Predicting motion of vulnerable road users using high-definition maps and efficient convnets, in: 2020 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2020, pp. 1655–1662.
[38] B. Coifman, L. Li, A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset, Transp. Res., Part B, Methodol. 105 (2017) 362–377.
[39] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, N. Djuric, Multimodal trajectory predictions for autonomous driving using deep convolutional networks, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 2090–2096.
[40] S. Dai, L. Li, Z. Li, Modeling vehicle interactions via modified lstm models for trajectory prediction, IEEE Access 7 (2019) 38287–38296.
[41] Q. Deng, D. Söffker, Improved driving behaviors prediction based on fuzzy logic-hidden Markov model (fl-hmm), in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 2003–2008.
[42] N. Deo, A. Rangesh, M.M. Trivedi, How would surround vehicles move? A unified framework for maneuver classification and motion prediction, IEEE Trans. Intell. Veh. 3 (2) (2018) 129–140.
[43] N. Deo, M.M. Trivedi, Convolutional social pooling for vehicle trajectory prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 1468–1476.
[44] N. Deo, M.M. Trivedi, Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 1179–1184.
[45] N. Deo, M.M. Trivedi, Trajectory forecasts in unknown environments conditioned on grid-based plans, arXiv preprint, arXiv:2001.00735, 2020.
[46] T.K. Devi, A. Srivatsava, K.K. Mudgal, R.R. Jayanti, T. Karthick, Behaviour cloning for autonomous driving, Webology 17 (2) (2020) 694–705.
[47] F. Diehl, T. Brunner, M.T. Le, A. Knoll, Graph neural networks for modelling traffic participant interaction, in: 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2019, pp. 695–701.
[48] W. Ding, J. Chen, S. Shen, Predicting vehicle behaviors over an extended horizon using behavior interaction network, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 8634–8640.
[49] W. Ding, S. Shen, Online vehicle trajectory prediction using policy anticipation network and optimization-based context reasoning, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 9610–9616.
[50] Z. Ding, Z. Yao, H. Zhao, Ra-gat: repulsion and attraction graph attention for trajectory prediction, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, 2021, pp. 734–741.
[51] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, J. Schneider, Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2095–2104.
[52] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, Carla: an open urban driving simulator, in: Conference on Robot Learning, PMLR, 2017, pp. 1–16.
[53] A. Dulian, J.C. Murray, Multi-modal anticipation of stochastic trajectories in a dynamic environment with conditional variational autoencoders, arXiv preprint, arXiv:2103.03912, 2021.
[54] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C.R. Qi, Y. Zhou, et al., Large scale interactive motion forecasting for autonomous driving: the waymo open motion dataset, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9710–9719.
[55] T. Fernando, S. Denman, S. Sridharan, C. Fookes, Deep inverse reinforcement learning for behavior prediction in autonomous driving: accurate forecasts of vehicle motion, IEEE Signal Process. Mag. 38 (1) (2020) 87–96.
[56] M. Fu, T. Zhang, W. Song, Y. Yang, M. Wang, Trajectory prediction-based local spatio-temporal navigation map for autonomous driving in dynamic highway environments, IEEE Trans. Intell. Transp. Syst. 23 (7) (2021) 6418–6429.
[57] F. Furrer, M. Burri, M. Achtelik, R. Siegwart, Rotors—a modular gazebo mav simulator framework, in: Robot Operating System (ROS) the Complete Reference (Volume 1), 2016, pp. 595–625.
[58] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, C. Schmid, Vectornet: encoding hd maps and agent dynamics from vectorized representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11525–11533.
[59] K. Gao, X. Li, B. Chen, L. Hu, J. Liu, R. Du, Y. Li, Dual transformer based prediction for lane change intentions and trajectories in mixed traffic environment, IEEE Trans. Intell. Transp. Syst. (2023).
[60] J. Gehring, M. Auli, D. Grangier, Y.N. Dauphin, A convolutional encoder model for neural machine translation, arXiv preprint, arXiv:1611.02344, 2016.
[61] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: the kitti dataset, Int. J. Robot. Res. 32 (11) (2013) 1231–1237.
[62] P. Ghorai, A. Eskandarian, Y.-K. Kim, G. Mehr, State estimation and motion prediction of vehicles and vulnerable road users for cooperative autonomous driving: a survey, IEEE Trans. Intell. Transp. Syst. 23 (10) (2022) 16983–17002.
[63] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, F. Moutarde, Home: heatmap output for future motion estimation, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, 2021, pp. 500–507.
[64] T. Gindele, S. Brechtel, R. Dillmann, Learning driver behavior models from traffic observations for decision making and planning, IEEE Intell. Transp. Syst. Mag. 7 (1) (2015) 69–79.
[65] I. Gomes, D. Wolf, A review on intention-aware and interaction-aware trajectory prediction for autonomous vehicles, 2022.
[66] L. Gong, Q. Cheng, Exploiting edge features for graph neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9211–9219.
[67] D.S. González, O. Erkent, V. Romero-Cano, J. Dibangoye, C. Laugier, Modeling driver behavior from demonstrations in dynamic environments using spatiotemporal lattices, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 3384–3390.
[68] A. Graves, Generating sequences with recurrent neural networks, arXiv preprint, arXiv:1308.0850, 2013.
[69] J. Gu, C. Sun, H. Zhao, Densetnt: end-to-end trajectory prediction from dense goal sets, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15303–15312.
[70] Y. Guan, S.E. Li, J. Duan, W. Wang, B. Cheng, Markov probabilistic decision making of self-driving cars in highway with random traffic flow: a simulation study, J. Intell. Connect. Veh. 1 (2) (2018) 77–84.
[71] H. Guo, Q. Meng, X. Zhao, J. Liu, D. Cao, H. Chen, Map-enhanced generative adversarial trajectory prediction method for automated vehicles, Inf. Sci. 622 (2023) 1033–1049.
[72] Y. Guo, V.V. Kalidindi, M. Arief, W. Wang, J. Zhu, H. Peng, D. Zhao, Modeling multi-vehicle interaction scenarios using Gaussian random field, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2019, pp. 3974–3980.
[73] G. Han, J. Ma, S. Huang, L. Chen, S.-F. Chang, Few-shot object detection with fully cross-transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5321–5330.
[74] Z. Hao, X. Huang, K. Wang, M. Cui, Y. Tian, Attention-based gru for driver intention recognition and vehicle trajectory prediction, in: 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), IEEE, 2020, pp. 86–91.
[75] F. Hasan, H. Huang, Mals-net: a multi-head attention-based lstm sequence-to-sequence network for socio-temporal interaction modelling and trajectory prediction, Sensors 23 (1) (2023) 530.
[76] G. He, X. Li, Y. Lv, B. Gao, H. Chen, Probabilistic intention prediction and trajectory generation based on dynamic Bayesian networks, in: 2019 Chinese Automation Congress (CAC), IEEE, 2019, pp. 2646–2651.
[77] C. Hegde, S. Dash, P. Agarwal, Vehicle trajectory prediction using gan, in: 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE, 2020, pp. 502–507.
[78] C. Hermes, C. Wohler, K. Schenk, F. Kummert, Long-term vehicle motion prediction, in: 2009 IEEE Intelligent Vehicles Symposium, IEEE, 2009, pp. 652–657.
[79] L. Hewing, E. Arcari, L.P. Fröhlich, M.N. Zeilinger, On simulation and trajectory prediction with Gaussian process dynamics, in: Learning for Dynamics and Control, PMLR, 2020, pp. 424–434.
[80] B. Hjaltason, Predicting vehicle trajectories with inverse reinforcement learning, 2019.
[81] J. Ho, S. Ermon, Generative adversarial imitation learning, Adv. Neural Inf. Process. Syst. 29 (2016).
[82] L. Hou, S.E. Li, B. Yang, Z. Wang, K. Nakano, Structural transformer improves speed-accuracy trade-off in interactive trajectory prediction of multiple surrounding vehicles, IEEE Trans. Intell. Transp. Syst. 23 (12) (2022) 24778–24790.
[83] A. Houenou, P. Bonnifait, V. Cherfaoui, W. Yao, Vehicle trajectory prediction based on motion model and maneuver recognition, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 4363–4369.
[84] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, P. Ondruska, One thousand and one hours: self-driving motion prediction dataset, in: Conference on Robot Learning, PMLR, 2021, pp. 409–418.
[85] A. Hu, G. Corrado, N. Griffiths, Z. Murez, C. Gurau, H. Yeo, A. Kendall, R. Cipolla, J. Shotton, Model-based imitation learning for urban driving, Adv. Neural Inf. Process. Syst. 35 (2022) 20703–20716.
[86] D. Hu, An introductory survey on attention mechanisms in nlp problems, in: Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys), vol. 2, Springer, 2020, pp. 432–448.
[87] H. Hu, Q. Wang, M. Cheng, Z. Gao, Trajectory prediction neural network and model interpretation based on temporal pattern attention, in: IEEE Transactions on Intelligent Transportation Systems, 2022.

[88] Y. Hu, W. Zhan, L. Sun, M. Tomizuka, Multi-modal probabilistic prediction of interactive behavior via an interpretable model, in: 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2019, pp. 557–563.
[89] Y. Hu, W. Zhan, M. Tomizuka, Probabilistic prediction of vehicle semantic intention and motion, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 307–313.
[90] J. Huang, H.-S. Tan, Vehicle future trajectory prediction with a dgps/ins-based positioning system, in: 2006 American Control Conference, IEEE, 2006, p. 6.
[91] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, R. Yang, The apolloscape open dataset for autonomous driving and its application, IEEE Trans. Pattern Anal. Mach. Intell. 42 (10) (2019) 2702–2719.
[92] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, H. Chen, A survey on trajectory-prediction methods for autonomous driving, IEEE Trans. Intell. Veh. 7 (3) (2022) 652–674.
[93] Z. Huang, X. Mo, C. Lv, Multi-modal motion prediction with transformer-based neural network for autonomous driving, in: 2022 International Conference on Robotics and Automation (ICRA), IEEE, 2022, pp. 2605–2611.
[94] S. International, Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, SAE Int. 4970 (724) (2018) 1–5.
[95] H. Jeon, J. Choi, D. Kum, Scale-net: scalable vehicle trajectory prediction network under random number of interacting vehicles via edge-enhanced graph convolutional neural network, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 2095–2102.
[96] Y. Jiang, B. Zhu, S. Yang, J. Zhao, W. Deng, Vehicle trajectory prediction considering driver uncertainty and vehicle dynamics based on dynamic Bayesian network, in: IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022.
[97] A. Jindal, G.S. Aujla, N. Kumar, R. Chaudhary, M.S. Obaidat, I. You, Sedative: sdn-enabled deep learning architecture for network traffic control in vehicular cyber-physical systems, IEEE Netw. 32 (6) (2018) 66–73.
[98] C. Jung, D.H. Shim, Incorporating multi-context into the traversability map for urban autonomous driving using deep inverse reinforcement learning, IEEE Robot. Autom. Lett. 6 (2) (2021) 1662–1669.
[99] N. Kaempchen, B. Schiele, K. Dietmayer, Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios, IEEE Trans. Intell. Transp. Syst. 10 (4) (2009) 678–687.
[100] P. Karle, M. Geisslinger, J. Betz, M. Lienkamp, Scenario understanding and motion prediction for autonomous vehicles-review and comparison, in: IEEE Transactions on Intelligent Transportation Systems, 2022.
[101] V. Katariya, M. Baharani, N. Morris, O. Shoghli, H. Tabkhi, Deeptrack: lightweight deep learning for vehicle trajectory prediction in highways, IEEE Trans. Intell. Transp. Syst. 23 (10) (2022) 18927–18936.
[102] A. Kawasaki, A. Seki, Multimodal trajectory predictions for urban environments using geometric relationships between a vehicle and lanes, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 9203–9209.
[103] B. Kim, S.H. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J.S. Kim, J.W. Choi, Lapred: lane-aware prediction of multi-modal future trajectories of dynamic agents, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14636–14645.
[104] H. Kim, D. Kim, G. Kim, J. Cho, K. Huh, Multi-head attention based probabilistic vehicle trajectory prediction, in: 2020 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2020, pp. 1720–1725.
[105] S. Kim, H. Jeon, J.W. Choi, D. Kum, Diverse multiple trajectory prediction using a two-stage prediction network trained with lane loss, in: IEEE Robotics and Automation Letters, 2022.
[106] B.R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A.A. Al Sallab, S. Yogamani, P. Pérez, Deep reinforcement learning for autonomous driving: a survey, IEEE Trans. Intell. Transp. Syst. 23 (6) (2021) 4909–4926.
[107] R. Krajewski, J. Bock, L. Kloeker, L. Eckstein, The highd dataset: a drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 2118–2125.
[108] A. Kuefler, J. Morton, T. Wheeler, M. Kochenderfer, Imitating driver behavior with generative adversarial networks, in: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017, pp. 204–211.
[109] N. Kumar, N. Chilamkurti, J.J. Rodrigues, Learning automata-based opportunistic data aggregation and forwarding scheme for alert generation in vehicular ad hoc networks, Comput. Commun. 39 (2014) 22–32.
[110] N. Kumar, R. Iqbal, S. Misra, J.J. Rodrigues, Bayesian coalition game for contention-aware reliable data forwarding in vehicular mobile cloud, Future Gener. Comput. Syst. 48 (2015) 60–72.
[111] C. Laugier, I.E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, A. Nègre, Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety, IEEE Intell. Transp. Syst. Mag. 3 (4) (2011) 4–19.
[112] S. Lefèvre, D. Vasquez, C. Laugier, A survey on motion prediction and risk assessment for intelligent vehicles, ROBOMECH J. 1 (1) (2014) 1–14.
[113] V. Lefkopoulos, M. Menner, A. Domahidi, M.N. Zeilinger, Interaction-aware motion prediction for autonomous driving: a multiple model Kalman filtering scheme, IEEE Robot. Autom. Lett. 6 (1) (2020) 80–87.
[114] F. Leon, M. Gavrilescu, A review of tracking and trajectory prediction methods for autonomous driving, Mathematics 9 (6) (2021) 660.
[115] D. Li, H. Li, Y. Xiao, B. Li, B. Tang, Vehicle trajectory prediction for automated driving based on temporal convolution networks, in: 2022 WRC Symposium on Advanced Robotics and Automation (WRC SARA), IEEE, 2022, pp. 257–262.
[116] J. Li, B. Dai, X. Li, X. Xu, D. Liu, A dynamic Bayesian network for vehicle maneuver prediction in highway driving scenarios: framework and verification, Electronics 8 (1) (2019) 40.
[117] T. Li, L. Chen, Y. Wang, Autonomous driving behavior prediction method based on improved hidden Markov model, in: 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), IEEE, 2022, pp. 758–762.
[118] X. Li, G. Rosman, I. Gilitschenski, C.-I. Vasile, J.A. DeCastro, S. Karaman, D. Rus, Vehicle trajectory prediction using generative adversarial network with temporal logic syntax tree features, IEEE Robot. Autom. Lett. 6 (2) (2021) 3459–3466.
[119] X. Li, X. Ying, M.C. Chuah, Grip++: enhanced graph-based interaction-aware trajectory prediction for autonomous driving, arXiv preprint, arXiv:1907.07792, 2019.
[120] X. Li, X. Ying, M.C. Chuah, Grip: graph-based interaction-aware trajectory prediction, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2019, pp. 3960–3966.
[121] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, R. Urtasun, Learning lane graph representations for motion forecasting, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, in: Proceedings, Part II, vol. 16, Springer, 2020, pp. 541–556.
[122] C.-F. Lin, A.G. Ulsoy, D.J. LeBlanc, Vehicle dynamics and external disturbance estimation for vehicle path prediction, IEEE Trans. Control Syst. Technol. 8 (3) (2000) 508–518.
[123] L. Lin, W. Li, H. Bi, L. Qin, Vehicle trajectory prediction using lstms with spatial–temporal attention mechanisms, IEEE Intell. Transp. Syst. Mag. 14 (2) (2021) 197–208.
[124] J. Liu, X. Mao, Y. Fang, D. Zhu, M.Q.-H. Meng, A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving, in: 2021 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2021, pp. 978–985.
[125] X. Liu, Y. Wang, K. Jiang, Z. Zhou, K. Nam, C. Yin, Interactive trajectory prediction using a driving risk map-integrated deep learning method for surrounding vehicles on highways, IEEE Trans. Intell. Transp. Syst. 23 (10) (2022) 19076–19087.
[126] Y. Liu, X. Qi, E.A. Sisbot, K. Oguchi, Multi-agent trajectory prediction with graph attention isomorphism neural network, in: 2022 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2022, pp. 273–279.
[127] Y. Liu, J. Zhang, L. Fang, Q. Jiang, B. Zhou, Multimodal motion prediction with stacked transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7577–7586.
[128] S.S. Lodh, N. Kumar, P.K. Pandey, Autonomous vehicular overtaking maneuver: a survey and taxonomy, Veh. Commun. (2023) 100623.
[129] P. Lytrivis, G. Thomaidis, A. Amditis, Cooperative path prediction in vehicular environments, in: 2008 11th International IEEE Conference on Intelligent Transportation Systems, IEEE, 2008, pp. 803–808.
[130] F. Marchetti, F. Becattini, L. Seidenari, A.D. Bimbo, Mantra: memory augmented networks for multiple trajectory prediction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7143–7152.
[131] Q. Meng, H. Guo, Y. Liu, H. Chen, D. Cao, Trajectory prediction for automated vehicles on roads with lanes partially covered by ice or snow, IEEE Trans. Veh. Technol. (2023).
[132] Q. Meng, B. Shang, Y. Liu, H. Guo, X. Zhao, Intelligent vehicles trajectory prediction with spatial and temporal attention mechanism, IFAC-PapersOnLine 54 (10) (2021) 454–459.
[133] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, G.P. Gil, Multi-head attention for multi-modal joint vehicle motion forecasting, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 9638–9644.
[134] K. Messaoud, N. Deo, M.M. Trivedi, F. Nashashibi, Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation, in: 2021 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2021, pp. 165–170.
[135] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, F. Nashashibi, Attention based vehicle trajectory prediction, IEEE Trans. Intell. Veh. 6 (1) (2020) 175–185.
[136] K. Min, D. Kim, J. Park, K. Huh, Rnn-based path prediction of obstacle vehicles with deep ensemble, IEEE Trans. Veh. Technol. 68 (10) (2019) 10252–10256.
[137] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, Adv. Neural Inf. Process. Syst. 27 (2014).
[138] X. Mo, Z. Huang, Y. Xing, C. Lv, Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network, IEEE Trans. Intell. Transp. Syst. 23 (7) (2022) 9554–9567.
[139] X. Mo, Y. Xing, C.Lv. Recog, A deep learning framework with heterogeneous graph for interaction-aware trajectory prediction, arXiv preprint, arXiv:2012.05032, 2020.
[140] S. Mozaffari, O.Y. Al-Jarrah, M. Dianati, P. Jennings, A. Mouzakitis, Deep learning-based vehicle behavior prediction for autonomous driving applications: a review, IEEE Trans. Intell. Transp. Syst. 23 (1) (2020) 33–47.
[141] K.P. Murphy, Dynamic Bayesian Networks: Representation, Inference and Learning, University of California, Berkeley, 2002.
[142] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T.L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, et al., Scene transformer: a unified architecture for

predicting future trajectories of multiple agents, in: International Conference on Learning Representations, 2022.
[143] N. Nikhil, B. Tran Morris, Convolutional neural network for trajectory prediction, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018.
[144] K. Okamoto, K. Berntorp, S. Di Cairano, Driver intention-based vehicle threat assessment using random forests and particle filtering, IFAC-PapersOnLine 50 (1) (2017) 13860–13865.
[145] A. Pandey, D. Wang, Tcnn: temporal convolutional neural network for real-time speech enhancement in the time domain, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 6875–6879.
[146] R. Pepy, A. Lambert, H. Mounier, Reducing navigation errors by planning with realistic vehicle model, in: 2006 IEEE Intelligent Vehicles Symposium, IEEE, 2006, pp. 300–307.
[147] T. Phan-Minh, E.C. Grigore, F.A. Boulton, O. Beijbom, E.M. Wolff, Covernet: multimodal behavior prediction using trajectory sets, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14074–14083.
[148] T. Phan-Minh, F. Howington, T.-S. Chu, S.U. Lee, M.S. Tomov, N. Li, C. Dicle, S. Findler, F. Suarez-Ruiz, R. Beaudoin, et al., Driving in real life with inverse reinforcement learning, arXiv preprint, arXiv:2206.03004, 2022.
[149] D.J. Phillips, T.A. Wheeler, M.J. Kochenderfer, Generalizable intention prediction of human drivers at intersections, in: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017, pp. 1665–1670.
[150] A. Polychronopoulos, M. Tsogas, A.J. Amditis, L. Andreone, Sensor fusion for predicting vehicles' path for collision avoidance systems, IEEE Trans. Intell. Transp. Syst. 8 (3) (2007) 549–562.
[151] D.A. Pomerleau, Alvinn: an autonomous land vehicle in a neural network, Adv. Neural Inf. Process. Syst. 1 (1988).
[152] S. Qiao, D. Shen, X. Wang, N. Han, W. Zhu, A self-adaptive parameter selection trajectory prediction approach via hidden Markov models, IEEE Trans. Intell. Transp. Syst. 16 (1) (2014) 284–296.
[153] A. Quintanar, D. Fernández-Llorca, I. Parra, R. Izquierdo, M. Sotelo, Predicting vehicles trajectories in urban scenarios with transformer networks and augmented information, in: 2021 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2021, pp. 1051–1056.
[154] R. Rajamani, Vehicle Dynamics and Control, Springer Science & Business Media, 2011.
[155] Y.-Y. Ren, L. Zhao, X.-L. Zheng, X.-S. Li, A method for predicting diverse lane-changing trajectories of surrounding vehicles based on early detection of lane change, IEEE Access 10 (2022) 17451–17472.
[156] G. Rong, B.H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, et al., Lgsvl simulator: a high fidelity simulator for autonomous driving, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2020, pp. 1–6.
[157] A. Sachan, N.S. Chauhan, N. Kumar, Congestion minimization using fog-deployed drl-agent feedback enabled traffic light cooperative framework, in: 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid), IEEE, 2023, pp. 557–567.
[158] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[159] M. Schreiber, S. Hoermann, K. Dietmayer, Long-term occupancy grid prediction using recurrent neural networks, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 9299–9305.
[160] M. Schreier, V. Willert, J. Adamy, An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments, IEEE Trans. Intell. Transp. Syst. 17 (10) (2016) 2751–2766.
[161] R. Schubert, E. Richter, G. Wanielik, Comparison and evaluation of advanced motion models for vehicle tracking, in: 2008 11th International Conference on Information Fusion, IEEE, 2008, pp. 1–6.
[162] H.G. Seif, X. Hu, Autonomous driving in the icity—hd maps as a key challenge of the automotive industry, Engineering 2 (2) (2016) 159–162.
[163] A. Sharma, D.B. Jayagopi, Towards efficient unconstrained handwriting recognition using dilated temporal convolution network, Expert Syst. Appl. 164 (2021) 114004.
[164] Z. Sheng, Y. Xu, S. Xue, D. Li, Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving, IEEE Trans. Intell. Transp. Syst. 23 (10) (2022) 17654–17665.
[165] M.S. Shirazi, B.T. Morris, Looking at intersections: a survey of intersection monitoring, behavior and safety analysis of recent studies, IEEE Trans. Intell. Transp. Syst. 18 (1) (2016) 4–24.
[166] D. Silver, J. Veness, Monte-Carlo planning in large pomdps, Adv. Neural Inf. Process. Syst. 23 (2010).
[167] A. Somani, N. Ye, D. Hsu, W.S. Lee, Despot: online pomdp planning with regularization, Adv. Neural Inf. Process. Syst. 26 (2013).
[168] H. Song, D. Luan, W. Ding, M.Y. Wang, Q. Chen, Learning to predict vehicle trajectories with model-based planning, in: Conference on Robot Learning, PMLR, 2022, pp. 1035–1045.
[169] D. Soni, N. Kumar, Machine learning techniques in emerging cloud computing integrated paradigms: a survey and taxonomy, J. Netw. Comput. Appl. 205 (2022) 103419.
[170] N. Sriram, B. Liu, F. Pittaluga, M. Chandraker, Smart: simultaneous multi-agent recurrent trajectory prediction, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, in: Proceedings, Part XXVII, vol. 16, Springer, 2020, pp. 463–479.
[171] J. Strohbeck, V. Belagiannis, J. Müller, M. Schreiber, M. Herrmann, D. Wolf, M. Buchholz, Multiple trajectory prediction with deep temporal and spatial convolutional neural networks, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 1992–1998.
[172] L. Sun, W. Zhan, M. Tomizuka, Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 2111–2117.
[173] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, Adv. Neural Inf. Process. Syst. 27 (2014).
[174] C. Tang, R.R. Salakhutdinov, Multiple futures prediction, Adv. Neural Inf. Process. Syst. 32 (2019).
[175] S. Targ, D. Almeida, K. Lyman, Resnet in resnet: generalizing residual architectures, arXiv preprint, arXiv:1603.08029, 2016.
[176] I. Teeti, S. Khan, A. Shahbaz, A. Bradley, F. Cuzzolin, Vision-based intention and trajectory prediction in autonomous vehicles: a survey, in: Lud De Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, vol. 7, 2022, pp. 5630–5637.
[177] Q. Tran, J. Firl, Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression, in: 2014 Ieee Intelligent Vehicles Symposium Proceedings, IEEE, 2014, pp. 918–923.
[178] T.-D. Tran, X.-T. Vo, D.-L. Nguyen, K.-H. Jo, Combination of deep learner network and transformer for 3d human pose estimation, in: 2022 22nd International Conference on Control, Automation and Systems (ICCAS), IEEE, 2022, pp. 174–178.
[179] P. Trautman, A. Krause, Unfreezing the robot: navigation in dense, interacting crowds, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 797–803.
[180] I.M. Varma, N. Kumar, A comprehensive survey on sdn and blockchain-based secure vehicular networks, Veh. Commun. (2023) 100663.
[181] D. Vasquez, T. Fraichard, Motion prediction for moving objects: a statistical approach, in: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04, 2004, vol. 4, IEEE, 2004, pp. 3931–3936.
[182] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[183] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, F.-Y. Wang, Generative adversarial networks: introduction and outlook, IEEE/CAA J. Autom. Sin. 4 (4) (2017) 588–598.
[184] P. Wang, C.-Y. Chan, A. de La Fortelle, A reinforcement learning based approach for automated lane change maneuvers, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 1379–1384.
[185] X. Wang, K. Tang, X. Dai, J. Xu, J. Xi, R. Ai, Y. Wang, W. Gu, C. Sun, Safety-balanced driving-style aware trajectory planning in intersection scenarios with uncertain environment, IEEE Trans. Intell. Veh. (2023).
[186] Y. Wang, S. Chen, Multi-agent trajectory prediction with spatio-temporal sequence fusion, IEEE Trans. Multimed. (2021).
[187] Y. Wang, Z. Liu, Z. Zuo, Z. Li, L. Wang, X. Luo, Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control, IEEE Trans. Veh. Technol. 68 (9) (2019) 8546–8556.
[188] Y. Wang, C. Wang, W. Zhao, C. Xu, Decision-making and planning method for autonomous vehicles based on motivation and risk assessment, IEEE Trans. Veh. Technol. 70 (1) (2021) 107–120.
[189] Y. Wang, S. Zhao, R. Zhang, X. Cheng, L. Yang, Multi-vehicle collaborative learning for trajectory prediction with spatio-temporal tensor fusion, IEEE Trans. Intell. Transp. Syst. 23 (1) (2020) 236–248.
[190] Z. Wang, J. Guo, Z. Hu, H. Zhang, J. Zhang, J. Pu, Lane transformer: a high efficiency trajectory prediction model, IEEE Open J. Intell. Transp. Syst. (2023).
[191] M. Wazid, A.K. Das, N. Kumar, V. Odelu, A.G. Reddy, K. Park, Y. Park, Design of lightweight authentication and key agreement protocol for vehicular ad hoc networks, IEEE Access 5 (2017) 14966–14980.
[192] X. Weng, B. Ivanovic, K. Kitani, M. Pavone, Whose track is it anyway? Improving robustness to tracking errors with affinity-based trajectory prediction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6573–6582.
[193] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J.K. Pontes, et al., Argoverse 2: next generation datasets for self-driving perception and forecasting, arXiv preprint, arXiv:2301.00493, 2023.
[194] F. Wirthmüller, J. Schlechtriemen, J. Hipp, M. Reichert, Teaching vehicles to anticipate: a systematic study on probabilistic behavior prediction using large data sets, IEEE Trans. Intell. Transp. Syst. 22 (11) (2020) 7129–7144.
[195] C. Wissing, T. Nattermann, K.-H. Glander, T. Bertram, Interaction-aware long-term driving situation prediction, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 137–143.
[196] Y. Wu, G. Chen, Z. Li, L. Zhang, L. Xiong, Z. Liu, A. Knoll, Hsta: a hierarchical spatio-temporal attention model for trajectory prediction, IEEE Trans. Veh. Technol. 70 (11) (2021) 11295–11307.

[197] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2020) 4–24.
[198] Z. Wu, L. Sun, W. Zhan, C. Yang, M. Tomizuka, Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving, IEEE Robot. Autom. Lett. 5 (4) (2020) 5355–5362.
[199] M. Wulfmeier, P. Ondruska, I. Posner, Maximum entropy deep inverse reinforcement learning, arXiv preprint, arXiv:1507.04888, 2015.
[200] M. Wulfmeier, D.Z. Wang, I. Posner, Watch this: scalable cost-function learning for path planning in urban environments, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2016, pp. 2089–2095.
[201] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, J. Wang, Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models, IEEE Trans. Ind. Electron. 65 (7) (2017) 5999–6008.
[202] G. Xie, A. Shangguan, R. Fei, W. Ji, W. Ma, X. Hei, Motion trajectory prediction based on a cnn-lstm sequential model, Sci. China Inf. Sci. 63 (2020) 1–21.
[203] X. Xie, C. Zhang, Y. Zhu, Y.N. Wu, S.-C. Zhu, Congestion-aware multi-agent trajectory prediction for collision avoidance, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021, pp. 13693–13700.
[204] L. Xin, S.E. Li, P. Wang, W. Cao, B. Nie, C.-Y. Chan, B. Cheng, Accelerated inverse reinforcement learning with randomly pre-sampled policies for autonomous driving reward design, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2019, pp. 2757–2764.
[205] L. Xin, P. Wang, C.-Y. Chan, J. Chen, S.E. Li, B. Cheng, Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 1441–1446.
[206] Y. Xing, C. Lv, D. Cao, Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles, IEEE Trans. Veh. Technol. 69 (2) (2019) 1341–1352.
[207] D. Xu, Z. Ding, X. He, H. Zhao, M. Moze, F. Aioun, F. Guillemard, Learning from naturalistic driving data for human-like autonomous highway driving, IEEE Trans. Intell. Transp. Syst. 22 (12) (2020) 7341–7354.
[208] D. Xu, X. Shang, Y. Liu, H. Peng, H. Li, Group vehicle trajectory prediction with global spatio-temporal graph, IEEE Trans. Intell. Veh. (2022).
[209] D. Xu, X. Shang, H. Peng, H. Li, Mvhgn: multi-view adaptive hierarchical spatial graph convolution network based trajectory prediction for heterogeneous traffic-agents, in: IEEE Transactions on Intelligent Transportation Systems, 2023.
[210] X. Xu, X. Wang, Y. Wang, J. Cao, Vehicle trajectory prediction considering multi-feature independent encoding, Available at SSRN 4135360.
[211] Y. Xu, T. Zhao, C. Baker, Y. Zhao, Y.N. Wu, Learning trajectory prediction with continuous inverse optimal control via Langevin sampling of energy-based models, arXiv preprint, arXiv:1904.05453, 2019.
[212] J. Yan, Z. Peng, H. Yin, J. Wang, X. Wang, Y. Shen, W. Stechele, D. Cremers, Trajectory prediction for intelligent vehicles using spatial-attention mechanism, IET Intell. Transp. Syst. 14 (13) (2020) 1855–1863.
[213] S. Yang, Y. Chen, Y. Cao, R. Wang, R. Shi, J. Lu, Lane change trajectory prediction based on spatiotemporal attention mechanism, in: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2022, pp. 2366–2371.
[214] M. Ye, T. Cao, Q. Chen, Tpcn: temporal point cloud networks for motion forecasting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11318–11327.
[215] C. You, J. Lu, D. Filev, P. Tsiotras, Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning, Robot. Auton. Syst. 114 (2019) 1–18.
[216] J. Yu, M. Zhou, X. Wang, G. Pu, C. Cheng, B. Chen, A dynamic and static context-aware attention network for trajectory prediction, ISPRS Int. J. Geo-Inf. 10 (5) (2021) 336.
[217] W. Zeng, M. Liang, R. Liao, R. Urtasun, Lanercnn: distributed representations for graph-centric motion forecasting, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2021, pp. 532–539.
[218] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, et al., Interaction dataset: an international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps, arXiv preprint, arXiv:1910.03088, 2019.
[219] H. Zhang, Y. Wang, J. Liu, C. Li, T. Ma, C. Yin, A multi-modal states based vehicle descriptor and dilated convolutional social pooling for vehicle trajectory prediction, arXiv preprint, arXiv:2003.03480, 2020.
[220] K. Zhang, X. Feng, L. Wu, Z. He, Trajectory prediction for autonomous driving using spatial-temporal graph attention transformer, IEEE Trans. Intell. Transp. Syst. 23 (11) (2022) 22343–22353.
[221] K. Zhang, L. Zhao, C. Dong, L. Wu, L. Zheng, Ai-tp: attention-based interaction-aware trajectory prediction for autonomous driving, IEEE Trans. Intell. Veh. (2022).
[222] R. Zhang, L. Cao, S. Bao, J. Tan, A method for connected vehicle trajectory prediction and collision warning algorithm based on v2v communication, Int. J. Crashworthiness 22 (1) (2017) 15–25.
[223] S. Zhang, Y. Zhi, R. He, J. Li, Research on traffic vehicle behavior prediction method based on game theory and hmm, IEEE Access 8 (2020) 30210–30222.
[224] T. Zhang, W. Song, M. Fu, Y. Yang, M. Wang, Vehicle motion prediction at intersections based on the turning intention and prior trajectories model, IEEE/CAA J. Autom. Sin. 8 (10) (2021) 1657–1666.
[225] Y. Zhang, Y. Zou, J. Tang, J. Liang, A lane-changing prediction method based on temporal convolution network, arXiv preprint, arXiv:2011.01224, 2020.
[226] Z. Zhang, Resnet-based model for autonomous vehicles trajectory prediction, in: 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE), IEEE, 2021, pp. 565–568.
[227] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, et al., Tnt: target-driven trajectory prediction, in: Conference on Robot Learning, PMLR, 2021, pp. 895–904.
[228] J. Zhao, X. Li, Q. Xue, W. Zhang, Spatial-channel transformer network for trajectory prediction on the traffic scenes, arXiv preprint, arXiv:2101.11472, 2021.
[229] L. Zhao, Y. Liu, A.Y. Al-Dubai, A.Y. Zomaya, G. Min, A. Hawbani, A novel generation-adversarial-network-based vehicle trajectory prediction method for intelligent vehicular networks, IEEE Int. Things J. 8 (3) (2020) 2066–2077.
[230] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, Y.N. Wu, Multi-agent tensor fusion for contextual trajectory prediction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12126–12134.
[231] Z. Zhao, H. Fang, Z. Jin, Q. Qiu, Gisnet: graph-based information sharing network for vehicle trajectory prediction, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–7.
[232] Z. Zheng, Z. Zhang, L. Wang, X. Luo, Denoising temporal convolutional recurrent autoencoders for time series classification, Inf. Sci. 588 (2022) 159–173.
[233] Z. Zhou, M.M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, Unet++: a nested u-net architecture for medical image segmentation, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, in: Proceedings, vol. 4, Springer, 2018, pp. 3–11.
[234] Z. Zhu, N. Li, R. Sun, D. Xu, H. Zhao, Off-road autonomous vehicles traversability analysis and trajectory planning based on deep inverse reinforcement learning, in: 2020 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2020, pp. 971–977.
[235] B.D. Ziebart, A.L. Maas, J.A. Bagnell, A.K. Dey, et al., Maximum entropy inverse reinforcement learning, in: Aaai, Chicago, IL, USA, vol. 8, 2008, pp. 1433–1438.
[236] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C.G. Keller, et al., Making bertha drive—an autonomous journey on a historic route, IEEE Intell. Transp. Syst. Mag. 6 (2) (2014) 8–20.
[237] Q. Zou, H. Li, R. Zhang, Inverse reinforcement learning via neural network in driver behavior modeling, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 1245–1250.
[238] A. Zyner, S. Worrall, E. Nebot, A recurrent neural network solution for predicting driver intention at unsignalized intersections, IEEE Robot. Autom. Lett. 3 (3) (2018) 1759–1764.
[239] A. Zyner, S. Worrall, E. Nebot, Naturalistic driver intention and path prediction using recurrent neural networks, IEEE Trans. Intell. Transp. Syst. 21 (4) (2019) 1584–1594.
[240] A. Zyner, S. Worrall, J. Ward, E. Nebot, Long short term memory for driver intent prediction, in: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017, pp. 1484–1489.
[241] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint, arXiv:1710.10903, 2017.