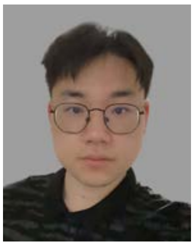
UDC 004.896:004.942

ANALYSIS AND REVIEW OF MULTIMODAL TRAJECTORY PREDICTION METHODS IN COMPLEX DYNAMIC SCENES: EVOLUTION FROM CLASSICAL MODELS TO DEEP LEARNING



Yi Tang
Ph.D. student from the
Belarusian State University
of Informatics and
Radioelectronics
tangyijcb@163.com

Yi Tang
A Ph.D. student from the Belarusian State University of Informatics and Radioelectronics. The research area is about autonomous vehicle under dynamic and complex environments.

Abstract. This article provides a systematic review of the mainstream algorithms and methods in the field of trajectory prediction for autonomous vehicles, categorizing them into four major approaches: traditional statistical methods, machine learning-based methods, deep learning-based methods, and hybrid models. Through a comprehensive analysis of the principles, strengths, weaknesses, and relevant literature of each approach, this study offers a detailed comparison of their performance characteristics and delves into their respective application scenarios. Furthermore, based on the current state of research, the article explores future directions for trajectory prediction technologies and proposes corresponding research recommendations.

Keywords: Trajectory Prediction; Autonomous driving; Machine learning; Deep learning.

Introduction. With the continuous growth in global vehicle ownership, issues such as traffic congestion, environmental pollution, and road safety have become increasingly severe. The emergence of autonomous driving technology offers new solutions to these challenges. By integrating advanced sensors, artificial intelligence algorithms, and communication technologies, autonomous vehicles can achieve self-navigation, path planning, and environmental perception, thereby improving traffic efficiency, reducing human driving errors, and lowering energy consumption [1]. Among these technologies, trajectory prediction is a critical component of autonomous driving systems. It enables the vehicle to anticipate the motion trends of surrounding traffic participants, providing the decision-making system with sufficient time to plan safe and efficient driving paths. This capability effectively mitigates potential collision risks and enhances the overall performance and reliability of autonomous vehicles.

In recent years, significant progress has been made in the field of vehicle trajectory prediction, with researchers proposing a variety of prediction methods that can be broadly categorized into four main types as illustrated in Figure 1: traditional statistical methods, machine learning-based methods, deep learning-based methods, and hybrid models. Traditional statistical methods, such as Kalman filtering and Monte Carlo methods, have been widely used in early trajectory prediction research due to their computational efficiency and simplicity. However, these methods exhibit notable limitations when dealing with complex traffic scenarios and vehicle interactions. Machine learning methods, such as Support Vector Machines (SVM), Gaussian

Processes (GP), and Hidden Markov Models (HMM), learn patterns and rules from historical data, making them more effective in handling complex traffic scenarios. Nonetheless, they require high-quality labeled data, extensive feature engineering, and are sensitive to data quality. Deep learning methods, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Graph Neural Networks (GNNs), excel in feature extraction and can handle large-scale datasets [2]. They demonstrate outstanding prediction accuracy, particularly in complex traffic scenarios. However, these methods come with high model complexity, long training times, and significant computational resource demands. Hybrid models combine the strengths of traditional statistical methods, machine learning, and deep learning, further improving prediction accuracy and robustness. Nevertheless, their model construction and optimization processes are complex, with substantial computational overhead.

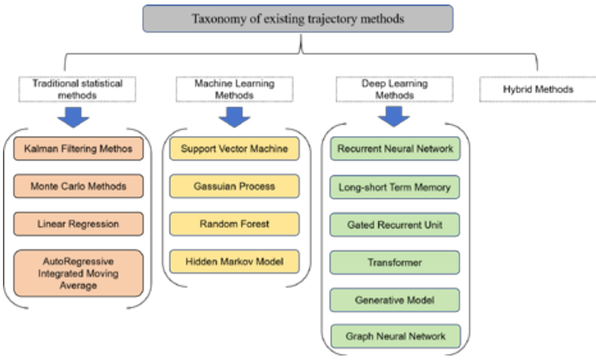


Figure 1. Classification of existing trajectory prediction methods

This study aims to comprehensively analyze and compare different trajectory prediction methods, exploring their underlying principles, strengths, weaknesses, and applicable scenarios. Through a systematic review of the existing literature, this paper summarizes the latest advancements and research outcomes in various methods, providing a comprehensive reference for researchers and technology developers. Finally, the paper discusses future directions for trajectory prediction technologies and proposes corresponding research recommendations, offering theoretical guidance and practical insights for subsequent research.

Task Definition. The task of trajectory prediction focuses on estimating possible future trajectories for target agents based on their observed motion history and surrounding environmental context. In the context of autonomous driving scenarios involving multiple dynamic agents (including autonomous vehicles), the surrounding map information is denoted as $\mathbf{M}$, while the observed trajectories of all agents are collectively represented as $\mathbf{X}=\left\{\mathbf{x}_0,\ldots,\mathbf{x}_{N_a}\right\}$. Specifically, $\mathbf{x}_i=\left\{\mathbf{x}_i^{-H+1},\ldots,\mathbf{x}_i^0\right\}$ denotes the historical trajectory of the agent over the preceding time steps. To address the inherent multimodal nature of future motion, a multi-agent motion predictor is implemented to generate plausible future trajectories for all agents in the scene, represented as $\mathbf{Y}=\left\{\mathbf{y}_0,\ldots,\mathbf{y}_{N_a}\right\}$. For each individual agent, possible future trajectories and their associated probability scores are predicted, enabling the modeling of diverse potential outcomes. These multimodal trajectories are expressed as $\mathbf{y}_i=\left\{\mathbf{y}_i^1,\ldots,\mathbf{y}_i^k\right\}$, where $\mathbf{y}_i^k=\left\{\mathbf{y}_{i,1}^k,\ldots,\mathbf{y}_{i,T}^k\right\}$ represents the predicted trajectory for the agent over a prediction horizon. The probability scores corresponding to these predicted trajectories are captured as $\mathbf{s}_i=\left\{s_i^1,\ldots,s_i^k\right\}$, providing a quantitative measure of their likelihood.

Table 1. Comparison of commonly used trajectory prediction datasets. Here ‘size’ denotes the number of trajectories in the dataset; ‘History’ and ‘Prediction’ denote the observed and future trajectories for each agent; ‘HD Map’ stands for High-definition maps that include lane markings, traffic rules, and other semantic information.

Dataset	Type	Size	History	Prediction	Participant	Environment
NGSIM [3]	Highway	6,000+	3s	5s	Vehicles	-
HighD [4]	Highway	110,000	3s	5s	Vehicles	-
Argoverse [5]	Urban	324,557	2s	3s	Vehicles	HD Map
INTERACTION [6]	Urban	11,000+	2s	3s	Vehicles, Pedestrians	HD Map
ETH/UCY [7]	Pedestrian	1,536	3.2s	4.8s	Pedestrians	-
Lyft Level 5 [8]	Urban	1,000+	1s	3s	Vehicles	HD Map
ApolloScape [9]	Urban	100,000+	2s	3s	Vehicles, Pedestrians	HD Map
NuScenes [10]	Urban	40,000	2s	4s	Vehicles, Pedestrians, Cyclists	HD Map

Dataset description and comparison. The choice of datasets in trajectory prediction research is critical for evaluating algorithm performance and training models effectively. Trajectory prediction datasets can be broadly classified into three categories based on their focus and characteristics: Highway Datasets, Urban Datasets, and Pedestrian Datasets. Highway Datasets (e.g., NGSIM, HighD) focus on vehicle behavior in highway scenarios, featuring simple traffic interactions, longer historical and prediction times (e.g., 3s history, 5s prediction), and are ideal for studying long-term vehicle maneuvers and lane changes. Urban Datasets (e.g., Argoverse, INTERACTION, Lyft Level 5, ApolloScape, NuScenes) emphasize complex traffic interactions in urban environments, characterized by shorter historical and prediction times (e.g., 1s–2s history, 3s prediction), inclusion of multiple traffic participants (vehicles, pedestrians, cyclists), and often provide HD maps for contextual information, making them suitable for studying multi-agent interactions and collision avoidance. Pedestrian Datasets (e.g., ETH/UCY) are dedicated to pedestrian movements in open spaces, featuring larger time steps (e.g., 0.4s) due to slower pedestrian speeds, a focus on social interactions and group behavior, and are ideal for studying pedestrian trajectory prediction and crowd dynamics. Table 1 presents a comparative analysis of commonly used trajectory prediction datasets, highlighting their key characteristics and differences.

Evaluation metrics. In trajectory prediction tasks, evaluation metrics are used to quantify the differences between the predicted results and the ground truth trajectories. Trajectory prediction performance metrics are essential for evaluating the accuracy and reliability of predictive models. Commonly used metrics include the Final Displacement Error (FDE), which measures the Euclidean distance between the predicted and true trajectory endpoints, and the Average Displacement Error (ADE), defined as the mean Euclidean distance between the predicted and actual trajectories over all time steps. Additionally, the Root Mean Square Error (RMSE) quantifies the square root of the average squared differences between predicted and true trajectory points, whereas the Mean Absolute Error (MAE) calculates the mean of absolute differences, providing a more intuitive measure of prediction deviation. Furthermore, the Miss Rate (MR) evaluates the proportion of predicted trajectory endpoints that deviate from the true endpoints by more than a predefined threshold, serving as a critical indicator of model reliability in safety-critical applications such as autonomous driving [11]. These metrics collectively offer a comprehensive framework for assessing both endpoint prediction accuracy and overall trajectory consistency. Below are commonly used evaluation metrics and their corresponding calculation formulas are shown below in formula (1):

$$\begin{cases} \mathrm{FDE} = \sqrt{(x_T - \hat{x}_T)^2 + (y_T - \hat{y}_T)^2} \\ \mathrm{ADE} = \dfrac{1}{T}\sum_{t=1}^{T}\sqrt{(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2} \\ \mathrm{RMSE} = \sqrt{\dfrac{1}{T}\sum_{t=1}^{T}\left[(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2\right]} \\ \mathrm{MAE} = \dfrac{1}{T}\sum_{t=1}^{T}\left[|x_t - \hat{x}_t| + |y_t - \hat{y}_t|\right] \\ \mathrm{MR} = \dfrac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\mathrm{FDE}_i > \mathrm{thres}\mathit{hold}) \end{cases} \tag{1}$$

Where (x_t, y_t) represents the position of the true trajectory at time step t, and $(\hat{x}_t, \hat{y}_t)$ represents the position of the predicted trajectory at time step t, with $t \in (0, T]$, where T is the total number of time steps. Additionally, N denotes the total number of samples, and $\mathbb{I}(\cdot)$ is an indicator function. The value is 1 if the condition is true, else 0.

Traditional statistical methods. Kalman filtering, linear regression, time series models (e.g., ARIMA), and Monte Carlo simulation are widely used methods in early trajectory prediction research. These approaches predict future trends based on historical trajectory data and are favored for their computational efficiency, simplicity, and ease of implementation [12]. They are particularly suitable for handling low-dimensional and structured trajectory data scenarios.

For example, the Kalman filter effectively addresses the uncertainty in vehicle states by recursively updating the state estimates. It models the uncertainty of the current vehicle state and its physical dynamics using Gaussian distributions, producing the mean and covariance matrix of the vehicle state at each future time step. This allows the calculation of an average trajectory with associated uncertainties. While the Kalman filter accounts for the uncertainty of predicted trajectories, its unimodal Gaussian distribution is insufficient to represent diverse operational modes.

The Monte Carlo method, on the other hand, generates potential future trajectories by randomly sampling input variables and applying physical models to approximate the state distribution. When generating trajectory samples, constraints such as limiting lateral acceleration below feasible thresholds or considering the physical limitations of the vehicle are applied to ensure the feasibility of operations. This method can predict the trajectories of traffic participants from either fully known states or uncertain states estimated through filtering algorithms.

Linear regression is one of the simplest and most intuitive methods among traditional statistical approaches. It constructs a predictive model by assuming a linear relationship between the dependent variable (target variable) and the independent variables (feature variables). For trajectory prediction, linear regression can fit the motion trend of a vehicle based on historical trajectory data (e.g., position, velocity, time) and predict its future positions. The mathematical expression is below in (2):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon \tag{2}$$

Where y represents the target variable (e.g., future position), $x_1, x_2, \ldots, x_n$ represent the feature variable (e.g., time or velocity), $\beta_1, \beta_2, \ldots, \beta_n$ denote the regression coefficients, and ε is the error term. The advantage of linear regression lies in its simplicity and interpretability, making it suitable for structured trajectory data. However, as it assumes a linear relationship, it struggles to capture nonlinear motion patterns in complex traffic scenarios and thus performs limitedly in dynamic traffic environments.

Machine learning based methods. Unlike traditional statistical methods, machine learning-based approaches rely on data-driven models for trajectory prediction. Classic machine learning methods include GP, SVM, Random Forests, and HMM.

Gaussian Process is an effective tool commonly used in prototype trajectory approaches. It treats trajectories as samples from a Gaussian Process and predicts possible trajectories by measuring the similarity between historical trajectories and prototype sets. When applying Gaussian Processes, parameters are determined based on the samples, and different methods vary in how prototype trajectories are constructed. However, models trained with specific trajectory samples often struggle to generalize to other scenarios.

Support Vector Machines can learn and recognize driver behaviors in complex environments. The key lies in finding support vectors satisfying classification requirements and identifying the optimal hyperplane to maximize the margin between classified data. In trajectory prediction, driving behaviors such as left turn, right turn, or going straight are usually categorized, and a kernel function is used to map the input data to a high-dimensional space for linear classification to predict trajectories. However, this method requires predefined driving behaviors, and such definitions can influence the final prediction outcomes.

Random Forests, as a nonlinear model based on an ensemble of decision trees, learn the complex features in trajectory data by constructing multiple decision trees. The core idea is to split input trajectory features (e.g., position, velocity, acceleration) into subspaces, where each decision tree independently learns these subspaces and outputs trajectory predictions through an ensemble voting mechanism. In trajectory prediction, Random Forests possess nonlinear modeling capability and exhibit high robustness to noise and outliers. Additionally, they are fast to train and can manage large-scale trajectory data effectively. However, the results are based on a collection of trees, which weakens interpretability. Furthermore, Random Forests heavily depend on the selection of input features, and they may perform less effectively in complex multi-modal interaction scenarios [13].

The HMM is a statistical model used to represent systems with hidden states that can be inferred from observable data. In the context of trajectory prediction, HMM models the behavior of traffic participants as a sequence of hidden states and observations. The model assumes that the system is a Markov process, meaning the future state depends only on the current state and not on the sequence of events that preceded it. The process is shown below in equation (3):

$$p(\mathbf{s}, \mathbf{o}) = p(s_1) \prod_{t=1}^{T-1} p(s_{t+1} \mid s_t) \prod_{t=1}^{T} p(o_t \mid s_t) \tag{3}$$

Where $p(s_1)$ represents the probability distribution of the initial state, $p(s_{t+1} \mid s_t)$ denotes the state transition probabilities that describe how states evolve over time, $p(o_t \mid s_t)$ represents the observation probabilities that describe how states generate observation data, $\mathbf{s} = (s_1, s_2, \ldots, s_T)$ refers to the hidden state sequence, and $\mathbf{o} = (o_1, o_2, \ldots, o_T)$ denotes the observation sequence.

Deep learning based methods. In recent years, the application of deep learning techniques in the field of trajectory prediction has been extensively studied. Due to their powerful feature learning capabilities, deep learning methods such as RNN, LSTM, Transformers, Gated Recurrent Units (GRUs), and GNN have demonstrated exceptional performance in capturing complex trajectory patterns. Figure 2 illustrates the general workflow of a sequential network. Historical trajectories are fed into the temporal model, which processes the data over multiple time steps. Each sequential unit automatically extracts features from the input data, and the final output generates the predicted future trajectory. These models are particularly effective in handling large-scale, complex, and highly diverse trajectory datasets, making them a robust choice for trajectory prediction tasks [14].

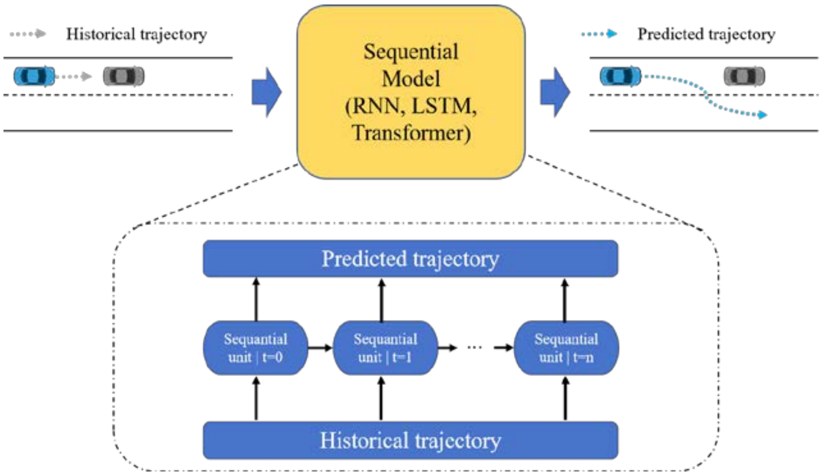


Figure 2. Process of sequential model

RNNs are specifically designed to handle sequential data by storing information from the previous time step and combining it with the current input to determine the output. However, when dealing with long sequences, RNNs are prone to issues such as vanishing or exploding gradients. Gated Recurrent Neural Networks, such as LSTM and GRU, effectively address these problems. Trajectory prediction models using RNNs can be categorized into single RNN models and multiple RNN models. Single RNNs are used for basic operations, unimodal trajectory prediction, or auxiliary models, while multiple RNNs are employed for more complex prediction tasks, such as predicting trajectories that consider the interaction with surrounding vehicles.

The attention mechanism, inspired by human cognitive processes, allows for the rapid selection of high-value information from a large dataset and has been widely applied in deep learning tasks. In trajectory prediction tasks, attention mechanisms are used to extract lane and vehicle attention to output the distribution of future trajectories or to model interactions among traffic participants. The Transformer model, which heavily relies on attention mechanisms, performs sequence-based tasks (e.g., machine translation) without using RNNs. Its advantages in handling time-series data have also been applied to trajectory prediction.

GNNs are suitable for vehicle trajectory prediction problems that rely on interaction-related factors, as shown in Figure 3; they treat each object in the environment as a node to form a graph, thereby capturing the dependencies among objects. Graph Convolutional Networks (GCNs) are the most popular GNN approach, extending convolutional operations from traditional image data to graph data. GCNs learn mapping functions to extract interaction-aware features from the characteristics of nodes and their neighbors. For example, the GRIP model treats each vehicle at each sampling time as a node, considers temporal and spatial relationships as well as interaction states, and uses a GCN model combined with an LSTM encoder-decoder to predict the trajectories of surrounding vehicles. Additionally, vector map-based GNN methods treat vehicles and vector maps as nodes, leveraging GNNs to capture interaction features between vehicles and between vehicles and maps, thereby improving trajectory prediction accuracy.

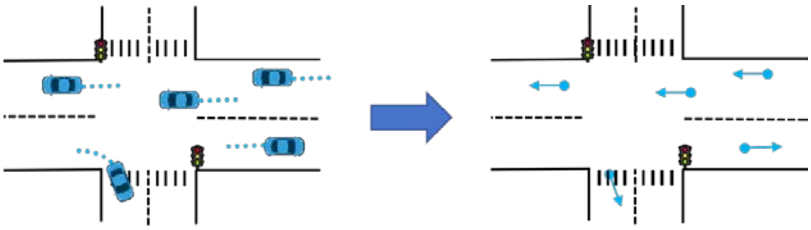


Figure 3. Illustration of graph-based representation. In this context, each agent is represented as a node in a graph, and the arrows of nodes indicate motion directions, velocities, and other relevant dynamic properties.

Hybrid methods. Hybrid methods aim to combine the strengths of various approaches to achieve higher prediction accuracy and robustness. For example, a Kalman filter can be used for initial trajectory prediction, followed by deep learning techniques to refine the results. Alternatively, combining Graph Neural Networks (GNNs) with Bayesian models can simultaneously capture complex trajectory relationships and uncertainty information. Such methods excel at integrating multimodal data and modeling complex relationships, but their algorithmic complexity and optimization challenges, coupled with high computational costs, limit their application in scenarios with strict real-time requirements.

Hybrid approaches integrate multiple types of methods to fully leverage their respective advantages, thereby improving the accuracy and robustness of trajectory prediction. For instance, physics-based models can be combined with learning-based methods, utilizing the prior knowledge of physics models and the data-driven capabilities of learning approaches to handle complex traffic scenarios more effectively.

Analysis and discussions. Trajectory prediction methods vary significantly in their assumptions, computational requirements, and the complexity of scenarios they can handle. To understand their relative strengths and limitations, it is essential to compare these approaches systematically. As illustrated in Table 2, We provide a concise analysis of the main trajectory prediction methods, focusing on their advantages, disadvantages, and potential application scenarios in dynamic environments.

Challenges and future trends. Trajectory prediction faces several key challenges that continue to hinder its performance in real-world scenarios. One significant issue is the long-tail distribution problem, where certain rare trajectory patterns appear infrequently in training data, making them difficult to predict accurately. Additionally, the real-time requirements of trajectory prediction impose strict constraints on computational efficiency, necessitating models that can deliver accurate predictions within milliseconds. Another challenge is data sparsity, where missing or incomplete data at certain time steps can compromise the model's ability to generalize effectively. Furthermore, the integration of multimodal data sources, such as visual inputs, radar signals, and high-definition maps, remains a critical challenge, as it requires designing mechanisms to effectively fuse heterogeneous data while preserving their complementary information. Lastly, adaptive dynamic modeling poses a significant challenge, as models must be capable of adjusting their parameters in real time to account for variations in traffic scenarios, environmental conditions, and the behavior of surrounding agents.

Table 2. Comparative analysis of trajectory prediction approaches.

Method	Description	Advantages	Disadvantages	Application Scenarios
Traditional Methods	Rely on physical models and kinematic assumptions (e.g., Kalman filters, ARIMA, Monte Carlo).	Computation efficient; suitable for simple traffic scenarios.	Struggle with nonlinear dynamics, complex interactions, and long-term dependencies.	Simple and static traffic scenarios; real-time prediction for low-complexity systems.
Machine Learning	Leverages labeled data for trajectory prediction (e.g., SVM, GP, Random Forest, HMM).	High accuracy in moderately complex scenarios; handles noise and stochasticity.	Sensitive to data quality; requires feature engineering; struggles with long-term dependencies.	Moderate-complexity traffic with sufficient labeled data; offline applications.
Deep Learning	Automatically extracts features and models complex interactions (e.g., RNN, CNN, Transformer, GNN).	Superior performance in high-dimensional, complex scenarios.	High computational cost; sensitive to hyperparameters; challenging deployment feasibility.	Complex and dynamic traffic scenarios; requires high computing resources.
Hybrid Methods	Combines traditional, ML, and DL methods to enhance prediction accuracy and robustness.	Leverages complementary strengths; robust in complex scenarios.	Intricate model design; increased computational demands; challenges in real-time deployment.	Highly dynamic and interactive scenarios; trade-offs between accuracy and efficiency.

Looking toward the future, several promising trends and research directions are emerging to address these challenges. Multimodal data fusion is expected to play a pivotal role, as combining diverse data sources can significantly enhance prediction accuracy and robustness in complex environments. Adaptive dynamic modeling is another critical avenue, enabling models to dynamically recalibrate parameters based on real-time data, thereby improving their adaptability to diverse traffic conditions. Reinforcement learning, with its capacity to optimize sequential decision-making, offers potential for improving the model's ability to anticipate future trajectories by learning from feedback in dynamic environments. Federated learning is gaining traction as a privacy-preserving approach to model training, allowing collaborative learning across distributed datasets while safeguarding sensitive user data. Finally, enhancing the interpretability of trajectory prediction models is becoming increasingly important, as interpretable models not only foster trust among users but also provide insights into the underlying decision-making processes, which is crucial for safety-critical applications such as autonomous driving. Together, these advancements are expected to drive the field toward more robust, efficient, and trustworthy trajectory prediction systems.

Conclusion. In this work, we provide a comprehensive analysis of existing algorithms and methodologies in trajectory prediction, systematically exploring their application scenarios, strengths, and limitations. In the future, we will focus on developing efficient, multimodal trajectory prediction frameworks that address challenges such as long-tail distributions, real-time performance, and data sparsity in complex and dynamic environments.

Reference list

[1] Parekh D, Poddar N, Rajpurkar A, et al. A review on autonomous vehicles: Progress, methods and challenges[J]. Electronics, 2022, 11(14): 2162.

[2] Rudenko A, Palmieri L, Herman M, et al. Human motion trajectory prediction: A survey[J]. The International Journal of Robotics Research, 2020, 39(8): 895-935.

[3] Coifman B, Li L. A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset[J]. Transportation Research Part B: Methodological, 2017, 105: 362-377.

[4] Krajewski R, Bock J, Kloeker L, et al. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems[C]//2018 21st international conference on intelligent transportation systems (ITSC). IEEE, 2018: 2118-2125.

[5] Chang M F, Lambert J, Sangkloy P, et al. Argoverse: 3d tracking and forecasting with rich maps[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 8748-8757.

[6] Zhan W, Sun L, Wang D, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps[J]. arXiv preprint arXiv:1910.03088, 2019.

[7] Pellegrini S, Ess A, Schindler K, et al. You'll never walk alone: Modeling social behavior for multi-target tracking[C]//2009 IEEE 12th international conference on computer vision. IEEE, 2009: 261-268.

[8] Li G, Jiao Y, Knoop V L, et al. Large car-following data based on lyft level-5 open dataset: Following autonomous vehicles vs. human-driven vehicles[C]//2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2023: 5818-5823.

[9] Huang X, Cheng X, Geng Q, et al. The apolloscape dataset for autonomous driving[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018: 954-960.

[10] Caesar H, Bankiti V, Lang A H, et al. nuscenes: A multimodal dataset for autonomous driving[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11621-11631.

[11] Shi S, Jiang L, Dai D, et al. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[12] Huang Y, Du J, Yang Z, et al. A survey on trajectory-prediction methods for autonomous driving[J]. IEEE Transactions on Intelligent Vehicles, 2022, 7(3): 652-674.

[13] Bharilya V, Kumar N. Machine learning for autonomous vehicle's trajectory prediction: A comprehensive survey, challenges, and future research directions[J]. Vehicular Communications, 2024: 100733.

[14] Pavel M I, Tan S Y, Abdullah A. Vision-based autonomous vehicle systems based on deep learning: A systematic literature review[J]. Applied Sciences, 2022, 12(14): 6831.

Author’s contribution

Yi Tang - conducted a comprehensive analysis of trajectory prediction methods, categorized existing approaches, and evaluated their applicability, advantages, and limitations across various scenarios.