IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 April 2024, accepted 25 April 2024, date of publication 29 April 2024, date of current version 16 May 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3394869*

TOPICAL REVIEW

# A Comprehensive Review on Deep Learning-Based Motion Planning and End-to-End Learning for Self-Driving Vehicle

**MANIKANDAN GANESAN[1], SIVANATHAN KANDHASAMY[2], BHARATIRAJA CHOKKALINGAM[1], (Senior Member, IEEE), AND LUCIAN MIHET-POPA[3], (Senior Member, IEEE)**

[1]Centre for Electric Mobility, Department of Electrical and Electronics Engineering, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Chennai, Tamil Nadu 603203, India
[2]Autonomous Systems Laboratory, Department of Mechatronics Engineering, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Chennai, Tamil Nadu 603203, India
[3]Department of Energy Technology, Østfold University College, 1757 Halden, Norway

Corresponding authors: Sivanathan Kandhasamy (sivanatk@srmist.edu.in), Bharatiraja Chokkalingam (bharatic@srmist.edu.in), and Lucian Mihet-Popa (lucian.mihet@hiof.no)

This work was supported in part by the Department of Science and Technology, Government of India, Promotion of University Research and Scientific Excellence (PURSE), under Award SR/PURSE/2021/65; and in part by the Autonomous System Laboratory, Department of Mechatronics Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai.

**ABSTRACT** Self-Driving Vehicles (SDVs) are increasingly popular, with companies like Google, Uber, and Tesla investing significantly in self-driving technology. These vehicles could transform commuting, offering safer, and efficient transport. A key SDV aspect is motion planning, generating secure, and efficient routes. This ensures safe navigation and prevents collisions with obstacles, pedestrians, and other vehicles. Deep Learning (DL) could aid SDV motion planning. AI tools and algorithms, like Artificial Neural Networks (ANNs), Machine Learning (ML) and DL can learn from data to create effective driving strategies, enhancing SDV adaptability to changing conditions for improved safety and efficiency. This survey gives a DL-based motion planning overview for SDVs, covering behaviour planning, trajectory planning, and End to End Learning (E2EL). It assesses various DL-based behaviour and trajectory planning methods, comparing and summarizing them. It also reviews diverse E2EL techniques including Imitation Learning (IL) and Reinforcement Learning (RL) gaining traction lately. Additionally, this review emphasizes the significance of two crucial enablers: datasets and simulation deployment frameworks for SDVs. The survey compares strategies using multiple metrics and highlights DL-based SDV implementation challenges, including simulation and real-world use cases. This article also suggests future research directions to address E2EL and DL-based motion planning limitations. The presented article is an excellent reference for scholars, engineers, and decision-makers who have an interest in DL-based SDV motion planning.

**INDEX TERMS** Behaviour planning, deep learning, end to end learning, motion planning, self-driving vehicles (SDV), trajectory planning.

## I. INTRODUCTION

In over the last thirty years, there has been a significant increase in worldwide research on Self-Driving Vehicles (SDVs). Recent sensor and processing technology advances, the potential to alter vehicular mobility, and the expected societal benefit have encouraged these advancements. Road accidents killed almost 1.3 million people annually, according to the WHO. Many of the 20–50 million non-fatal injuries cause impairments [1]. Reasons include human mistake, uneven speed, drunk driving, and distracted driving. SDVs greatly reduce driver mistake and irresponsibility

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Chen.

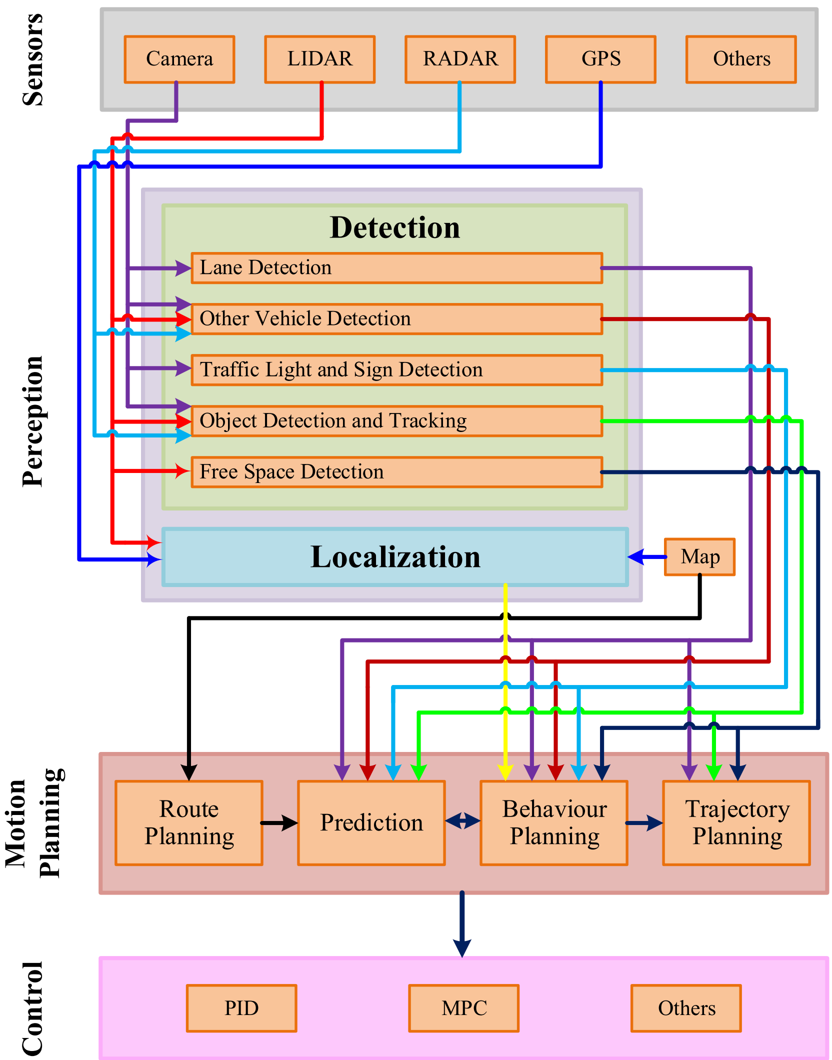


**FIGURE 1. Functional blocks of SDV.**

in vehicle collisions. Physically or visually handicapped people who cannot drive will also have personal mobility. SDVs could help minimize driving stress by optimizing transportation time. Further, Interest in SDVs has grown rapidly in government, industry, academia, and the public. Due to advances in Artificial Intelligence (AI) and computing hardware [2], SDVs can modernize transportation. In particular, broad SDV adoption offers great potential to reduce traffic accidents and congestion, especially in densely populated urban areas [3]. Reliability and safety difficulties limit SDVs to experimental programmers, notwithstanding experts' advances. Installing various sensors on small and medium-sized vehicles improves performance, safety, efficiency, and situational awareness. However, even with many sensors, SDVs struggle to recognize and respond to complex circumstances. To successfully implement self-driving technology, planning approaches must be safe, resilient, and adaptable [4].

### *A. BACKGROUND*

The well-established planner uses the modular technique, often known as rule-based planning. SDV requires this method and other methods including perception [5], localization, and control [6], as shown in figure 1. This suggested strategy is streamlined and modified from papers [7] and [8]. This method is crucial to modular approach. Interpretability makes the modular approach framework beneficial for finding faulty modules if the system acts unexpectedly or has defects. Section II discusses just modular approach planning. The modular approach to planning involves two main components: global route planning, which establishes a road-level path from the initial point to the desired destination, and local behavioral and trajectory planning, which develops a short-range trajectory. Despite its widespread use in the business, modular approach planning demands a lot of processing resources and manual heuristic functions. This research focuses on deep learning-based behaviour and

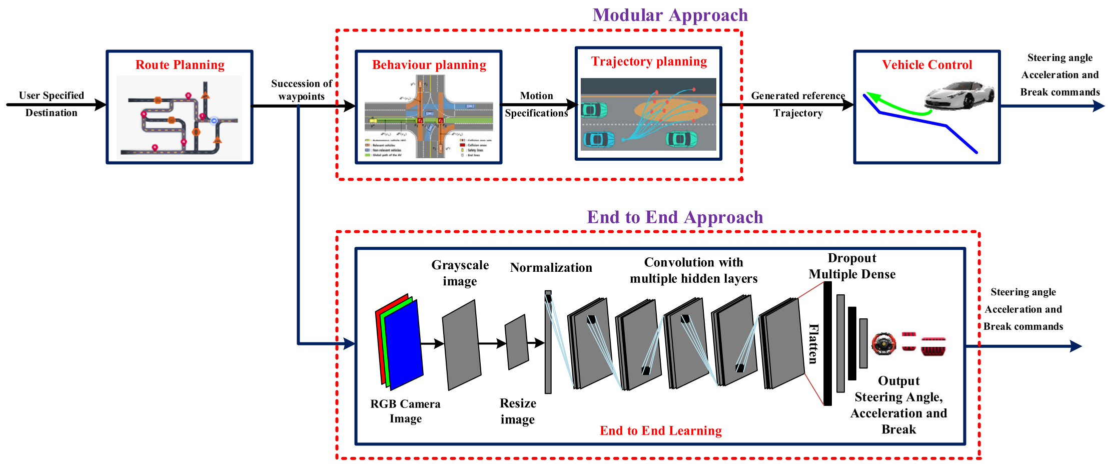


**FIGURE 2.** Modular and End to End Approach for motion planning.

trajectory planning algorithms inside the modular approach planning method and E2EL methods as shown in figure 2. The numerous DL and E2EL motion planning algorithms utilized in SDVs have been thoroughly reviewed

The implementation of motion planning is of utmost importance for SDVs to function within a secure setting. It permits the ego-vehicle to move from its starting point to the end point., taking into account factors such as road boundaries, dynamics of vehicle, road obstacles, and traffic regulations [9]. In the disciplines of SDV, motion planning algorithms including Graph search-based planners [10], Sampling Based Planners, Interpolating Curve Planners [11] and Numerical Optimization [12] have had considerable success.

**TABLE 1.** Key differences between traditional motion planning and DL based motion planning.

| Features | Traditional Motion Planning | DL based Motion Planning |
| --- | --- | --- |
| Adaptability to Change | Lower | Higher |
| Real-time Performance | Lower | Higher |
| Generalization | Lower | Higher |
| Computational Cost | Lower (often) | Higher (training) |
| Explainability | Higher | Lower (it will be a black box) |
| Planning Granularity | Limited (predefined steps) | Flexible (continuous adaptation) |
| Obstacle Handling | Explicitly programmed | Can learn from data |
| Integration with Sensors | Limited | Can directly incorporate sensor data |
| Success Rate in Cluttered Environments | Lower | Higher |

However, Traditional SDVs use pre-programmed regulations, which is like building a vehicle with a rulebook for every driving circumstance. This technique works well for simple conditions, but it struggles with unpredictable drivers, complex environments like construction zones, and real-time decision-making. Learning is more dynamic with DL. Because they learn from massive amounts of driving data, these models can "understand" complex traffic patterns and predict driver behaviour better than pre-programmed restrictions. They can also adjust their plans in real time based on sensor data to avoid unexpected obstacles and driver behaviour. DL empowers SDVs with human-like adaptability, making them safer and more capable in real-world scenarios. Table 1 further clarifies the key differences between traditional motion planning and DL based motion planning.

### *B. RELATED STUDIES*

In the expansive realm of academic literature, a multitude of studies have been dedicated to exploring the diverse dimensions of SDV technology. Among these investigations, A significant portion has concentrated on using DL algorithms for SDV motion planning [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]. However, while existing surveys have provided comprehensive overviews of this area, they often fall short in their treatment of behavior planning and trajectory planning as distinct entities. Comparison of existing survey shown in table 2. This lack of separate in-depth analysis undermines our ability to grasp the nuances and challenges unique to each component.

Moreover, a striking gap in the literature lies in the absence of thorough implementation analysis and metrics evaluation. While theoretical frameworks and algorithmic advancements are crucial, their practical implementation and performance metrics are equally vital for the successful deployment of SDV systems. Without a detailed examination of implementation strategies and rigorous evaluation of performance metrics, our understanding remains incomplete, hindering progress in the field.

It is essential to address these shortcomings in order to advance SDV technology. By conducting separate,

**TABLE 2.** Comparison of Existing survey related to DL based motion planning techniques for SDVs.

| Paper and year | Review Article Titles | Modular approach | | End to End Approach | | | Focus on Dataset and platform | Focus on Metrics analysis | Focus on Implem -entation analysis |
|---|---|---|---|---|---|---|---|---|---|
| | | Behaviour planning | Trajectory planning | DL Methods | IL methods | RIL methods | | | |
| Zhou H, 2019 [13] | Review of Learning-Based Longitudinal Motion Planning for Autonomous Vehicles: Research Gaps Between Self-Driving and Traffic Congestion | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Claussmann L, 2019 [14] | A review of motion planning for highway autonomous driving | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| Muhammad K, 2020 [15] | Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ |
| Aradi S,2022 [16] | Survey of deep reinforcement learning for motion planning of autonomous vehicles | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Grigorescu S, 2020 [17] | A survey of deep learning techniques for autonomous driving | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Ye F, 2021[18] | A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Elallid BB, 2022 [19] | A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ |
| S. Teng, 2023[20] | Motion planning for autonomous driving: The state of the art and future perspectives | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| P. S. Chib, 2024[21] | Recent Advancements in End-to-End Autonomous Driving Using Deep Learning: A Survey | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| Wang N, 2024[22] | A survey on path planning for autonomous ground vehicles in unstructured environments | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| A. Khanum, 2023[175] | Involvement of Deep Learning for Vision Sensor-Based Autonomous Driving Control: A Review | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| **Ours** | A Comprehensive Survey on Deep Learning-Based Motion Planning and End-To-End Learning for Self-Driving Vehicle | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

comprehensive surveys on behavior planning, trajectory planning, implementation analysis, and metrics evaluation, researchers can offer nuanced insights into each aspect of SDV systems. Such holistic examinations will not only deepen our understanding but also pave the way for more effective and reliable SDV capable of navigating real-world scenarios with confidence and precision. Hence, this survey mainly focuses on the DL based Motion planning (Behaviour planning and Trajectory planning) and End-to-End Learning (E2EL) methods (IL and RL).

### *C. ARTICLE STRUCTURE AND CONTRIBUTION*

In this comprehensive survey, we delve into the intricacies of motion planning and E2EL techniques tailored for SDVs. The survey is meticulously structured into several sections to provide a thorough examination of the topic. In Section I, we offer a succinct introduction to the pertinent literature, laying the groundwork for an in-depth exploration of motion planning and E2EL for SDVs. Section II.A is dedicated to elucidating the diverse DL methodologies employed for behavioral planning within the context of SDVs. This section meticulously dissects various DL techniques utilized to enhance the decision-making capabilities of SDVs.

Furthermore, Section II-B offers a detailed analysis of the DL methodologies employed specifically for trajectory planning, a crucial aspect of SDV operation. Our discussion of E2EL for SDVs begins in Section III. This section delves into the intricacies of leveraging E2EL methodologies (IL and RL) to give SDVs the capacity to efficiently perform driving duties by enabling them to learn straight through data gathered from sensors. In the IV section of our review paper, we addressed SDVs practical enablers such as datasets and simulation deployment frameworks. In particular, we examined datasets for model training and validation and simulation deployment platforms due to their ability to simulate real-world settings. We found that high-quality datasets and strong simulation deployment frameworks

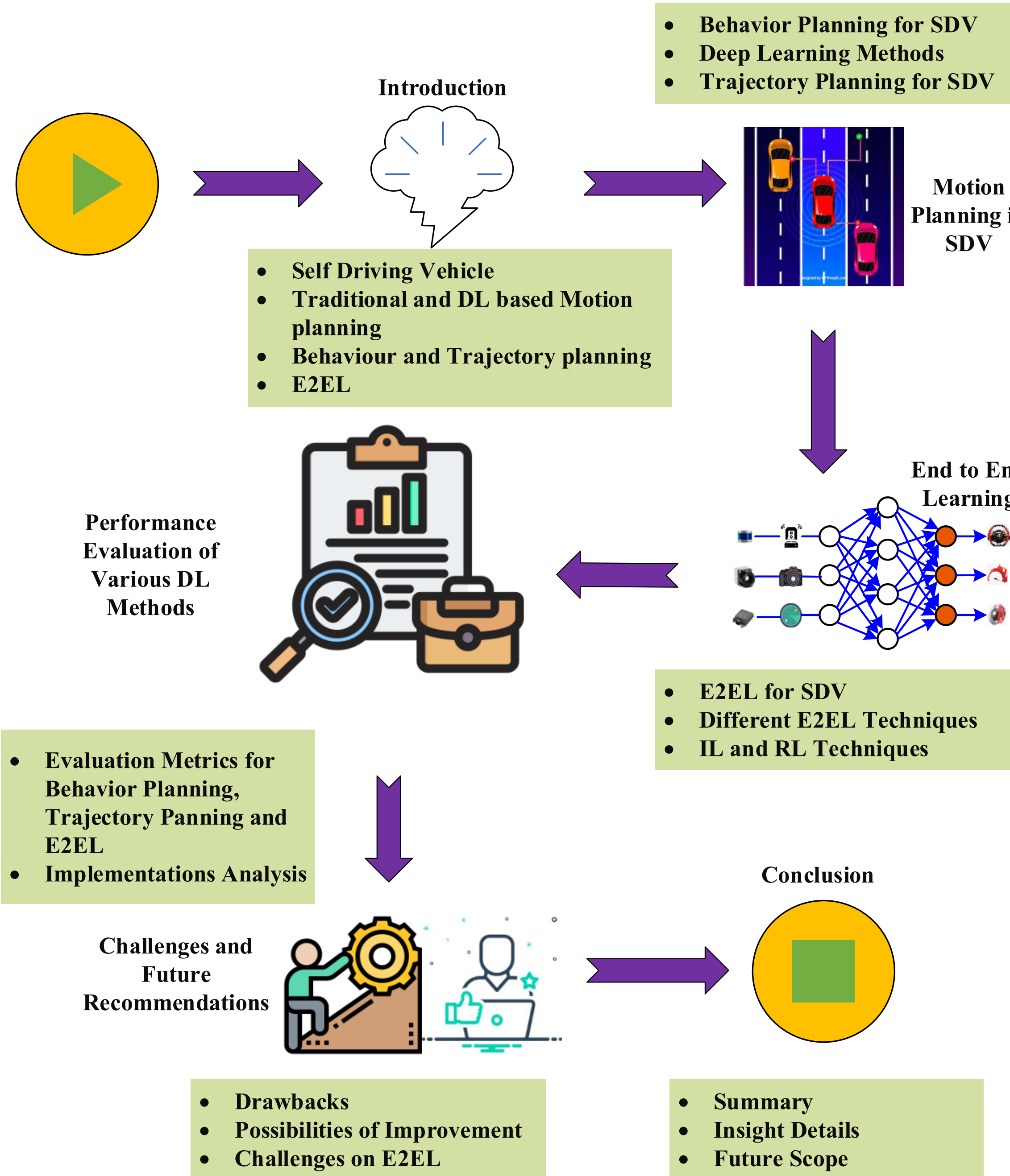


**FIGURE 3.** Survey organization.

improve SDVs dependability and efficacy. Performance evaluation and comparative analysis of the reviewed models are meticulously discussed in Section V. This section offers insights into the effectiveness and efficiency of different approaches, aiding in identifying promising avenues for further research and development. Moreover, In Section VI, we examined different types of implementations of motion planning algorithms for Self-Driving Vehicles (SDVs) and compared these various implementation types across the reviewed articles. Additionally, we provided explanations for why the major focus of research is primarily on simulation rather than real-world demonstration. Further, Section VII elucidates current challenges encountered in the domain of SDV motion planning and E2EL, while also providing future recommendations to overcome these hurdles and advance the field. Finally, Section VI serves as the conclusive segment of this survey, encapsulating key findings, summarizing crucial insights, and offering concluding remarks on the surveyed topics. The organization chart provided in Figure 3 visually represents the structured flow of the survey, facilitating a clear understanding of its contents and relationships between different sections.

The following is the significant contribution of this paper:

- A concise and comprehensive introduction, along with an existing literature review in the area of motion planning for SDVs, is presented.
- The article also provides a thorough literature review on DL-based SDV behaviour planning and trajectory planning, along with an extensive survey of E2EL for SDVs.
- This article also provides a deep understanding of various datasets and simulation deployment frameworks utilized for SDV development.
- The discussion includes implementation percentages among the reviewed papers for DL-based motion planning (pointing out the behavior and trajectory planning) and E2EL.
- The comparison and analysis of percentages related to various implementation types found within the reviewed papers are meticulously presented.
- Various algorithms in DL-based motion planning (including behaviour planning and trajectory planning) and E2EL are compared and discussed in terms of performance metrics.
- Different challenges and their corresponding recommendations for DL-based motion planning (including behaviour and trajectory planning) and E2EL for SDVs are discussed.

## II. OVERVIEW OF MOTION PLANNING PIPELINE IN SDV

The modular approach pipeline for SDV motion planning is provided here. Behaviour and trajectory planning make up the sophisticated local motion planning modular approach pipeline. The vehicle makes high-level judgements based on its environment and purpose during behaviour planning. This component determines the vehicle's dynamic road behaviour using numerous decision-making techniques. Overtaking, lane changes, pedestrian yielding, and intersection manoeuvres are examples. This site collects sensor data, predicts vehicle and object movements, and creates driving tactics using complex algorithms. Trajectory planning refines high-level judgements into a vehicle trajectory after behaviour planning. Considering vehicle dynamics, road constraints, and safety, this component generates a smooth and practical SDV path. Trajectory planning algorithms optimize routes for safety and efficiency by considering vehicle speed, acceleration, road curvature, and impediments. Probabilistic sampling, optimization, and spline fitting generate safe, comfortable passenger pathways. Splitting the motion planning pipeline into behaviour and trajectory planning lets SDV systems handle complex scenarios safely and efficiently. Modular design lets complicated algorithms and models for each task be integrated, providing resilient and adaptive SDV systems.

### A. DETAILED SURVEY ON DL-BASED BEHAVIOUR PLANNING FOR SDV

This chapter embarks on a comprehensive exploration of DL-based behavior planning techniques tailored specifically for SDVs. It delves into the intricacies of how DL algorithms are utilized to interpret sensor data, understand the surrounding environment, and formulate high-level decisions for SDV behavior. By leveraging DL's capabilities in processing vast amounts of data and learning complex patterns, behavior planning systems can adapt to diverse driving scenarios and exhibit human-like decision-making capabilities.

By providing an in-depth survey of DL-based behavior planning for SDVs, this chapter seeks to clarify the state-of-the-art, highlight new developments, and suggest directions for further study and advancement In this crucial field of SDV technology. Shi et al. [23] proposed a Deep Neural Network (DNN) algorithm which is used to predict the combined car following and lane-changing behaviour of SDV. Authors Developed a Switch Neural Network (SNN) by utilizing Temporal Convolution Neural Network (TCN), Bi-directional Long-Short Term Memory (BiLSTM) and Attention Mechanism. A mathematical model for predicting vehicle trajectory in two dimensions was created. X, Y-denoted Cartesian Coordinate, which indicates the target vehicle's position, is used to describe a vehicle's trajectory. Meanwhile, A FIS_LSTM model was created by Wang et al. [24] to track the behaviour of SDVs when changing lanes. The FIS_LSTM model integrates LSTM network with Fuzzy Inference System (FIS). The primary determinants of lane-changing are drivers and surrounding traffic conditions. FIS is used to integrate driver behaviour and surrounding traffic conditions. Based on drivers' cognitive abilities, fuzzy rules are developed.

Additionally, Gonzalez-Miranda and Ibarra-Zannatha [25] implemented a behaviour selector for SDV using Feed Forward Neural Network (FFN) with Autonomy hardware. The purpose of this model is used to select the proper behaviour (such as Lane keeping, passing parked cars, Emergency breaks, and parking based on passenger request) based on the surrounding traffic environment. For the environment perception, CNN based yolov3 DL algorithm was implemented. Further, using an Attention Enhanced Residual-MBi-LSTM neural network, Wu et al. [26] developed a prediction model for lane change intentions. The purpose of this model is to predict the driver's anticipated behaviour when changing lanes. The initiative makes use of the HighD dataset to extract the ego-vehicle and surrounding vehicle's trajectories.

Furthermore, to forecast a driver's intention to change lanes, Tang et al. [27] employed the Multi_LSTM technique. HIL (Hardware in Loop) simulation and NGSIM (Next Generation Simulation) dataset were utilized, The SDV driving conditions and the effects of the surrounding cars are taken into consideration when building the test set and training set. The suggested model was designed

**TABLE 3.** Summary of commonly used DL methods for behaviour planning in SDVs.

| Deep Learning Technique | Performance | Advantage | Disadvantage | Limitation |
|---|---|---|---|---|
| Feedforward Neural Network | Moderate | • Simpler architecture, easier to train.<br>• Can be used for tasks like traffic sign recognition or initial behavior selection | • Limited ability to handle complex relationships or sequential data<br>• Its suitable for long-term planning or dynamic decision-making | • Requires large datasets for complex problems |
| Convolutional Neural Network (CNN) | High (Image Recognition) | • Excellent at processing sensor data (cameras, LiDAR)<br>• Identifies objects, lanes, and traffic signals | • Requires large, labeled datasets for training<br>• Computationally expensive for real-time processing | • Limited to interpreting visual data, may not capture motion or intent |
| Recurrent Neural Network (RNN) | Good (Behaviour Prediction) | • Handles sequential data like sensor readings over time<br>• Predicts vehicle behaviour and maneuvers | • It struggles with long-term dependencies in complex scenarios<br>• Prone to vanishing/exploding gradient problems | • Limited ability to adapt to unforeseen events or unexpected situations |
| Long Short-Term Memory (LSTM) | High (Long-Term Planning) | • Improved RNN architecture for complex sequences<br>• Captures long-term dependencies in traffic patterns | • More complex to train than RNNs<br>• Requires significant computational resources | • Similar limitations to RNNs in handling highly dynamic or unpredictable situations |
| Gated Recurrent Unit (GRU) | Good (Behaviour Prediction) | • Similar to LSTMs, but simpler architecture<br>• Handles sequential data and predicts vehicle behavior | • It struggles with very long-term dependencies compared to LSTMs<br>• Less computationally expensive than LSTMs | • Similar limitations to RNNs in handling highly dynamic or unpredictable situations |
| Generative Adversarial Network (GAN) | High (Generative tasks) | • Powerful for generating realistic data<br>• It learns complex distributions | • Training is unstable<br>• Requires careful hyperparameter tuning | • Limited interpretability of generated data |

to discover the characteristics of vehicle behaviour and the relationship between time series of different states during a lane change. Authors strongly proposed that once prediction time increases rule-based model accuracy decreases but the proposed model accuracy increases. Besides this, Wang et.al [28] developed an autonomous lane-changing system using LSTM-based DL technology. The system aims to mimic human decision-making by using a combination of visual and sensory data, as well as an ANN that learns from real-world driving scenarios. The author notes that traditional rule-based systems used in current SDVs cannot handle complex real-world situations like merging onto highways. The proposed system can be improved on these limitations by learning from experience and using a probabilistic approach to decision-making. The article concludes that the system shows promises for enhancing the safety and efficiency of SDV technologies.

Meanwhile, Lin Li et al. [29] deployed Recurrent Neural Networks (RNNs) to forecast lane changes in vehicles. For SDV, the technology can help the SDV perform better lane changes and minimize collisions. The RNN uses sensor data including vehicle relative location, speed, and acceleration to predict lane changes in a short time. The system also filters false alerts and ensures inference correctness using rules. Additionally, A prediction making decisions method for lane changes was presented by Yonghwan Jeong [30] to increase the effectiveness of SDVs on highways. The suggested model analyses and forecasts lane shifts using Bidirectional-LSTM, resulting in safer and faster self-driving on highways. The Bi-LSTM network is an RNN capable of learning prolonged dependencies in consecutive data. By utilizing Bi-LSTM, the algorithm is capable of making forecasting in real time because it is able to learn as well as adjust to adjustments in driving conditions.

Furthermore, for on-road SDV, Xiao Wang et al. [31] created a technique called LSTM_CRF, which enables the decision-making process of an SDV to become more human-like. The model enhances the precision and effectiveness of self-driving systems by combining the advantages of the LSTM network and Conditional Random Field (CRF). The LSTM_CRF technique aims to predict the next maneuver of an SDV by analyzing various sensory inputs such as camera and lidar data, as well as the current driving state such as speed and acceleration. The suggested method could increase the safety and reliability of SDVs by enabling them to make human-like decisions in complex and dynamic driving situations.

In addition, a Deep Bidirectional RNN Network (DBRNN) was deployed by Oluwatobi Olabiyi et. al [32] that takes input as the past states of the SDV, such as position, steering angle, and speed, and anticipates the SDV future behaviours, such

**TABLE 4.** Technical comparison of different deep learning algorithms in behaviour planning for SDVs.

| Paper and year | Proposed DL algorithm | Type of Behaviour | Dataset | Dataset Ratio | Hardware | Software | Response Time or Time Before Action | Output | Implementation |
|---|---|---|---|---|---|---|---|---|---|
| [23]Shi et. at 2022 | Switch neural network (Bi-LSTM, TCN, AM) | Car following Lane changing | NGSIM | 2250/-/250 trajectories | CPU-Ryzen 3700X, GPU-RTX 3090 | Python with Pytorch | RS-21.7ms | Future Lane Changing Decision (FLCD) and Future trajectory | Simulation +NA |
| [24]Weida Wang et.al 2022 | FIS_LSTM | Lane changing | NGSIM and HIL data | 75%/-/25% data | CPU- intel i7-10 700@2.90GHz RAM-16GB | FIS and Prescan | TBA- 3s | FLCD and Future trajectory | HIL |
| [25]Miranda et. al 2022 | FFN, CNN with yolov3 | Lane-keeping, car passing, parallel park, emergency stop | VOC 2012 and Real data | 18016/-/- images | AUTOMINY Vehicle | Git hub's software darknet ROS | Not reported | Best behaviour selection based on surrounding environment status | Simulation +RWI |
| [26] Zhanqian Wu et. al 2022 | Attention enhanced Residual MBi-LSTM | Lane changing | HighD | 69,861 total trajectories 75%/15%/15% | HIL platform | MATLAB Simulink, Carsim | TBA-2.07s | FLCD | HIL |
| [27]Liang Tang et. al 2020 | Multi LSTM | Lane changing | NGSIM | 3330 total samples 70%/-/30% | HIL platform | Not reported | Not reported | FLCD | HIL |
| [28]Tao Wang et. al 2021 | modified LSTM | Lane changing | VTD data | 10800 total sets 8:2 ratio training and validation | CPU-i7-7700 @ 3.60GHz RAM-32 GB | Virtual Test Drive, Simulink, PreScan/ | RT-5.75s TBA-1.75s | Left FLCD based on the surrounding environment | HIL |
| [29] Lin Li et. al 2021 | Intention Inference Based on RNN (LSTM_GRU) | Lane changing | NGSIM | Total 300 trajectories 70%/-/30% trajectories | Not reported | TensorFlow, Carsim with Simulink | Not reported | FLCD based on congestion in different lanes | Simulation +NA |
| [30] Jeong 2021 | Bi-LSTM based RNN | Lane changing | NGSIM and Argoverse | total 20108 data set 1120/320/160 data sets | Data collection vehicle. | MATLAB | TBA- 2.7s | FLCD based on the surrounding environment | Simulation +RWI |
| [31] Xiao Wang et. al 2018 | LSTM with CRF | Lane changing and lane keeping | NGSIM | Entire data set | CPU- intel i7 8550u RAM-8GB | Keras | RT-14ms | FLCD based on the surrounding environment | Simulation +NA |
| [32] Olabiyi et. al 2017 | DBRNN (LSTM_GRU) | Lane changing, braking action, turns action | 35 hours of real driving data | 70%/15%/15% image data | GPU-Nvidia GTX | Tensorflow | RT-5s | Lane Changing (LC), Braking and turns action prediction SDV | Simulation +NA |
| [33] Yingshi Guo et. al 2021 | AT-BiLSTM | Lane changing and lane keeping | Real data collected by 42 Km driving | 1220 total samples 60%/20%/20% samples | driving simulator platform | Not reported | TBA-3s | LC and lane keeping prediction of ego vehicle | HIL |
| [34] Hao Zixu et. al 2020 | attention based GRU | Lane changing | NGSIM | Not reported | Not reported | Not reported | TBA-3s | LC prediction of the ego vehicle | Simulation +NA |
| [35] Zhensong Wei et. al 2019 | Deep Residual Neural Network (DRNN) | Lane changing | NU Drive 1000 with IMU data | 187440/4500/ 24626 images, IMU- 6 entry vector | Nvidia GeForce GTX, RAM-64GB, 4 core 4.20 GHz | Tensorflow | RT- 0.028s | LC prediction of the ego vehicle | Simulation +NA |

as turning left or right, accelerating or braking. The DBRNN design enables the model to capture the context of the driver's conduct in both the past and the future and makes it suitable for predicting driver reactions in complex driving scenarios.

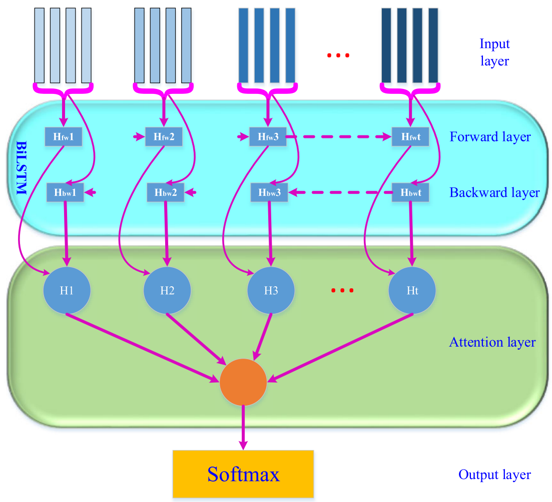


**FIGURE 4.** AT-BiLSTM architecture proposed by [33].

Meanwhile, Yingshi Guo et al. suggested a networked technique for identifying drivers' lane-changing intentions [33]. The proposed method predicts driving behaviour using camera and sensor data. (AT-BiLSTM) is an attention mechanism based BiLSTM network shown in figure 4. This work influences intelligent transportation technologies that improve driver safety and traffic efficiency. Further, An AT based Gated Recurrent Unit (GRU) model by HAO Zixu et al. [34] recognized driver intention and predicted vehicle trajectory. The model predicts vehicle direction and behaviour using steering angle, acceleration, speed, driver gaze behaviour, and road context data from the car's sensors. The proposed paradigm is ideal for complex route geometry and unexpected events. Furthermore, using a Deep Residual Neural Network (DRNN), Zhensong Wei et al. [35] detect lane-changing behaviour using vision. The authors design and test the method using the Naturalistic Driving Study (NDS), a publicly available dataset of real-world driving footage from diverse viewpoints. This work aims to develop a reliable and scalable system to detect lane-changing behaviour in fully autonomous and ADAS cars. The method uses a Region-based Convolutional Neural Networks (RCNN) model, which has performed well in image and video object detection. The author trains the RCNN model to detect stable, changing right, shifting left, and ambiguous behaviour using the NDS dataset. Summary of the commonly used DL algorithms are presented in table 3.

However, all the aforementioned DL-based behaviour planning algorithms were trained and evaluated in their corresponding dataset and experimented with their respective implementation method, the outcomes demonstrated that the aforementioned algorithms were better performed when compared with their corresponding state of art algorithms. Table 4 compares technical details of recently available various DL algorithms in behaviour planning for SDV based on the following parameters such as type of DL algorithm utilized, type of input, type of output, Type of Dataset, Software and Hardware Utilized, Response Time (RT)/ Time Before Action (TBA) and way of Implementation, which are more important for the researcher to get deep knowledge in this field of behaviour planning for SDVs.

### B. DETAILED SURVEY ON DL-BASED TRAJECTORY PLANNING FOR SDV

SDVs are becoming increasingly popular, and one of the fundamental challenges in their development is trajectory planning. The process of trajectory planning entails figuring out the best route for SDV to travel to get to its destination quickly and safely. It is a complex task that considers several elements, including the state of the roads, traffic, environmental variables, etc. DL approaches have demonstrated promising achievements in overcoming this issue in recent years. DL techniques are applicable to a variety of domains, notably SDVs, and are capable of learning patterns as well as structures from massive volumes of data [36]. They can aid in enhancing the reliability and accuracy of trajectory planning for SDV, making them safer and more reliable in real-world scenarios. An overview of trajectory planning for SDV using DL has been discussed in this section. It will cover the different DL techniques used in trajectory planning for SDV using the DL framework.

Stefano Pini et. al [37] described a new approach to SDV that aims to increase safety by enabling vehicles to absorb knowledge from the past and adjust to a variety of driving situations. The strategy is predicated on the idea of combining experts, which involves combining multiple models or algorithms to achieve better performance. To implement this approach, the author suggests training different models to focus on specific aspects of driving, such as predicting the behaviour of other vehicles, identifying obstacles, or planning safe and efficient routes. These models can then be combined into a single system that can dynamically adapt to changing road conditions and make more accurate predictions. The article presents a compelling vision of how a mixture of expert approaches could potentially increase the dependability and safety of SDT in real-world conditions.

Additionally, Dan Wang et. al [38] have deployed a new approach to SDVs that relies on DNNs for trajectory learning. To extract pertinent characteristics from the input data and simulate temporal dependencies, the authors suggest a hybrid architecture that blend CNN and RNN. In simulator tests, the strategy produced encouraging results by correctly anticipating the SDV prospective trajectory and preventing collisions with surrounding vehicles and obstacles. The system's adaptability to different driving scenarios is one of its strengths. However, good performance necessitates a lot of training data. Further testing and refinement are necessary before the approach can be deployed in real-world situations.

Besides this, Ting Chen et. al [39] have developed an innovative approach for predicting the trajectory of human

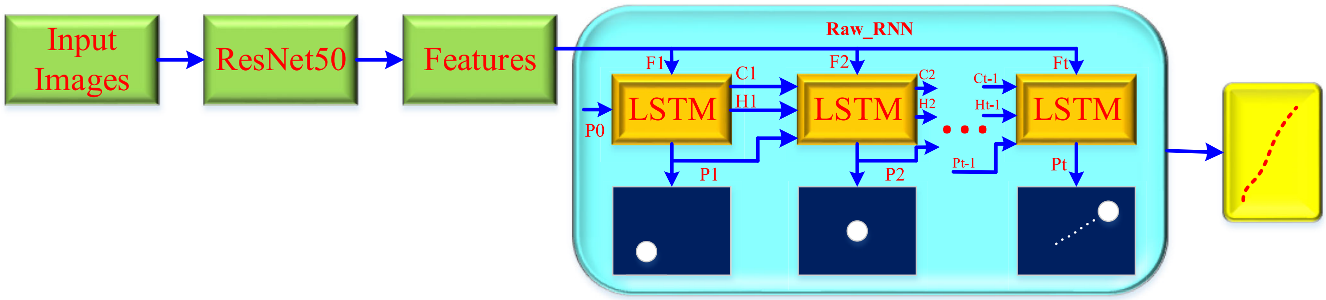


**FIGURE 5.** CNN _ Raw-RNN architecture proposed by [38].

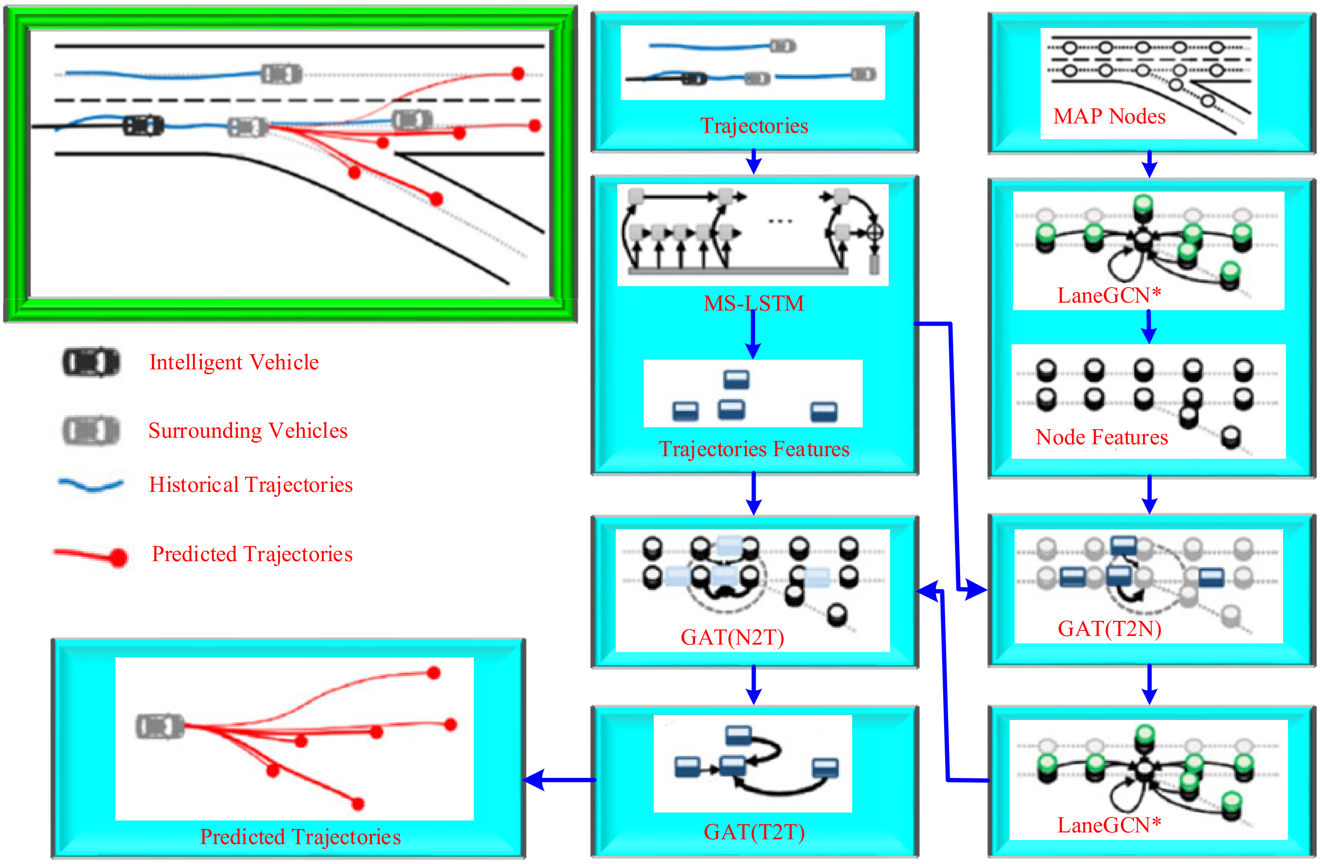


**FIGURE 6.** LaneGCN architecture proposed by [45].

motion in complex environments. The proposed method uses both visual and auditory cues to predict movements. The authors demonstrate the proposed method's effectiveness in predicting complex human motion trajectories in dynamic and cluttered environments. In addition, to detect traffic conflicts at unsignalized crossings, Qianxia Cao et. al [40] have developed a real-time vehicle trajectory estimation technique. To forecast each vehicle's upcoming motion as it approaches an intersection, the system employs a DL-based method. The CNN and LSTM networks are the foundation of the strategy, which can process the vehicle trajectory data's spatial and temporal characteristics. The system's ability to detect traffic conflicts in real-time has the potential to improve traffic safety at unsignalized intersections, where conflicts are more likely to occur.

Similarly, a DL-based technique for vehicle trajectory forecasting in top-view picture sequences has been proposed by Zahra Salahshoori Nejad et. al [41]. To extract characteristics from the input image sequences and to forecast the prospective motion of the vehicle, the system employs a CNN and LSTM network respectively. The method is created to consider the spatial and temporal characteristics of the SDV movements, Along with the roadway profile and several environmental elements that could influence the motion of the vehicle. Additionally, a multi-modal vehicle trajectory forecasting system was created by Wei Tian et al. [42] using cooperative learning of lane direction, vehicle communication, and intended action. Based on a vehicle's orientation and position concerning the lanes on the road, the lane direction model forecasts the possibility that it will be in a particular lane. The suggested approach trains the three algorithms simultaneously using a multiple-tasking learning framework, enabling them to exchange knowledge and enhance one another's performance.

Meanwhile, for scenarios involving highway driving, Ruben Izquierdo et al. [43] have suggested a system for predicting vehicle trajectory that makes use of a Bird's Eye View (BEV) depiction. Using a top-down, high-angle image of the road, the method predicts the subsequent trajectory of the SDV on the highway using a BEV rendering of the driving scenario. Based on the SDV state, velocity, and acceleration as well as the spatial information that the CNN has retrieved, the LSTM network forecasts its future course. Besides this, a trajectory prediction model with a corrective mechanism has been brought forward by Pin Lv et. al [44] for connected and SDVs. To learn the temporal changes in traffic conditions and forecast the prospective trajectories of surrounding vehicles, the proposed model makes use of an LSTM neural network. To account for potential errors in the predictions, the model also has a correction mechanism that modifies the projected trajectories based on the CAV's present condition and the predicted trajectories. The authors claim the proposed method is particularly suitable in scenarios where the traffic is complex, and the CAV is surrounded by multiple nearby vehicles. The suggested model, which includes a corrective mechanism, could be used to improve the trajectory prediction skills of real-world SDVs.

Further, Bing Zhou et. al [45] have innovated an improved version of the LaneGCN trajectory forecasting technique for SDV shown in figure 6. The original LaneGCN technique models the spatial relationships between several lanes and predicts the paths of other actors in the environment using a Graph Convolutional neural Network (GCN). The suggested improvement to the technique includes the use of a Dynamic Graph Convolutional Network (DGCN) that can adaptively adjust the weights of the graph convolution according to the current traffic condition. The suggested enhancement to the LaneGCN algorithm can increase real-world SDV performance and contribute to the further development of SDV.Additionally, a DL-based methodology for trajectory prediction using regionally clustered data has been developed by Aditya Shrivastava et. al [46]. To estimate a moving object's trajectory based on its prior locations and timestamps, the suggested method used LSTM neural networks. A series of location-time pairs of a moving object are used as the network's input and are clustered according to their proximity to one another. The trajectory estimation problem is simplified using clustering, which also captures the basic temporal and spatial trends in the data. The research emphasizes the importance of clustering techniques in decreasing the complexity of trajectory estimation.

Furthermore, A spatiotemporal LSTM network has been proposed by Zhengwei Bai et al. [47] as part of a DL-based method for motion planning in SDVs. The suggested method utilizes a sequence of input pictures from a front-facing camera to forecast the future mobility of the vehicle while taking into consideration the spatiotemporal properties of the traffic environment. There are two sub-networks in the spatial temporal LSTM network, one is used for the processing of spatial data and the other is used for the processing of temporal data.

Meanwhile, an innovative DL-based method for motion planning in SDV has been developed by Sheng Song et al. [48]. In the suggested method, camera pictures and motion commands are input sensor data, and a CNN is used to learn an attribute description of those data. Then, to forecast the future trajectory of the SDV, an attribute description is sent to an LSTM network. To produce precise trajectory predictions, the LSTM network considers both the spatial and temporal relations in the sensor input. The research also emphasizes the dataset's usefulness in creating and assessing DL-based motion planning models for SDV. Further, to forecast multimodal trajectory in self-driving scenarios, Henggang Cui et. al [49] have presented a DL-based technique. The suggested method employs deep convolutional networks to multimodally model the spatiotemporal interdependence of vehicle trajectories. The authors provide a brand-new dataset made up of more than 300 hours' worth of in-depth sensor data gathered from various cities using a fleet of SDVs equipped with cameras, RADAR, and LIDAR sensors. The effectiveness of the suggested strategy is proved on a sizable and varied real-world dataset, highlighting its potential for usage in SDV.

In addition, to learn local state motions for SDV, Sorin Mihai Grigorescu et. al [50] suggest a neuroevolutionary technique. The suggested method employs a neuroevolutionary algorithm to develop neural networks that can forecast the upcoming motion of the vehicle by its present condition and sensor inputs. The technique employs a function of fitness to evaluate the trajectory's correctness and smoothness. The neuroevolutionary algorithm uses a population of neural networks that are randomly initialized and evaluated in a simulated driving environment. The fitness function is used to determine which individual's trajectory is the most fit for the following generation by evaluating the smoothness and accuracy of the anticipated trajectory.

Similarly, Yonghwan Jeong et. al [51] suggest employing an LSTM-RNN to forecast the motions of nearby vehicles as a solution for motion planning in SDV at multi-lane turn road crossings. In an uncertain environment, the suggested method anticipates the velocity of the nearby vehicles while

**TABLE 5.** Technical comparison of different deep learning algorithms in trajectory planning for SDVs.

| Paper and year | proposed Model | DL Algorithms | Dataset | Dataset Ratio | Hardware | Software Environments | Response Time | Output | Implementa -tion |
|---|---|---|---|---|---|---|---|---|---|
| [37] Pini et. al (2022) | SafePath Net | RCNN & FFN | Real data collected from SDV environment | 270 hours/60 hours/- | Not reported | Not reported | Not reported | Future Trajectory (FT) of SDV | Simulation+ NA+RWI |
| [38] Dan Wang et. al (2021) | CNN_Raw-RNN network | CNN & LSTM | Real data - GAC dataset 50 hours of data collected from different conditions | 4,80,000/60 000/ 60000 Images | Not reported | Not reported | Not reported | FT of SDV for 30 m | Simulation+ NA+RWI |
| [39] Ting Chen et. al (2022) | Improved CNN | CNN | Nuscenes-data set from the camera, LIDAR and RADAR | 32186/8560 /9041 set instances | CPU-Intel Xeon Gold 5118 (2.30GHz) GPU-RTX5000 | Python 3.8.8 PyTorch 1.10.0 CUDA 10.2 | 6seconds | FT of SDV for 50m | Simulation |
| [40] Qianxia Cao et. al (2021) | Four-layer LSTM with Yolov5 | LSTM | Real data- 4 hours of traffic flow video from surveillance cameras | Not reported | GPU | TensorFlow | 2seconds | FT at intersection | Simulation +NA+RTI |
| [41] Nejad et. al (2021) | Not reported | CNN_LSTM | High D- 147 hours,44500km with 25fps and 4k resolution | Not reported | GPU- Nvidia GeForce GTX 1050 and 8GB RAM | PyTorch | Not reported | 5 seconds of FT | Simulation +NA |
| [42] Wei Tian et. al (2022) | Not reported | LSTM | NGSIM, HighD and Argoverse | Not reported | Not reported | PyTorch | Not reported | 5 seconds of FT | Simulation +NA |
| [43] Izquierdo et. al (2022) | U net (6 layers) | CNN | PREVENTION-recorded at 16 Hz total of 6 hours | 9/-/2 sequences | Not reported | Not reported | Not reported | 2 seconds of FT | Simulation +NA |
| [44] Pin Lv et. al (2022) | PF_CNN_L STM | CNN_LSTM | Next-Generation Simulation (NGSIM) (45 min) | 31.5/4.5/9 minutes | CPU-Intel Core i9 9900, RAM 64 GB, GPU RTX 2080 | PyTorch | 0.2 seconds | FT of surrounding vehicles | Simulation +NA |
| [45] Bing zhou et. al (2022) | improved LaneGCN | LaneGCN | Argoverse total 324557 scenarios | 205942/394 72/78143 scenarios | 4 TITAN-X GPUs | PyTorch | 5 seconds | FT of SDV | Simulation +NA |
| [46] Shrivastav a et. al (2021) | clustered LSTM_RN N | LSTM_RNN | Real data from T drive using GPS contains 17.7 million data points | 14.16M/3.5 4M/- data points | Not reported | Tensorflow 2.0 | 0.96 seconds | FT of SDV | Simulation +NA |
| [47] Z Bai et. al (2018) | spatiotemp oral LSTM network | (Conv-LSTM) & (3D-CNN) | Comma-80GB raw image data and steering angle data | Not reported | CPU-Core (TM) i7- 6700 RAM-32 GB, GPU- GTX 980 | Python with Keras | Not reported | Steering angle for FT of SDV | Simulation +NA+RTI |
| [48] Song et. al (2018) | deep cascaded neural network | CNN & LSTM | ETS 2 simulator data set- 8 hours of driving data images and motion command recorded in 30FPS | 8 hours/-/3 scenes | CPU-Core i7-7700K @4.2 GHz, RAM-32GB, GPU-NVIDIA GTX 1080Ti | Ubuntu 16.04 and Caffe | 20 times prediction in 1 second | Steering angle | Simulation +NA+RTI |
| [49] Cui et.al (2019) | Not reported | CNN | 240 hours of real data using SDV with camera, lidar, radar | total 7.8 million data with a ratio of 3:1:1 | GPU-16 Nvidia Titan X | TensorFlow | 10ms | FT of SDV and pedestrian | Simulation +NA+RTI |
| [50] Grigoresc u et. al (2019) | Neuro Trajectory | CNN & LSTM | Synthetic data-GridSim and real data from the test car with camera, lidar, radar | Not reported | camera-MFC430 Lidar-Quanergy M8, Radar-ARS430 | GridSim | Not reported | FT of an ego-vehicle | Simulation+ NA+RWI |
| [51] Jeong et. al (2020) | Not reported | LSTM based RNNs | 484 vehicle trajectories -real data using SDV with camera, lidar, GPS | 11,662 / 4,998/- samples | CPU-i7 3.2Ghz RAM-16 GB, storage-512 GB | Not reported | 1.2 s | FT of surrounding vehicles | Simulation+ NA+RWI |
| [52] Leordean u et. al (2020) | Not reported | CNN | UED-Real data-21 h of driving videos at 30 fps | Not reported | Nvidia | Not reported | Not reported | Current location and FT | Simulation+ NA+RWI |

accounting for the unpredictability and variability of the motion. The LSTM-RNN model is equipped with the ability to forecast the future movements of nearby vehicles up to two seconds in advance and iwas developed using real-life vehicle trajectory information. The generated trajectory is reliable and optimal for the SDV to travel via the intersection, using anticipated trajectories as input. Furthermore, Marius Leordeanu and Iulia Paraicu [52] combine ocular localization with trajectory prediction to provide an approach for SDV navigation. The suggested approach employs a DNN to forecast the vehicle trajectory from visual input, and visual localization to determine the SDVs location in the driving environment. The authors created their dataset called as Urban European Driving (UED) Dataset and the Map, which contains 35km of driving data with a duration of 21 hours with LIDAR and camera sensors. The neural network can anticipate the vehicle's future course up to one second in advance after being trained using a UED dataset of images and matching LIDAR data. Even in enormous-scale situations, the approach can determine the exact location of the vehicle with great precision.

However, all the aforementioned DL-based trajectory planning algorithms were trained and evaluated in their corresponding dataset and experimented with their respective implementation method, the outcomes demonstrated that the aforementioned algorithms were better performed when compared with their corresponding state of art algorithms. Table 5 compares technical details of recently available various DL algorithms in Trajectory planning for SDVs based on the following parameters such as type of DL algorithm utilized, type of input, type of output, Type of Dataset, Software and Hardware Utilized, Response Time (RT) and way of Implementation, which are more important for the researcher to get deep knowledge in the field of trajectory planning for SDVs.

## III. DETAILED SURVEY ON END-TO-END LEARNING FOR SDVs

In machine learning one of the important approaches is E2EL in which a system learns to perform a task directly from raw input data to output predictions without relying on manual feature engineering or intermediate steps [53]. E2EL can be used in the context of SDV to train a neural network to operate a vehicle using sensory input like camera images or LIDAR data. The conventional method for developing SDV entails segmenting the issue into various modules, such as perception, localization, planning, and control [54]. Every component is designed to perform a certain function, with the output of a particular component serving as the input for the next.

However, this modular approach requires many human expertise in designing and tuning each module, and it can be difficult to integrate the modules into a cohesive system. In contrast, E2EL can simplify the development process by allowing the system to learn the entire driving task in a unified framework. Training the system involves using a sizable dataset of input-output pairings, in which the intended driving behaviours, such as steering angle, acceleration, and braking, are produced from the incoming sensory data. Without taking any intermediary stages, the neural network learns to directly map input to output. In the early years some survey papers are detailed discussed different architecture and training methods for E2EL [55] also some other researchers discussed driving datasets that are publicly available and simulated testing platforms [56]. In this section we discussed different E2EL techniques for SDV in the aspect of various important parameters such as hardware stack, software stack, type of simulator utilized and way of implementation. E2E self-driving is now conceivable because of the popularity of DL techniques. E2EL has shown optimistic outcomes in the creation of SDV. Everything started with an NVIDIA experiment [53], in which researchers used a CNN to steer a business vehicle utilizing just the frontal road's monocular picture as input.

Zhengyuan Yang et. al [57] present a unique method for SDV that combines multi-modal multi-task control with picture perception. The suggested technique makes use of a DNN to process multiple sources of sensory data, including visual, LIDAR, and RADAR inputs, to predict the optimal control actions for the vehicle. A trained model E2EL to manage multiple driving tasks, such as lane keeping, obstacle avoidance and intersection crossing. To identify control actions by extracting features from the provided input data, the authors combine CNNs and RNNs. The suggested strategy combines multiple modalities of sensory data, which improves the robustness of the system and allows for more accurate control decisions. Besides this, Junekyo Jhung et. al [58] developed a new approach for steering control in SDVs. The proposed approach maps raw sensor information to directional commands directly using a CNN, eliminating the need for intermediate feature extraction steps. Additionally, the model includes a closed-loop feedback system that adjusts the steering commands in real-time based on feedback from a front-facing camera.

Meanwhile, in vehicle-centric driving videos, Li Du et al. [59] devised a new technique for anticipating future frames (FFPRE). Without the need for explicit motion estimate or scene comprehension, the suggested solution is an E2E method that directly transfers previous video frames to subsequent ones. To determine the links between time and space in the entered data, the authors employ a DNN with both convolutional and recurrent layers. The capacity of the suggested method to learn intricate spatiotemporal patterns straight from unprocessed video data is one of its key advantages. This allows the model to make accurate predictions even in challenging scenarios, such as occlusions or sudden changes in the scene. Similarly, an innovative method for steering control in SDV is presented by Tianhao Wu et. al. in [60]. The suggested technique forecasts the appropriate steering angle for the vehicle using a complex neural network that considers both present and future spatiotemporal variables. The authors combine convolutional and recurrent layers to extract spatial and temporal characteristics from the input data, which includes pictures from a camera that faces forward and LiDAR data. The capacity of the suggested strategy to include future spatiotemporal aspects in the control choice is one of its main advantages. This allows the model to anticipate changes in the scene and adjust the steering angle, accordingly, enhancing the vehicle's performance and safety.

Further, A system for autonomous navigation employing DL and a multi-camera configuration with RGB and depth pictures was proposed by José A. Diaz Amado et al. [61]. The CNN employed in the neural network produces the vehicle's steering and throttle commands by using RGB and depth pictures as inputs. The four cameras that together make up the multi-camera framework employed in the paper have a full 360 degrees of perception. The RGB and depth images are captured simultaneously by each camera, resulting in a total of eight input streams for the neural network. With the help of a depth sensor and a stereo camera system, depth pictures are produced. This study offers a potential method for autonomous navigation based on DL and a multi-camera system, which may find use in the development of SDT. Furthermore, using DL techniques, Chanyoung Jung et. al. [62] suggested a novel method for SDVs. The goal of this approach is to make SDVs more reliable and accurate by estimating the time-to-line crossing (TLC) and using this information to change the SDV's speed and trajectory. One crucial part of SDV is the TLC, which calculates how long it takes the vehicle to reach the lane barrier. Based on data from the vehicle's sensors, including its cameras and LIDAR, the proposed system uses a DNN to forecast the TLC. Then, based on the anticipated TLC, the vehicle's speed and trajectory are modified to keep it in its lane.

Additionally, a novel method for SDT is presented by Myoung-jae Lee et. al. [63] employing an E2E DL algorithm. The suggested method makes use of a CNN that receives input from the vehicle's raw sensors data, such as front-facing camera pictures and steering angle data. Without explicitly extracting and selecting features, the CNN is trained to output the appropriate steering angle for the vehicle depending on the input data. Further, incorporating E2EL, Tanmay Vilas Samak et al. [64] provide a novel method for SDV. The recommended approach makes use of a DNN to discover an association between sensor data and vehicle control outputs. The neural network is trained using a technique called behavioural cloning, where the network has been trained to replicate the actions of a human driver. The authors give a thorough explanation of the approach's neural network architecture, this has multiple dense and convolutional layers. The approach was able to successfully navigate the vehicle through various scenarios, such as Simplistic driving, rigorous driving, and obstacle avoidance.

Meanwhile , *Simone Mentasti et. al* [65] propose a novel approach to lateral control of SDVs using a multi-state E2EL framework. The author argues that traditional lateral control methods for SDVs, such as model-based approaches or rule-based approaches, have limitations in terms of robustness and adaptability to different driving conditions. E2EL techniques, in contrast, have demonstrated promising outcomes in several applications. These methods learn directly from unprocessed sensor inputs to regulate outputs. The multi-state framework consists of two sets of state-specific neural networks, where each network is responsible for a specific lateral control task, such as anticipating situations and preserving the vehicle lane position by adjusting the steering angle. Furthermore, for forecasting steering angles in SDV, D V Prasad Mygapula et al. [66] developed a method to understand the relationship between sensor input data and appropriate steering angles, the suggested technique employs a CNN. The CNN is trained using the SullyChen dataset, which consists of pictures taken with a front-mounted camera and the accompanying steering angles. Test results on a model vehicle powered by batteries show that the recommended approach can accurately predict steering angles, with an $R^2$ score of 0.819 and a test loss of 0.0354.

In addition, a machine learning method for E2E motion planning in SDV with an optical flow distillation method was deployed by Hengli Wang et al. [67]. Based on input data from a camera positioned on the vehicle, the suggested technique employs a CNN to forecast steering angles and speeds. The CNN is trained using a NuScenes dataset that includes both image frames and ground truth steering and speed values. The suggested technique also uses optical flow distillation, which is a method for distilling optical flow information into a compact representation that can be applied to increase the precision of the direction and speed estimations. A closed loop CARLA simulated environment was used to assess the suggested methodology, and the results demonstrate its accuracy, which has a Success Rate of 88.67% and a Right Lane rate of 93.16%.

Furthermore, using multi-modal sensor fusion and DNN to classify scenes, Zhiyu Huang et al. [68] created a DL technique for E2E based SDVs. The proposed approach combines data from multiple sensors, including cameras, lidar and odometer, to provide the vehicle ability to sense its surroundings and make decisions to drive. The DNN used in the perception module is a CNN that is experimented on a CARLA simulator dataset of labelled sensor data. The network's goal is to discover the connections between the incoming sensor information and the related scene representation. A distinct DNN, used by the decision-making module, receives the scene representation as input and provides steering and speed directives

Besides this, the use of LIDAR point cloud data was suggested by Xianyong Yi et. al [69] in their unique self-driving approach. The author presents a DL framework that processes LIDAR data in an E2E way, enabling steering decisions to be made by the vehicle without the use of extra sensors or human input. Furthermore, Tinghan Wang et. al [70] introduced a novel approach for SDVs that is independent of irrelevant roadside objects, using an auto-encoder architecture. The author offers a DL method that processes camera images in an E2E way, allowing the vehicle to make direction-finding decisions without the need for additional sensors or human intervention. The study also includes an extensive analysis of the network's performance, showing that it is robust to variations in lighting conditions and different road environments.

Additionally, Jie Hu et. al [71] proposed a Bilateral Guide Network (BGNet) for enhancing scene understanding in self-driving using DL. The Driving Affordances Path (DAP) and Visual Guide Path (VGP) are part of BGNet. The author aims to develop a framework that can process sensor data from front camera images to generate an in-depth awareness of the surroundings, which can be used for safe and reliable self-driving. For semantic segmentation, the author uses a fully convolutional network (FCN)-based DNN. The FCN is trained on a CARLA simulator with an autopilot mode dataset of annotated images utilizing both supervised and unsupervised learning strategies. Meanwhile, A DL model for anticipating the steering behavior of SDVs utilizing a temporal and spatial attention mechanism is put forth by Lei Han et al. [72]. The suggested model is an E2E framework that receives a series of pictures from the SDV's front-view camera and outputs the associated steering angle. The objective of the algorithm is to accurately forecast the steering angle by learning temporal as well as spatial data from the input images. The developed model employs CNN architecture to learn spatial characteristics. The task of removing important elements from the source images belongs to CNN. The collected attributes are then supplied into an LSTM network, which is at the forefront of capturing the temporal relationships among the input images. At each time step, the suggested model employs a selective attention mechanism to concentrate on specific parts of the input image. By applying the attention mechanism in both spatial and temporal dimensions, the model is able to focus on essential areas of the image and time steps that are crucial for steering angle prediction.

Besides this, Satya R. Jaladi et. al [73] developed a gamification framework to train and evaluate E2E models for learning human highway driving. The developed framework consists of a game-like environment where the player (i.e., the model is trained) drives a car on a highway and earns rewards for completing tasks such as staying within lanes, avoiding collisions, and keeping a secure distance from other vehicles. The suggested framework demonstrates how gamification can enhance E2E systems for human roadway driving in terms of accuracy and efficacy. Meanwhile, an incremental E2EL strategy for lateral control in SDV was proposed by Jaerock Kwon et al. [74]. The suggested method makes use of a DNN architecture to extract the correct steering angle from pictures captured by a vehicle's frontal camera. The information is gathered from the CARSIM simulation tool employing HIL, this includes an actual motorsport wheel with shifting gears and pedals. The suggested model was simulated using the Gazebo/ROS back-end OSCAR (Open-Source Robotic Car Architecture for Research and Education) simulator.

Further, by combining data from several sensors to create a 3D representation of the environment, Nguyen Thi Hoai Thu et al. [75] created a technique for motion planning in SDV. The suggested method generates the appropriate lateral control angle and velocity for the vehicle using an E2E RegNet Y 16GF DL model. The sensor fusion is achieved by implementing a transformer encoder. This technique fuses LiDAR and camera data to accurately detect and track obstacles in the vehicle's path. The vehicle's safe and effective trajectory is then planned to use the 3D map produced from the sensor data Furthermore, a novel E2E approach to SDV is put out by Oskar Natan et al. [76] that makes use of systems with multiple agents and semantic depth cloud mapping. The goal of the project is to create a system that is resilient to variations in weather and lighting and can function in intricate and dynamic surroundings. The suggested system relies on a DNN framework that accepts a collection of RGB images, LiDAR point clouds, GPS, and Speedometer data as input. The authors use a semantic depth cloud mapping approach to generate a 3D representation of the environment that incorporates both geometric and semantic information. The system is able to operate in intricate and dynamic surroundings thanks to the recommended network, which is also resistant to variations in weather and lighting. However, all the aforementioned DL-based E2EL algorithms were trained and evaluated in their corresponding dataset and experimented with their respective implementation method, the outcomes demonstrated that the aforementioned algorithms were better performed when compared with their corresponding state of art algorithms.

Table 6 compares technical details of recently available various E2EL techniques for SDV depending on the following parameters such as type of DL algorithm utilized, type of input, pre-processed image size, type of output, Type of Dataset, simulator, Software frameworks, Hardware Utilized and way of Implementation, which are more essential for the researcher to learn in-depth information about the topic of E2EL for SDVs.

Further, the E2EL approach are divided into two primary types based on the learning approaches employed: Imitation Learning (IL) through supervised learning and Reinforcement Learning (RL), which integrates unsupervised learning techniques.

### A. IMITATION LEARNING

IL is a machine learning technique used in the development of SDV to help them learn from expert demonstrations. In this context, expert demonstrations are often supplied by human drivers or simulated scenarios in which human-like driving behaviour is used as examples for the SDVs to replicate. IL enables SDVs to learn how to navigate complicated environments, follow traffic rules, and manage a range of driving situations. IL further classified into three categories as Behaviour Cloning (BC), Direct Policy Learning (DPL) and Inverse Reinforcement Learning (IRL).

#### 1) BEHAVIOUR CLONING

The most common method of IL in the field of SDV is known as BC, which has emerged as the dominating

**TABLE 6. Technical comparison of different deep learning algorithms in end-to-end learning for SDVs.**

| Paper with Year | Proposed method | DL algorithm | Dataset | Input data | Pre-processed Image Size | Simulator | Hardware | Software frameworks | Output | Implementa-tion |
|---|---|---|---|---|---|---|---|---|---|---|
| [57] Zhengyuan Yang (2018) | Multimodel MultiTask vehicle control (MMVC) | CNN | Udacity and SAIC | RGB image and actual speed | 200X66 | Udacity | Not Reported | Not Reported | Steering angle and Speed control | Simulation +NA |
| [58] Junekyo Jhung (2018) | DAVE_2SKY | CNN | Real data Set 2h of driving in Yonsai University | Image with size of 160 × 90 | 160X40 | Prescan | Sekonix camera DriveTm PX2 Computer | Prescan with Caffe and MATLAB Simulink | Steering wheel angle | Simulation +RWI |
| [59] Li Du (2019) | FFPRE | CNN with RNN | DR (eye)VE NVIDIA | RGB image | 224X360 | DR (eye)VE | 8G GTX 1080 GPU | PyTorch | Steering angle | Simulation |
| [60] Tianhao Wu (2019) | MSINet | CNN with LSTM | Udacity | Three front-view camera images 480X640 | 3X480X640 | Udacity | 4 Geforce GTX Titan GPU | Simulation Pytorch | Steering angle | Simulation +RWI |
| | | | UESTC | real-time 1280X1024 | 3X480X640 | NA | single Geforce GTX Titan GPU | ROS | | |
| [61]José A. Diaz Amado (2019) | Modified Pilotnet | CNN | V-REP simulator Data Set | Three front-view cameras output merged as a single image | 66x200 | V-REP simulator | 1/4 scaled Electric car with Jetson TX2 Developer Kit | ROS in Linux 16.04 OS with Python/C++ | Steering angle | Simulation +RWI |
| [62] Chanyoung Jung (2020) | Time to Line Crossing (TLC) | bi-CLSTM | Udacity and Custom test | Three front-view camera images | 640X480X3 | Udacity | 2 NVIDIA RTX-2080TI GPU | PyTorch | Steering angle | Simulation +RWI |
| [63] Myoung-jae Lee (2020) | Not Reported | CNN with LSTM | Euro Truck Simulator dataset 300Gb | One front view camera image | 80X200 | Euro Truck Simulator | AMD Ryzen Threadripper 2950x, RAM-128 GB, GPU-RTX 2080Ti (x2),1TB SSD + 8TB HDD | Euro Truck Simulator with LogitechG29 | Steering angle | HIL |
| [64] Tanmay Vilas Samak (2020) | Not Reported | CNN | Udacity Modified Environment | One front camera image 320×160 | 64X64 | Unity game engine | CPU-Intel I7-8750H GPU-NVIDIA RTX 2070 | Udacity, Python 3.6.8, Tensorflow 1.14.0 | Speed, Throttle, Brake and Steering Angle | Simulation +NA |
| [65]Simone Mentasti (2020) | Two CNN | CNN | Monza ENI circuit | Front camera image | 1)200X66 2)672X376 | Assetto Corsa | GPU-Nvidia 1660Ti | Not Reported | Scenario and steering angle | Simulation |
| [66]D V Prasad Mygapula (2021) | CNN model 2 | CNN | Sully Chen | Front camera image 370X110 | 250X70 | NA | Model Electric Car with Webcam and Jetsen TX1 processor | Not Reported | Steering Angle | Simulation +RWI |
| [67]Hengli Wang (2021) | IVMP | CNN | NuScenes | Surrounding view images | Not Reported | CARLA simulator | 2 NVIDIA GeForce RTX-2080TI GPU | Not Reported | FT of 5s | Simulation |
| [68]Zhiyu Huang (2021) | MSF_SU | CNN | CARLA urban scene | Front camera image 800X600 | 224×224 | CARLA simulator | NVIDIA RTX 2080Ti GPU | Not Reported | Steering angle and Speed control | Simulation |
| [69]Xianyong Yi (2022) | Modified Pointnet++ | DCNN | 3 hours and 300000 frames | LiDAR | NA | CARLA simulator | GPU - NVIDIA GeForce RTX 3050 Ti | Ubuntu18.04 Python 3.7, Pytorch | Steering Angle | Simulation |

**TABLE 6.** *(Continued.)* **Technical comparison of different deep learning algorithms in end-to-end learning for SDVs.**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| [70]Tinghan Wang (2022) | Auto-Encoder (ANN) | CNN | PreScan dataset | Front camera image size 240 × 320 × 3 | 50 × 50 × 3 | PreScan | Not Reported | TensorFlow | Steering Angle | Simulation |
| [71]Jie Hu (2022) | Bilateral Guide Network (BGNet) | CNN with GRU | autopilot of CARLA data set of 6000 images | Front Camera image | Not Reported | CARLA simulator | single RTX 3090 | Not Reported | Speed, Steering, Throttle, Brake | Simulation +NA |
| [72]Lei Han (2022) | V + A + SAtt + Tatt | CNN with Conv LSTM | Comma2k19 and Udacity 25,832 sequence samples | image size 1164X 874 and 1920 X 1200 actual steering wheel angle | 80 X 240 X 3 | Udacity | GeForce RTX 2080Ti GPU. | TensorFlow | Steering Angle | Simulation +NA |
| [73]Satya R. Jaladi (2022) | Modified VGG 19 | CNN | Grand Theft Auto V data set | 70, 000 images 480 P resolution | 200x60 | Grand Theft Auto V | Xbox controller, mobile RTX 2060 with 6GB | TensorFlow 2.0, Python | steering Angle and throttle | Simulation |
| [74]Jaerock Kwon (2022) | (Bio-Inspired Machine Intelligence Network) BIMINet | CNN | CARSIM simulator data set | front camera image size 800X151 | 160X160 | OSCAR and Gazebo/ROS simulator | CPU: Intel i7-6700 3.40 GHz, RAM: 32 GB, GPU: NVIDIA GeForce GTX745 | ROS with Ubuntu18.04.5 LTS OS, CUDA 9 and cudnn 7.1.2, | Steering Angle | HIL |
| [75]Nguyen Thi Hoai Thu (2023) | pre-trained RegNet Y 16GF | CNN_LSTM | Longest6 dataset from CARLA simulator | camera images 704 × 160 × 3 and LiDAR point clouds 256 × 256 × 2 | 704 × 160 | CARLA simulator | Not Reported | Pytorch 1.11 CUDA 11.3. | Depth Estimation, FT, BEV semantic Estimation | Simulation +NA |
| [76]Oskar Natan (2023) | Not Reported | GRU | CARLA simulator modified dataset | RGBD camera image 300×400, GPS, and speedometer | 256 × 256 | CARLA simulator | NVIDIA GeForce RTX 3090 | Ubuntu 20 PyTorch | level of steering, throttle and brake | Simulation +NA |

strategy [53], [77]. Within the framework of this approach, after using expert trajectories to train the model, the agent uses a classifier/regressor to replicate the policy. In order to learn the intended policy, the BC technique is a passive approach that does not involve active participation. Rather, it simply observes the entire command execution process. This presupposes that the state-action pairs that comprise each and every trajectory are independent of one another.

Bojarski et al. [53] pioneered BC, training a CNN to predict steering commands from monocular camera images for lateral control. However, it lacks longitudinal control. In contrast, Codevilla et al. [78] introduced conditional imitation learning (CIL), which includes both lateral and longitudinal control. Using inputs like images and high-level commands, CIL produces longitude and latitude control commands, marking a milestone in self-driving imitation learning with CNNs.

Adding to the CIL framework [78], researchers combine geographical information, preplanned path, or point clouds [79], [80], and [81] to improve robustness and generalisation. These approaches are not interpretable, even with advantages like faster feedback and less uncertainty. In order to lessen this, layers of intermediate representation are added. The direct perception method is proposed by Chen et al. [82] and predicts affordances in urban self-driving scenarios. Low-level controller operations are informed by these affordances, which are displayed in Bird's Eye View (BEV). This is furthered by Sauer et al. approach [83] Conditional Affordance Learning (CAL) shown in figure 7. which excels in complex urban traffic and uses video data and high-level commands for intermediate representations. Additionally, utilising LiDAR data and HD Maps, Urtasun's team presents interpretable end-to-end planners [84], [85] that enable safer trajectory predictions in comparison to using monocular pictures alone.

The key characteristic of the BC method is its reliance on expert-generated training examples, as a result, a portion of the states experienced during policy execution make up the training dataset. Consequently, if the dataset suffers from bias or overfitting, the method's ability to generalize is constrained. Additionally, when the agent encounters unfamiliar states, learning the appropriate recovery behaviour becomes challenging.

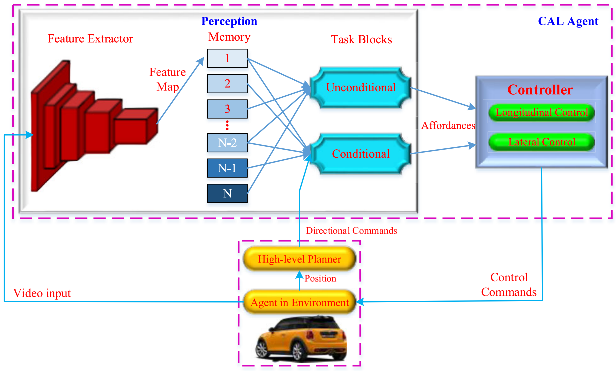


**FIGURE 7.** CAL model proposed by [83], The camera picture and instructions are sent to the CAL agent by CARLA. The last N maps are stored while the image is transformed into a feature map. These aid in affordance prediction along with instructions. The way that temporal features are used varies among tasks. CARLA's updates and observations are computed by the controller.

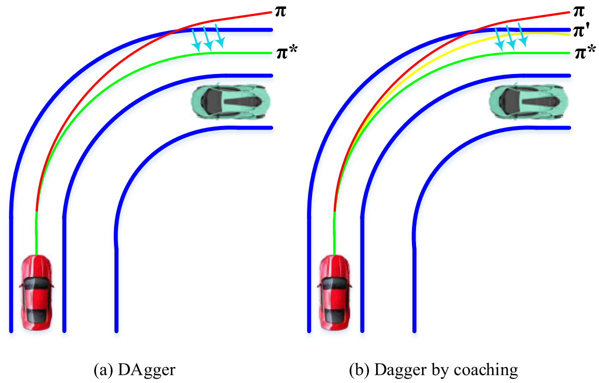


**FIGURE 8.** Dagger methods for SDVs proposed by [86] and [88].

### 2) DIRECT POLICY LEARNING

Direct Policy Learning (DPL) is a training methodology that originates from BC. DPL operates by evaluating the existing policy and obtaining training data that is more suitable for promoting self-optimization. In contrast to BC, DPL makes use of professional driving trajectories to help the agent fix present mistakes. This attribute of DPL mitigates the constraints of BC that arise from insufficient data. We give an outline of some DPL approaches in the section that follows.

Ross et al. [86] introduced Dataset Aggregation (DAgger), an online imitation learning technique based on the Follow-the-Leader algorithm [87]. DAgger uses every state-action combination it encounters to actively modify the principal classification tool or regression tool, viewing each validation repetition as an e-learning example. Which is shown in figure 8(a). Nonetheless, the discrepancy between the policy space and the learning space can impair its learning efficacy. In response, He et al. [88] proposed DAgger by Coaching shown in figure 8(b), which employs a coach to demonstrate easily learnable policies. These demonstrated policies gradually converge to the true label. The coach produces a balanced policy that is noticeably superior than the novice's anticipated actions but not appreciably worse than the real controlling signal.

Other researchers have highlighted drawbacks in DAgger methods [86], [88], such as inadequate generalisation, imprecise gathering of data, and ineffective query procedures. To address these issues, Zhang et al. [89] introduced the SafeDAgger procedure, focusing on enhancing query efficiency and reducing reliance on label accuracy. Additionally, Hoque et al. [90] proposed the ThriftyDAgger framework, incorporating human interaction in unusual circumstances, while Yan et al. [91] presented a new DPL training initiatives for mapless scenarios' navigation tasks, both aimed at improving model generalization and robustness.

DPL is a web-based learning policy that is iterative and reduces the amount and distribution of datasets needed. It also makes it easier to update policies over time by efficiently removing negative data.

### 3) INVERSE REINFORCEMENT LEARNING

In order to overcome the shortcomings of the aforementioned techniques, IRL explores for the fundamental causes of the association between inputs and outputs. IRL takes a new approach to training model for different tasks. Instead of needing pre-programmed rewards or perfect demonstrations, IRL observes how an expert performs a task (their "trajectories") and tries to figure out what motivates them (the reward function). It then uses this inferred reward function to train a policy (its decision-making system) to achieve similar goals. IRL has three main approaches: max-margin, Bayesian, and maximum entropy methods.

The max-margin method optimises reward functions by increasing the difference between optimal and suboptimal policies using expert trajectories. Several methods linearly aggregate data to show reward functions as independent. Andrew Wu [92] developed three reward function refinement techniques and the first max-margin IRL approach. Pieter et al. [93] optimised Wu's method to uncover latent weight-feature links by treating expert reward functions as explicitly created linear combinations of known features.

Quality and distribution of expert trajectories limit these techniques. Umar et al. [94] propose game-theoretic IRL multiplicative weights for apprenticeship learning. The agent receives feature weight policy knowledge and updates the reward function using linear programming to reach a stationary policy. An interpretable planning system proposed by Phan-Minh et al. [95] shown in figure 9. It generates trajectory, filters safety, and scores trajectory. Perceptual data predicts future trajectories, an interpretable safety filter assures basic safety, and DeepIRL assesses predicted trajectories. [96] and [97] introduce preference and inference formulations to allow users choose actions based on preferences, enhancing model performance.

Bayesian approaches, the second element of IRL, maximise reward posterior distributions by using the optimised trajectory or prior distribution. Ramachandran et al. [98]

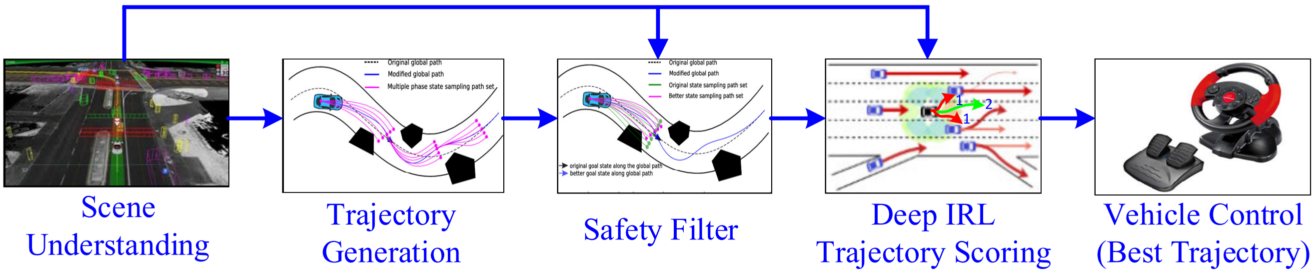


**FIGURE 9.** Deep IRL trajectory scoring methods for SDVs proposed by [62].

proposed the Bayesian IRL model, using previous distributions to infer a posterior distribution of the predicted reward variable. Levine et al. [99] added a kernel function to the Bayesian IRL model [98] to advance reward prediction and unseen driving performance. Moreover, Brown et al. [100] use sampling to construct a Bayesian IRL model, estimating upper bounds on return differences without the need for a reward function by utilising expert trajectories. In a different study, Palan et al. [101] provide the DemPref model, which addresses efficiency concerns in conventional approaches and improves query quality by using trajectory data for a simple reward function and active query development. Notably, DemPref does not depend exclusively on expert trajectories at the highest level. IRL's third component is the maximum entropy approach, which estimates the reward function during optimisation.

Maximum entropy methods are better for continuous spaces than prior IRL methods and may reduce expert trajectories' negative effects. Ziebart [102] presented the Maximum Entropy IRL model, which mitigates noise and poor behaviour in the expert trajectory, comparable to [92]. The agent linearly maps features to rewards to optimise the reward function under supervision.

Many studies [103], [104], [105] have used maximum entropy IRL in real world SDV application. The algorithm Generative Adversarial Imitation Learning (GAIL) [104] is crucial to this subject. Using a generative adversarial network (GAN), GAIL model's expert trajectory distributions to reduce state drift from limited datasets. Expert trajectory reconstruction and policy development allow GAIL to function like human drivers in particular circumstances. Co-GAIL [106], InfoGAIL [107] and Directed-InfoGAIL [108], build upon the groundwork laid by [104] and provide competitive outcomes in numerous application fields.

IRL offers numerous valuable contributions to SDV technology. Nevertheless, similar to the approaches mentioned earlier, it faces challenges in addressing rare cases. Enhancing the robustness and interpretability of IRL effectively represents a forthcoming area of research.

IL experts leads to action. Summary of above discussed IL methods shown in Table 7. It may be less possible to generalise a dataset with overfitting or an uneven distribution. The agent acts erratically when led to an uncertain state. A lot of academics use virtual and real data along with data enrichment to improve dataset dispersion. These initiatives guarantee generalizability and competitiveness of the methods.

### B. REINFORCEMENT LEARNING

IL techniques necessitate a large amount of personally labelled data, and drivers may make different decisions in identical situations, causing training uncertainty. Researchers avoid labelled data with RL algorithms for E2EL. RL trial-and-error agents maximize environmental numerical rewards. In constant interaction with the environment, the agent learns the best goal-achieving policy. According to this framework, two primary methods in RL are utilized to attain optimal policies: value-based RL and policy-based RL. Moreover, hierarchical reinforcement learning (HRL) and multi-agent reinforcement learning (MARL) are considered effective strategies derived from these approaches, showing promise in resolving intricate problems and aligning well with real-world driving situations. Utilizing RL techniques for training SDVs has emerged as a burgeoning trend in E2EL for SDV research.

#### 1) VALUE BASED RL

Value-based RL seeks to evaluate various activities in a state and provide a value to every action according to the anticipated reward it provides. The agent gains the ability to associate actions and states with rewards, and it uses this knowledge to make the best choices.

Among value-based techniques, Q-learning [109] is well-known. Mnih et al. [110] introduced the first DL technique based on Q-learning, which learned control signals straight from screenshots.

In order to address stability difficulties with high-dimensional perception data, Wolf et al. [111] also integrate Q-learning into SDV systems, defining driving manoeuvres and selecting them based on picture information.

The suggested conditional DQN [112] technique shown in figure 10. It improves predictive stability and, in certain cases, achieves performance close to human driving. Alizadeh et al. [113] use a DNN and a DQN agent to make high-level decisions while dynamically striking a balance between safety and agility. By merging DQN from control model, Ronecker et al. [114] suggest a harmless navigation

**TABLE 7.** Summary of imitation learning techniques.

| Method | Performance | Advantages | Disadvantages | Limitations | Implemented In |
|---|---|---|---|---|---|
| Behavior Cloning (BC) | High (when expert demonstrations are good) | • Simple to implement It learn complex behaviors | • Relies on good quality expert demonstrations<br>• It is susceptible to errors in the expert demonstrations | • It might not perform well to unobserved circumstances | [53], [77] – [85] |
| Direct Policy Learning (DPL) | Potentially high, but requires careful design of reward function | • Can learn complex behaviors<br>• It incorporates feedback from a human expert | • Can be susceptible to errors in the expert demonstrations<br>• It is computationally expensive | • It require high computing resources.<br>• Requires careful design of the interactive demonstrator | [86] – [91] |
| Inverse Reinforcement Learning (IRL) | It is good, but requires defining the desired behavior | • It learns from human preferences | • Requires defining a reward function that captures the desired behavior<br>• It is sensitive to the choice of reward function | • Can be sensitive to the choice of reward function<br>• Defining the desired behavior can be challenging | [92] – [108] |

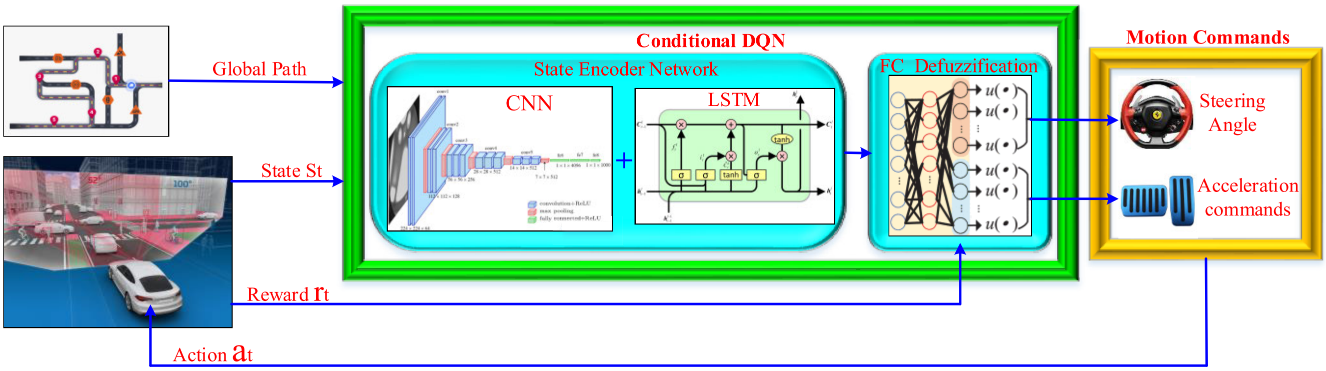


**FIGURE 10.** Architecture of conditional DQN [112].

technique for cars on highways, exhibiting effective and SDV behaviour in road traffic circumstances.

Constrained Policy Optimisation (CPO) [115] is a all-purpose algorithm that guarantees near-constraint satisfaction in each iteration, responding to security concerns in E2EL for SDV. Li et al. [116] incorporate a risk awareness algorithm for safer lane changes into DRL frameworks. Chow et al. [117] present safe policy optimisation techniques that tackle issues in constrained Markov decision processes (CMDP) by employing a Lyapunov-based methodology [118]. By combining policy and neural barrier certificate learning, Yang et al. [119] create a model-free safe reinforcement learning algorithm. Mo et al. [120] use Monte Carlo Tree Search to lessen risky actions when doing overtakes on roads.

### 2) POLICY - BASED RL

The value-based technique only allows for discrete commands, but SDV necessitates continuous control for fine-grained modifications. Policy-based approaches, on the other hand, perform well in multidimensional actions environments with continuous instructions, providing stronger convergence and investigation capabilities compared to value-based methods.

Implementing RL in real-world SDVs presents considerable hurdles. Kendall et al. [121] proposed actor-critic algorithms shown in figure 11. It is used the Deep Deterministic Policy Gradient (DDPG) algorithm [122] to achieve human-level effectiveness for lane-following with only monochromatic photos. Wang et al. [123] proposed a solution based on human expertise lane-change policy that is suitable to single or numerous vehicles and doesn't depend on V2X interaction.

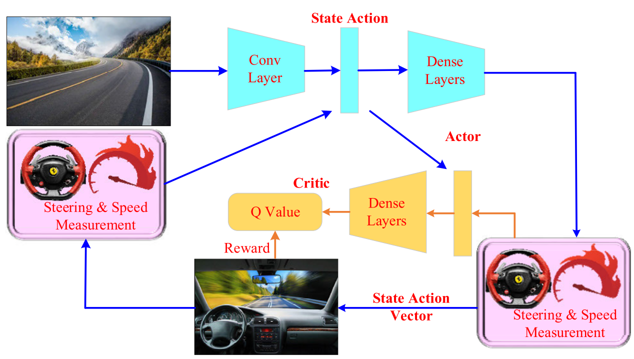


**FIGURE 11.** Actor – Critic algorithm proposed by [121].

To address crowded traffic conditions, Saxena et al. [124] used the proximal policy optimisation (PPO) approach [125] to train an enforcement policy by simulating relationships with other vehicles with the aim to minimise crashes and improve the comfort of travellers. Ye et al. [126] enhanced their work by using PPO to automate switching lanes on real highways, assuring avoidance of crashes and seamless driving. Other research has demonstrated [127], [128] the effectiveness of PPO-based RL technique in E2EL for SDV, with enhanced policy learning efficiency and trajectory exploration diversity.

Learning RL guidelines from scratch can be tedious. Mixing RL with techniques such as IL and curriculum learning offers a viable approach. Liang et al. [129] used IL and DDPG to improve exploration effectiveness in continuous space, offering a configurable gating system for centralised management. Tian et al. [130] employed RL to learn from expert knowledge in following trajectory tasks, adjusting them with both IL and continual RL algorithms.

Huang et al. [131] improved the training effectiveness of RL algorithms by incorporating human prior experiences, solving the long-tail problem of SDVs through professional human expertise. Wu et al. [132] suggested a human guidance-based RL technique that prioritises knowledge replay, which improves effectiveness and efficacy in challenging circumstances. Hence, enhancing driving task effectiveness may necessitate combining several strategies and creating training techniques tailored to particular tasks.

### 3) HIERARCHICAL RL

While RL techniques demonstrate promise in many areas, they are criticised for their difficult training procedure, which is especially problematic in SDVs because of dynamic circumstances and multidimensional input information, which result in lengthy period of training and utilisation of resources [133]. To tackle this, HRL divides the main job into a hierarchy of smaller responsibilities, each with a distinct objective and set of rules. The agent can handle lesser-sized subproblems due to this hierarchical organisational structure, which lowers learning difficulty and improves manageability.

For lane-changing tasks, Chen et al. [134] recommend a two-level approach in which the low-level system carries out the process of selected instructions, while the high-level system decides whether to carry out a lane shift. Furthermore, [135] and [136] extended the research based on methodology [134] by integrating additional data, such as vehicle heading angle, speed, and position, to improve the effectiveness of the low-level control system. These approaches present viable ways to create reliable and secure SDVs capabilities.

The body of research on HRL's ability to generalise is growing. Using a high-level layer and a kernel-based lowest-squares policy repetition technique, Lu et al. [137] present an HRL technique for self-deciding and mobility management in unpredictable traffic situations shown in figure Z. To improve model generalizability, Duan et al. [138] split mobility responsibilities into three different models using a centralised policy network. Building on earlier research, Cola-HRL [139] combines a continuous-lattice state space representation, low-level controller, and high-level planner to provide higher making decisions efficiency across a range of scenarios as in comparison with state-of-the-art techniques.

### 4) MULTI-AGENT RL

MARL addresses situations in which heterogeneous traffic players engage in mutual influence, thereby substantially impacting one other's policies [140]. Others' actions in single-agent systems frequently conform to predetermined guidelines, which results in overfitting and determinism regulations [141]. MARL commonly uses Decentralised Partially Observable Markov Decision Processes (DEC-POMDPs) with the goal of learning decision-making strategies for multiple agents. However, the rapid growth of the state space with agent numbers is a barrier for Multi-Agent System (MAS) [142] training.

Designing efficient learning algorithms is one way of dealing with dimensionality problems. In order to empower both collaborative and adversarial endeavours, Kaushik et al. [143] use parameter-sharing Deep Deterministic Policy Gradient (DDPG) for twin assignments, injecting assignments into the observation space. Wang et al. [144] combine the exchange of graph data across agents in a variety of circumstances, utilising Proximal Policy Optimisation (PPO) to generate actions continuously and permitting interaction among vehicles within a predetermined range.

MARL provides a global viewpoint for multi-vehicle management, whereas Reinforcement Learning (RL) for lane-changing decisions is mostly single-agent oriented. In mixed-traffic highway circumstances, Zhou et al.

Reference [145] discuss SDV lane shifts in conjunction with human-driven vehicle judgements. MARL approaches seem promising for handling complicated planning and decision-making challenges, even beyond lesser assignments. Chen et al. [146], for example, train agents to avoid crashes in scenarios with time-critical converging highways.

Giving credit is important in collaborative systems with multiple agents. Using a collaborative policy learning technique, Han et al. [147] offer a reward shifting mechanism to promote permanent cooperation between SDVs. Peng et al. [148] achieve higher performance across several measures by incorporating psychological socialisation principles into Coordinated Policy Optimisation (CoPO) shown in figure X for Self-Driven Particles (SDP) structures.

Although RL is popular for self-directed learning, insufficient sample effectiveness persists as a problem. While deep neural networks help with approximation of functions and learning representations, yet interpretability remains tough. Summary of Different RL methods shown in Table 8.

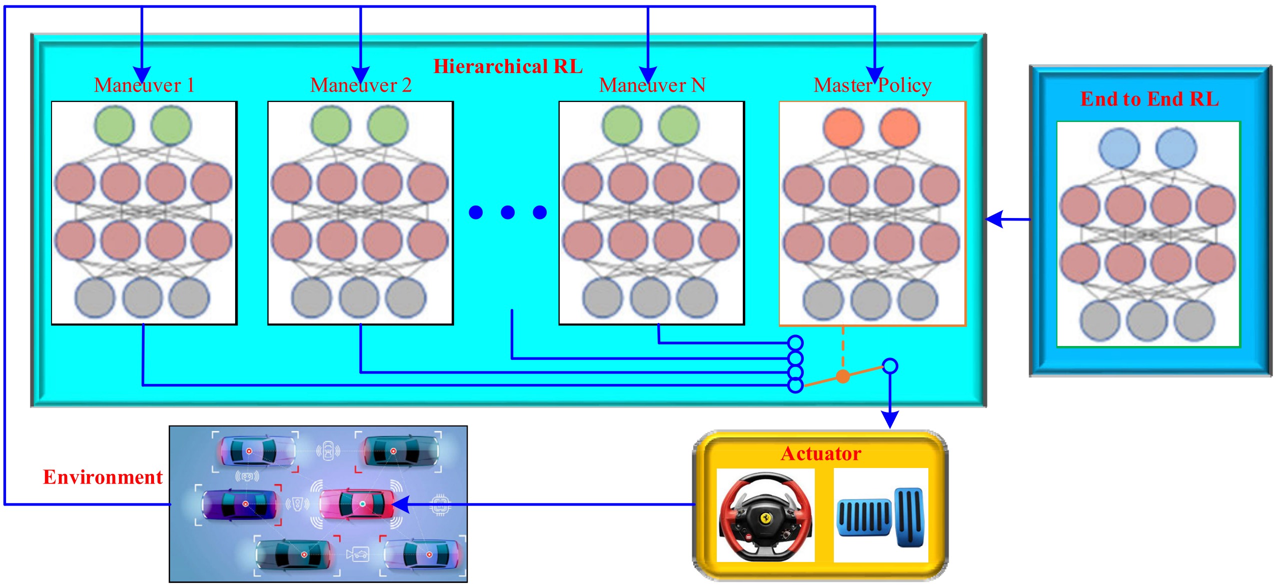


**FIGURE 12.** Hierarchical RL architecture proposed by [138].

**TABLE 8.** Summary of reinforcement learning techniques.

| Method | Performance | Advantages | Disadvantages | Limitations | Implemented in |
|---|---|---|---|---|---|
| Value-Based RL (e.g., Q-Learning) | Good for simple tasks | • Easy to implement<br>• It learns from trial and error | • Slow convergence, Curse of dimensionality (high dimensional state space)<br>• Difficulty in handling sparse rewards | • Not suitable for complex driving scenarios with high-dimensional state spaces | [109 – [120] |
| Policy-Based RL (e.g., DDPG, PPO) | High potential for complex tasks | • Efficient learning<br>• It handles continuous control problems | • Sensitive to hyperparameter tuning<br>• It is prone to instability | • Requires careful design of the policy architecture and exploration strategy | [121] – [132] |
| Hierarchical RL | Can handle complex tasks with sub-goals | • Decomposes complex tasks<br>• It improves learning efficiency | • Increased complexity, It Requires defining sub-goals and reward structure for each level | • Designing hierarchical structure and reward functions can be challenging | [133] – [139] |
| Multi-Agent RL (MARL) | Suitable for interactions with other vehicles and pedestrians | • Considers interactions with other agents<br>• It improves decision-making in multi-agent environments | • High computational complexity, Difficulty in defining reward functions for cooperation | • Requires significant computational resources and careful design of reward functions to encourage cooperation | [140] – [148] |

## IV. PRACTICAL ENABLERS IN SDV DEVELOPMENTS

Practical enablers like datasets and simulators play crucial roles in the development and advancement of SDV technologies. These tools provide essential resources and environments for training, testing, and validating SDV, ultimately contributing to the safe and efficient deployment of SDVs on public roads.

### A. DATASET

Datasets are collections of labelled sensor data captured from real-world driving scenarios. These datasets contain various types of information, including images, lidar scans, radar readings, and GPS coordinates, annotated with labels such as object classifications, lane markings, traffic signs and vehicle trajectories. Datasets serve as training inputs for Machine learning algorithms, allowing SDV systems to learn to recognize and interpret different objects, obstacles, and environmental cues. High-quality and diverse datasets are essential for training robust and reliable self-driving models capable of handling a wide range of driving conditions and scenarios. Even though creating and putting up own datasets for SDVs takes time, there are many common and significant datasets already available that may be used for study, in this section we detailed discussed about various open-source dataset for SDVs.

A2D2 [149] With almost 41,000 labelled cases and 38 characteristics, the Audi Autonomous Driving Dataset (A2D2) has a total size of roughly 2.3 TB. Sorted according to the type of annotation, it includes 3D bounding boxes and semantic segmentation.

ApolloScape [150] is a dynamic project that aims to advance multiple areas of SDV. It provides 1000km trajectories for urban traffic, 80,000 lidar point clouds, and over 100,000 street view frames.

Another notable dataset for 3D object tracking and motion predictions is Argoverse 1 [151]. It provides extensive sensor data, such as LiDAR point clouds, forward-facing stereoscopic pictures, and 360-degree pictures from seven cameras. Thanks to its diversified sensor data and semantic maps, Argoverse, which covers over 300,000 vehicle trajectories collected from 290 km of mapped lanes, offers rich knowledge necessary for furthering research and development in prediction systems.

Berkeley DeepDrive [152] This dataset, also referred to as BDD 100K, offers 100,000 annotated films and ten tasks for assessing image recognition software. It includes information on geographic and meteorological diversity, over 100 million frames, and more than 1000 hours of driving experience.

Cityscapes [153] provides a large dataset that has been collected in complicated urban environments. It carefully annotates pictures offering pixel-by-pixel segmentation for thirty distinct classes, such as cars, people, streets, and traffic signals. Cityscapes is a well-known example of a difficult baseline for urbanised semantic segmentation tasks. The Comma.ai Driving Dataset [154] captures real-world driving scenarios from a Tesla Model S using cameras, LiDAR, GPS, and IMU sensors. This diverse data (112,000 video frames) is valuable for training SDV algorithms in tasks like object detection and lane following. While full access might be limited, it offers a glimpse into real-world driving data for researchers in this field.

Benchmarking the KITTI Vision Suite [155] The 2012 release of the KITTI dataset, which includes real-world computer vision benchmarks, made it a pioneer in the field of SDV research. It has received more than 4000 scholarly citations.

Lyft Level 5 [156] provides more than 1,000 hours of data, making it a noteworthy dataset for motion prediction in SDVs. In addition to 17,000 sceneries, it has an 8,500-lane segment high-resolution aerial image and a high-definition semantic mapping with over 15,000 human annotations. It is an essential standard for SDVs, assisting activities like mobility planning and forecasting with its rich annotations and broadened data.

A vital tool for SDVs, nuScenes [157] provides a variety of datasets suited to the requirements of perception systems. Using LiDAR, radars, and cameras, it gathers data from metropolitan areas in Boston and Singapore. Six cameras provide detailed environmental views. This dataset is extensively used for multi-view object identification tasks and, by enabling a wide range of activities and establishing new industry standards.

The Waymo Open Dataset [158] introduced in 2019, significantly contributes to the advancement of SDV research. It has a major impact on the field by providing a substantial amount of multimodal sensory information with thorough annotations. Especially, the dataset's extensive coverage of a wide range of driving situations and geographical areas improves the practicality and resilience of several tasks like as tracking, segmentation, and detection.

The highD dataset [159] provides a comprehensive collection of realistic vehicle trajectories on German roads. Among them are 110,000 vehicles and trucks' refined trajectories. This dataset tackles the limitations of existing scenario-based safety validation measurement approaches, which frequently lack enough high-quality data and realistic road behaviour among users.

The INTERACTION dataset [160] comprises an extensive semantic map and covers a broad spectrum of complex navigation scenarios. For diverse tasks like mobility forecasting, imitation learning, and judgement validation, this feature makes it versatile. The integration of data from several nations improves the analysis of driving behaviours across cultural differences, which is important for the advancement of SDVs worldwide.

Argoverse 2 [161], an expansion of Argoverse 1 [148], offers the largest dataset for SDVs. It features complex driving scenarios and vital functions like 3D object tracking. Covering six cities and various situations, this dataset provides multimodal data supporting algorithmic advancements in SDV development.

Talk2BEV [162] is a pioneering dataset that combines vision-language models with BEV maps for SDV applications. With over 20,000 human-annotated inquiry types, it enhances understanding of maneuver scenarios using state-of-the-art techniques. The Talk2BEV-Bench standard supports activities such as intent prediction and decision-making, offering a versatile framework for research and progress.

The IDD-3D [163] dataset includes 12k labelled driving LiDAR frames from several traffic circumstances, as well as multimodal data from several cameras and LiDAR sensors. It makes a substantial addition to the creation of SDV for India. Its focus on capturing the complexities of Indian roads provides valuable data for researchers and developers working on this technology.

Summary of different dataset utilized for the development of SDV presented in Table 9. At the moment, datasets play a critical part in the process of exercise and validating techniques for SDVs, thereby creating the important groundwork that is required for the application of these methods and the evolution of technology with regard to SDVs.

### B. SIMULATION AND DEPLOYMENT FRAMEWORKS

A Simulation and Deployment Framework for SDVs motion planning provides a comprehensive platform for developing, testing, and deploying motion planning algorithms in both simulated and real-world scenarios. It enables researchers and developers to create and validate motion planning algorithms in simulated environments before deploying them onto actual

**TABLE 9. Summary of different dataset related to SDVs.**

| Dataset | Year | Sensors | Environment | Size | File Formats | Primary Tasks |
|---|---|---|---|---|---|---|
| A2D2 [146] | 2020 | Cameras, LiDAR, GPU, IMU | Diverse (urban, rural) | 1.3 million images | Images, LiDAR point clouds, GPS data, IMU data annotations | Object detection, tracking, semantic segmentation |
| ApolloScape [147] | 2019 | Cameras, LiDAR, GPS, IMU | Urban | 146k images, 1000 km trajectories | Images, LiDAR point clouds, GPS data, IMU data annotations | Object detection, tracking, lane detection, semantic segmentation, path planning |
| Argoverse 1 [148] | 2019 | Cameras, LiDAR | Urban (Pittsburgh, Miami) | 300k Trajectories | Images, LiDAR point clouds, camera calibration, annotations | Object detection, tracking, motion forecasting |
| BDD100K [149] | 2020 | Cameras, GPS, IMU | Urban | 100,000 images of 12M of data | Images, GPS data, IMU data and annotations | Object detection, traffic light detection, driving behaviour analysis |
| Cityscapes [150] | 2016 | Cameras | Urban (50 German cities) | 5,000 fine annotations, 20,000 coarse annotations | Images, annotations | Semantic segmentation, scene understanding |
| Comma.ai [151] | 2016 | Cameras, LiDAR, GPS, IMU | Various | 112,000 frames | Videos, LiDAR point clouds, GPS data, IMU data (limited access) | Object detection, lane detection, behaviour cloning (limited access) |
| KITTI [152] | 2013 | Cameras, LiDAR | Urban | 41k frames | Images, LiDAR point clouds, calibration data, annotations | Object detection, tracking, stereo vision |
| Lyft Level 5 [153] | 2021 | Cameras, LiDAR, radar, IMU, GPS | Urban | 1.1K hours of data (limited access for research) | Images, LiDAR point clouds, radar data, IMU data, GPS data, annotations | Object detection, tracking, lane detection, behaviour prediction (limited access) |
| NuScenes [154] | 2019 | Cameras, LiDAR, radar, | Urban | 1,000 drives 40k of data | Images, LiDAR point clouds, radar data, annotations | Object detection, tracking, , motion forecasting, scene understanding |
| Waymo Open Dataset [155] | 2019 | Cameras, LiDAR, radar | Urban | 1,500 hours, 230k of data | Videos, LiDAR point clouds, annotations (limited access) | Object detection, tracking, , motion forecasting (limited access) |
| highD [156] | 2018 | Cameras, LiDAR | Urban | 45K km distance (limited access for research) | Images, LiDAR point clouds, annotations | Object detection, tracking, lane detection, behaviour prediction (limited access) |
| INTERACTION [157] | 2021 | Cameras, LiDAR | Highway | 48 hours of data 110k Trajectories | Videos, LiDAR point clouds, annotations | Object detection, tracking, lane detection, behaviour analysis on highways |
| Argoverse 2 [158] | 2023 | Cameras, LiDAR | Urban (Miami) | 6 million frames | Images, LiDAR point clouds, camera calibration, annotations | 2D and 3D Object detection, tracking |
| Talk2BEV [159] | 2023 | QA pairs | Urban | 20,000 diverse question categories | large vision-language models with BEV maps | understanding of maneuver scenarios |
| IDD-3D [160] | 2023 | Cameras, LiDAR | Urban (India) | 12,000 LIDAR frames | LiDAR point clouds, annotations | 3D object detection and tracking tasks with different traffic condition |

SDVs, thereby accelerating the development process and ensuring robust performance in real-world conditions. In this section we are discussed about different simulator and their features related to motion planning for SDVs

As a result of the development of open-source SDV simulation platforms, algorithm testing in this field has become much easier. For example, the German Aerospace Centre is responsible for developing the SUMO [164] platform, which is a platform for simulating traffic at a tiny scale. In addition to providing comprehensive evaluation capability for huge-scale mobility methods, SUMO also includes an intuitive user interface that is compatible with a variety of data forms. There is widespread recognition for the extensive features that SUMO possesses, and it has become an essential component in simulation initiatives. In addition, LGSVL Simulator [165] is an open-source gem for SDV developers. This high-fidelity simulator creates realistic environments to test SDV algorithms. It integrates with popular frameworks like Autoware [166] and Apollo [167], saving you time on code setup. LGSVL doesn't stop there - you can customize sensors, design new objects, and even build digital replicas of real-world roads. With its focus on realism and ease of use, LGSVL Simulator integrated with [166] or [167] is a high powerful tool for

**TABLE 10.** Summary of various sumulation and deployment frameworks.

| Features\ Platform | Focus | Open Source | Sensor Simulation | Weather | Traffic Simulation | Scenario Design | Visualization | Strengths | Weaknesses |
|---|---|---|---|---|---|---|---|---|---|
| CARLA [168] | Urban environments | Yes | LiDAR, Camera, Radar, GPS, IMU | Rain, Snow, Fog | Vehicles, Pedestrians, Cyclists | Python API | 3D viewer, Sensor views | Open-source, realistic graphics, diverse sensors | Limited weather & traffic control |
| LGSVL [165] | Urban environments | No | LiDAR, Camera, Radar | Rain, Snow, Fog | Vehicles, Pedestrians, Cyclists | Python API | 3D viewer, Sensor views | High fidelity, real-time simulation | Commercial license |
| Baidu Apollo [167] | Urban & Highway | Partially | LiDAR, Camera, Radar | Rain, Snow, Fog | Vehicles, Pedestrians, Cyclists | Scenario editor, Python API | 3D viewer, Sensor views | Open-source integrates with Baidu hardware | Less open-source, limited to Baidu tools |
| Mcity | Urban & Highway | No | LiDAR, Camera, Radar | Limited | High-fidelity traffic | Scenario editor | 3D viewer | High-fidelity traffic simulation | Relies on University access |
| AirSim [171] | Varied (incl. urban) | Yes | LiDAR, Camera, Depth | Rain, Snow, Fog | Vehicles, Pedestrians | Python API | 3D viewer, First-person view | Integrates with Microsoft Azure | Primarily for research |
| rFpro[170] [37] | Racing simulator (adaptable) | No | Limited sensor options | Limited weather options | Limited traffic options | Scripting languages | 3D viewer | Realistic racing physics | Limited to racing scenarios |
| Autoware [166] | Open-source, ROS-based | Yes | LiDAR, Camera, Radar (ROS integration) | Rain, Snow, Fog (ROS integratio n) | Vehicles, Pedestrians, Cyclists (ROS integration) | ROS tools, Python | RViz (ROS visualization tool) | Open-source, ROS-based, active community | require some ROS expertise |
| SUMO [164] | Traffic simulation (macroscopic) | Yes | None (traffic simulation) | Limited | High-fidelity traffic (macroscopic) | Scripting languages | 2D viewer (macroscopic) | Large-scale traffic modeling & simulation | Not suitable for sensor-based control |
| TORCS [169] | Racing simulator (microscopic) | Yes | None | Limited | No traffic simulation | Track files, scripting | 3D viewer (microscopic) | Real-time racing environment | Not suitable for sensor-based control |

working on the future of SDV development in real-world environment.

Additionally, CARLA [168] is a crucial tool for ego-vehicle Self-driving. It's an open-source simulator focused on urban scenarios, facilitating development, training, and validation of core SDV components. With its realistic virtual environment, developers can test algorithms under various conditions. Its open-source nature fosters collaboration and innovation, accelerating progress in SDV technologies.

Further, TORCS [169] and rFpro [170] are leading simulators for multi-vehicle interaction studies. With 50+ vehicle models and 20+ tracks, they offer rich environments for research. Their ability to simulate races with up to 50 vehicles simultaneously provides invaluable insights into scalability and behaviour. Specially [169] Open-source nature fosters collaboration and customization, advancing research in SDVs.

Furthermore, AirSim [171] is a high-fidelity simulator developed by Microsoft for aerial and ground vehicles. It offers realistic environments, sensor simulation, physics-based dynamics, and open-source customization. AirSim [171], [164] enables developers to test and validate SDVs, accelerating research in robotics and artificial intelligence.

The summary of various simulation and deployment framework for SDV shown in table 10.

## V. PERFORMANCE EVALUATION ANALYSIS OF DL- BASED MOTION PLANNING AND E2EL TECHNIQUE FOR SDVs

This section compares the effectiveness of several DL techniques for tasks that are related to the motion planning of SDVs. This section is separated into four parts that deal with various DL-based behaviour planning, DL-based trajectory planning, DL-based E2EL and types of implementation strategies. Performance assessments in behaviour planning are evaluated based on commonly employed measures such as prediction accuracy, Recall, Precision and F1-Score. Performance assessments in trajectory planning are compared using commonly employed metrics, such as the Average Distance Error (ADE), Final Distance Error (FDE), Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) concerning the common horizon forecast time. Additionally, performance assessments in the DL-based E2EL are compared using commonly employed metrics Success Rate (SR) and RMSE. Further compared and analysed different types of implementation techniques. In this section, we utilized many equations which are adopted from [20], [34], [43], and [53].

### A. PEFORMANCE EVALUATION ANALYSIS OF BEHAVIOUR PLANNING METHODS

A key step in assessing the success of machine learning models is to evaluate their performance according to behaviour prediction accuracy, Precision, Recall and F1-Score. These metrics are employed to assess how well a model predicts a system's behaviour. Firstly, a model's ability to accurately anticipate a system's behaviour is measured by its behaviour prediction accuracy. It is defined as the proportion of accurately predicted outcomes to all forecasts. The formula for behaviour prediction accuracy is referred in (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where TP stands for "True Positive" (the number of correctly predicted positive outcomes), TN for "True Negative"(the number of correctly predicted negative outcomes), FP for "False Positive"(the number of incorrectly predicted positive outcomes) and FN for "False Negative" (the number of incorrectly predicted negative outcomes). For instance, overall prediction accuracy is used to evaluate most behavior prediction strategies. Different proposed models to predict various behaviour of SDVs discussed in Section II-A. Based on the various aforementioned model results we compared various DL-based Techniques for behaviour prediction and their corresponding performance accuracy is shown in Figure 13.

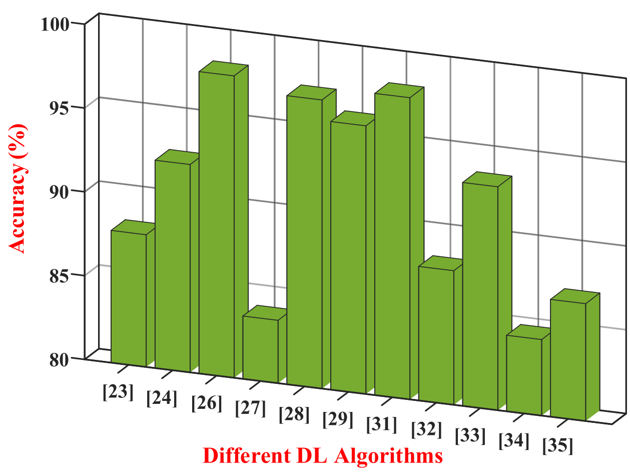


**FIGURE 13.** **Comparison of various behaviour planning algorithm's accuracy.**

The various discussed model accuracies are in the range of 80 % to 100 %. We considered the range of accuracy from 95% to 100%, 85% to 95% and 80% to 85% as excellent, moderate and least performance respectively. Excellently performed algorithms for a SDV in behavior prediction are AT_Mbi_LSTM [26], LSTM_CRF [31], LSTM [28] and LSTM_GRU [29] achieved higher accuracy of 98.01%, 98%, 97.22% and 96% respectively. Further AT_BiLSTM [33], FIS_LSTM [24], DBRNN [32], SNN [23] and DRNN [35] models achieved a moderate behaviour prediction accuracy of 93.33%, 92.4%, 88%, 87.89% and 87% respectively. Then, the least performing algorithms for a SDV for behaviour prediction are AT_GRU [34] and Multi_LSTM [27] achieved lower accuracy of 84.5% and 83.75% respectively. Hence, we observed the top most and least most performance accuracy among the early discussed algorithms for behaviour prediction of SDVs are achieved by AT_Mbi_LSTM [26] and Multi_LSTM [27] respectively. Overall, the selection of a DL algorithm for SDV behaviour prediction should be based on its accuracy and suitability for handling the specific driving scenarios encountered.

To provide a more accurate assessment of a model's performance, the F1-Score statistic combines precision and recall. It refers to the harmonic average of recall and precision. Refer (2) to calculate F1- Score.

$$F1 - Score = 2\frac{PrecisionRecall}{Precision + Recall} \tag{2}$$

F1- Scores of various discussed models are in the range of 80% to 100%. The different discussed models such as LSTM_CRF [31], AT_Mbi_LSTM [26], LSTM_GRU [29], AT_BiLSTM [33], Bi_LSTM [30] and AT_GRU [34] has Achieved F1- scores are 98.9%, 96.18%, 96%, 93%, 91.76% and 84.33% respectively. Hence, we observed top most and least most F1- scores among the early discussed algorithms for behaviour prediction of SDVs are achieved by LSTM_CRF [31] and AT_GRU [34] respectively. Hence, this comparison provides valuable insights into the performance of these algorithms and helps optimize the performance of SDVs for safety and reliability.

Next, the percentage of genuine positives among all correctly predicted positive outcomes is known as precision. In other words, it is the proportion of genuine positives to the total of both true and false positives. Equation 3 represents the precision formula.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

The different discussed models such as LSTM_CRF [31] [32], LSTM_GRU [29], [30], AT_Mbi_LSTM [26], [27], AT_BiLSTM [33], [34], Bi_LSTM [30], [31] and AT_GRU [34], [35] achieved precision are 99.5%, 96%, 94.47%, 94.17%, 86.67% and 85.07% respectively. We observed top most and least most precision among the early discussed algorithms for behaviour prediction of SDVs are achieved by LSTM_CRF [31], [32] and AT_GRU [34], [35] respectively.

Finally, the percentage of true positives that are actual genuine positives is known as recall. In other words, it is the proportion of genuine positives to the total of true positives and false negatives. Refer (4) to calculate the recall.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

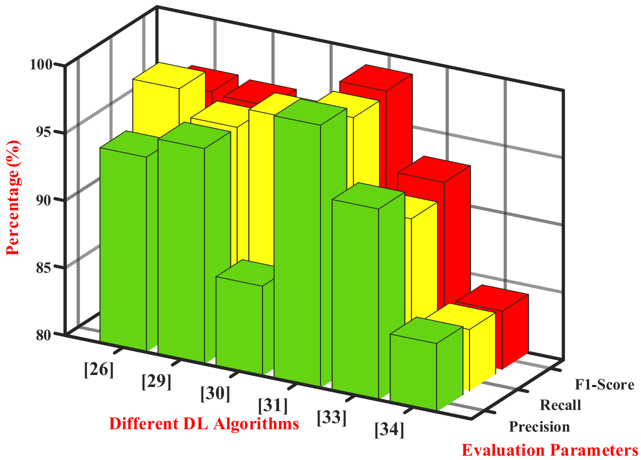


**FIGURE 14.** Comparison of various behaviour planning algorithms F1-Score, precision and recall.

The different discussed models such as LSTM_CRF [31], AT_Mbi_LSTM [26], Bi_LSTM [30], LSTM_GRU [29], AT_BiLSTM [33] and AT_GRU [34] has achieved recall percentages are 99.5%, 96%, 94.47%, 94.17%, 86.67% and 85.07% respectively. we observed top most and least most recall among the early discussed algorithms for behaviour prediction of SDVs are achieved by LSTM_CRF [31] and AT_GRU [34] respectively. These metrics provide a comprehensive evaluation of a model's performance in predicting the behaviour of a system. A comparison of various behaviour planning algorithms metrics Precision, Recall and F1- Score is shown in Figure 14.

### B. PERFORMANCE EVALUATION ANALYSIS OF TRAJECTORY PLANNING METHODS

Measuring the efficiency of DL approaches for forecasting the trajectory of SDV requires performance evaluation criteria. These measures evaluate how well the model forecasts the vehicle's position in the future based on its previous positions and movements. This section compares and analyses the four most often used performance evaluation metrics: MAE, FDE, ADE and RMSE. These metrics are represented in the unit of meter. Lower values for these metrics indicate better performance, and the criteria for each application will determine the appropriate metric.

To calculate the average deviation for each time step in the projected trajectory, the ADE compares the expected positions to the actual positions. The expression for ADE is referred in (5)

$$\mathrm{ADE} = \frac{1}{N}\sum\nolimits_{i=0}^{N} ||Pi - Pi^{*}|| \tag{5}$$

where N stands for the anticipated trajectory's number of time steps, Pi represents the vehicle's predicted position at time step i, and Pi* denotes the equivalent ground truth position.

Previously discussed DL algorithms for trajectory planning metrics ADE and FDE are compared in Figure 15. We considered prediction horizon 3s and 5s as a common factor to compare ADE and FDE among different DL-based trajectory planning for SDVs respectively. Previously discussed algorithms SafePathNet [37], P-LSTM-M-map [42], improved LaneGCN [45], Improved CNN [39] and U net (6 layers) [43] have achieved ADE values are 0.22, 0.51, 0.51, 0.565 and 0.6 in meter respectively.

The FDE calculates the separation between the vehicle's anticipated final location and its actual final position using ground truth data. The formula for FDE is referred in (6)

$$FDE = ||PN - PN^{*}|| \tag{6}$$

where PN* denotes the relevant ground truth final position and PN represents the vehicle's anticipated ultimate position. Previously discussed algorithms SafePathNet [37], U net (6 layers) [43], improved LaneGCN [45], P-LSTM-M-map [42] and Improved CNN [39] have achieved FDE values are 0.31, 0.795, 0.804, 0.996 and 1.03 in meter respectively. We observed that SafePathNet [37] ADE and FDE values are very less and better performance when compared with other models.

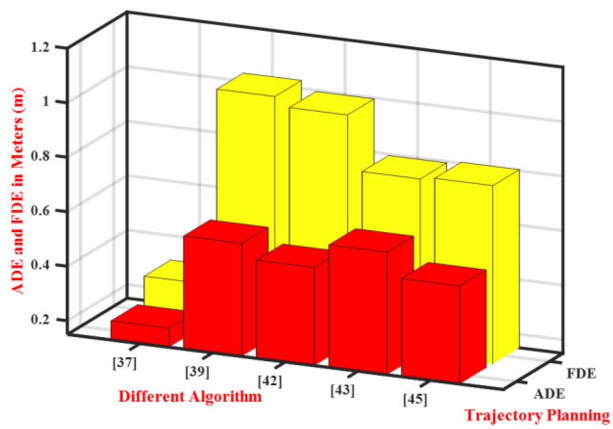


**FIGURE 15.** Comparison of various trajectory planning algorithms ADE and FDE values.

The average absolute difference between the anticipated positions and the ground truth positions for each time step in the expected trajectory is measured by the MAE. The formula for MAE is referred to in (7).

$$\mathrm{MAE} = \frac{1}{\mathrm{N}}\sum\nolimits_{\mathrm{i}=0}^{\mathrm{N}} ||\mathrm{Yi}-\mathrm{Yi}^{*}|| \tag{7}$$

where Yi is the anticipated position of the vehicle at time step i, Yi* is the equivalent ground truth position, and N is the number of time steps in the predicted trajectory. Previously discussed models CNN_Raw-RNN [38], Four layer LSTM [40] and U net (6 layer) [43] have achieved MAE values are 0.113, 0.29 and 0.38 in meters respectively. We observed that CNN_Raw-RNN [38] has a less MAE value when compared with other models which indicate that it better performed. Comparison of various trajectory planning algorithms MAE values are shown in Figure 16.

For each time step in the expected trajectory, the RMSE calculates the difference between the predicted positions and

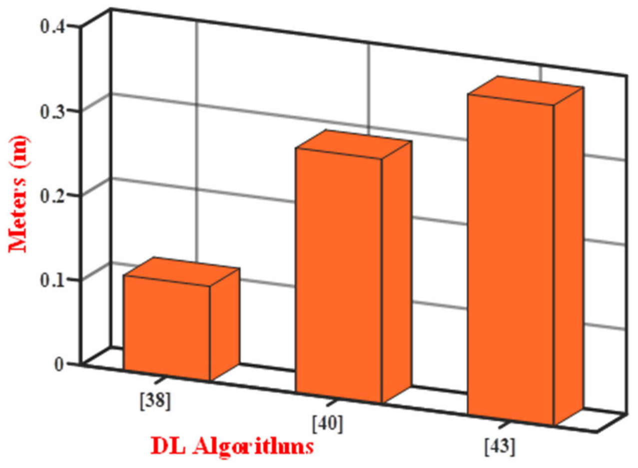


**FIGURE 16.** Comparison of various trajectory planning algorithms MAE values.

the actual positions. The formula for RMSE is expressed in Equation 8

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(Pi - Pi^*)^\wedge 2} \tag{8}$$

Some of the previously discussed models U net (6 layer) [43], AT_CNN_LSTM [41], NeuroTrajectory [50] and PF_CNN_LSTM [44] have achieved RMSE values are 1.23, 1.91, 2.09, and 4.26 in meter respectively. We observed that U net (6 layer) [43] has a less RMSE value when compared with other models. Comparison of various trajectory planning algorithms RMSE values are shown in Figure 17.

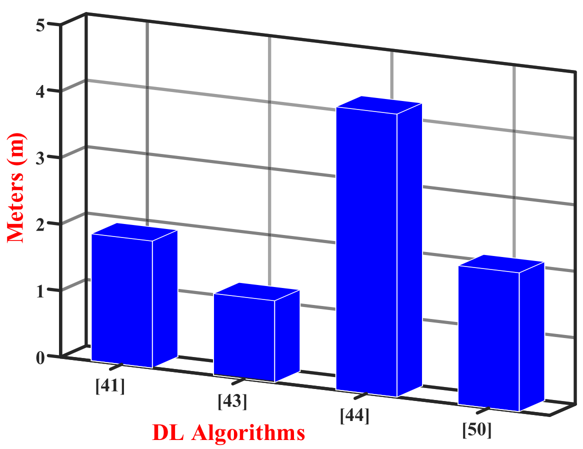


**FIGURE 17.** Comparison of various trajectory planning algorithms RMSE values.

Hence, the effectiveness of DL models for trajectory prediction of SDV is thoroughly compared and analyzed using ADE, FDE, MAE, and RMSE. These measurements support evaluating how well the model forecasts the vehicle's future assignments and how well it works in various traffic situations, such as in congested areas or inclement weather. Lower values for these metrics indicate better performance and the particular requirements of the application will determine which metric is best.

### C. PERFORMANCE EVALUATION ANALYSIS OF END-TO-END LEARNING METHODS

Metrics for performance evaluation are necessary to assess the efficacy of E2EL approaches for SDV. These measures evaluate how well the model predicts several aspects of driving, including steering angle, acceleration, and braking. In this context, this section compares and analyses Success Rate (SR) and RMSE, two widely used performance measurement metrics. The success Rate (SR)of an E2EL model for SDV is by evaluating its performance on a set of predefined tasks, including lane detection, object detection, and path planning. The success rate can be computed as the percentage of tasks completed correctly by the model.

Various discussed model Success Rate (SR) of E2EL for SDVs are Compared in Figure 18. It clearly shows that Modified Pointnet++ [69], MSF_SU [68], IVMP [67], Intention net [80] and Conditional Imitation Learning (CIL) [81] scored success rate is 93.6%, 91%, 88.67%, 75.28% and 60.72% respectively. We observed that Modified Pointnet++ [69] has achieved a high SR and CIL [81] achieved the least SR.

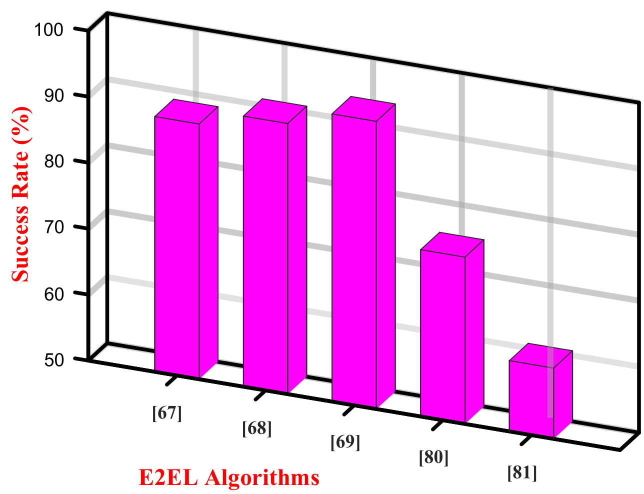


**FIGURE 18.** Comparison of different E2EL success rate.

In machine learning, the RMSE is a popular evaluation statistic for regression problems. RMSE can be used in the context of SDVs to assess how well the model predicts steering angles or other continuous variables. E2EL for SDV entails teaching a DNN to directly output a control signal, such as the steering angle, throttle, and brake, after receiving input from sensors like cameras and lidar.

By contrasting its outputs with the ground truth data, RMSE can be used to assess the network's predictions' accuracy. This statistic calculates how closely the actual values match the projected values, with a lower RMSE indicating better accuracy. By minimizing the RMSE loss during training, the network can be optimized to make more

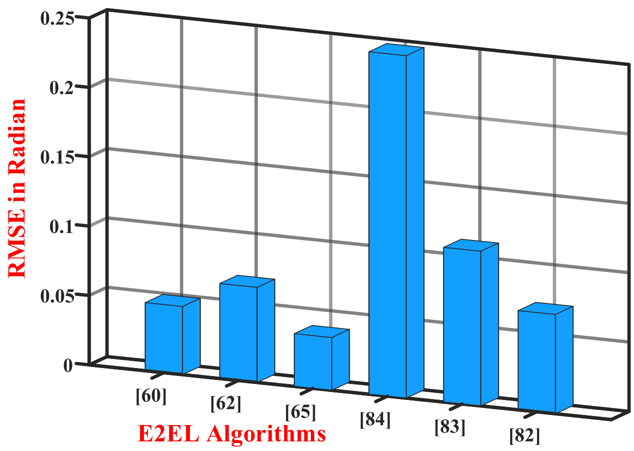


**FIGURE 19.** Comparison of E2EL technique RMSE value.

accurate predictions, which is essential for safe and reliable SDV.

Some of the previously discussed models of E2EL for SDV Two CNN [65], MSINet t+4 [60], Time to Line Cross (TLC) model [62], Deep Steering [82], HCA [83] and Cg23 [84] has achieved RMSE values are 0.038, 0.0491, 0.06849, 0.07153, 0.11145 and 0.24679 in radian respectively. We observed that Two CNN [65] has a less RMSE value when compared with other models. A comparison of various E2EL algorithm's RMSE values is shown in Figure 19.

## VI. IMPLEMENTATION ANALYSIS OF DL- BASED MOTION PLANNING AND E2EL TECHNIQUE FOR SDVs

The advent of SDVs represents a transformative shift in transportation, promising safer, more efficient, and convenient mobility solutions. However, realizing the full potential of self-driving requires effective implementation of sophisticated motion planning algorithms. Implementation analysis in this context involves assessing the practical deployment of these algorithms to ensure safe and efficient navigation in real-world environments. The outcomes of implementation analysis in this domain provide critical insights for refining motion planning algorithms, optimizing system performance, and enhancing the safety and reliability of SDVs. By systematically evaluating the practical deployment of motion planning algorithms, stakeholders can address challenges, identify opportunities for improvement, and accelerate the adoption of SDV technology.

Hence, we compared the way of implementation percentage between different categories of implementation. We grouped implementations based on their type. Groping of different reviewed papers depends on the above-mentioned category shown in Figure 20. In behaviour planning there are three types of implementations are identified, they are Hardware in Loop Simulation (HIL Simulation), Simulation with Numerical data Analysis (Simulation + NA) and Simulation with Real World Implementation (Simulation + RWI). In trajectory planning there are three types of implementations are identified, they are Simulation with Numerical data Analysis (Simulation + NA), Simulation with Numerical data Analysis with Real Time Implementation (Simulation + NA + RTI) and Simulation with Numerical data Analysis with Real-world Implementation (Simulation + NA + RWI). In E2EL there are three types of implementations are identified, they are Simulation, Simulation with Numerical data Analysis (Simulation + NA) and Simulation with Real World Implementation (Simulation + RWI).

### A. COMPARISON ANALYSIS OF DIFFERENT TYPES OF IMPLEMENTATIONS

In this section, we are comparing the several types of implementations for behaviour planning, trajectory planning, end to end planning and overall comparison between simulation and real-world implementation shown in Figure 4. From this survey, in behaviour planning three types of implementation groups were categorised, they are Simulation+NA, HIL Simulation and Simulation+RWI calculated percentages are 46.16, 38.46 and 15.38 respectively as shown in Figure 21(a). Besides this in trajectory planning another three types of implementation groups are categorised, they are Simulation+NA, Simulation+NA+RTI and Simulation+NA+RWI calculated percentages are 40, 26.66 and 33.34 respectively as shown in Figure 21(b). Besides this in E2EL, another three types of implementation groups are categorised, they are Simulation, Simulation+NA and Simulation+RWI calculated percentages are 38.89, 33.34 and 27.77 respectively as shown in Figure 21(c). Hence, the overall implementation percentage shown in Figure 21(d) which is compared between simulation and real-world implementation has calculated percentages are 71.74 and 28.26 respectively. We observed that most of the researcher implemented their work by software simulation and a smaller number of researchers implemented their work in the real world. The main reason behind this implementation in the real world needs more funds and a hardware approach which is a big challenge for the researcher.

Further, there are some other reasons why the implementation of various research is high in simulation but low in real-world implementation:

- Cost: It can be costly to conduct research in real-world settings, particularly when it involves massive experiments or intensive data collection. Simulations are often less costly because they can be running on computers and require fewer resources.
- Safety: Simulations can be used to evaluate hypotheses and conduct experiments without putting people or the environment at risk. This is crucial in the field of self-driving since mistakes might have severe consequences.
- Control: In simulations, researchers have a high degree of control over variables and conditions, which is often not possible in real-world settings. This allows for more precise and targeted experiments, leading to more reliable results.

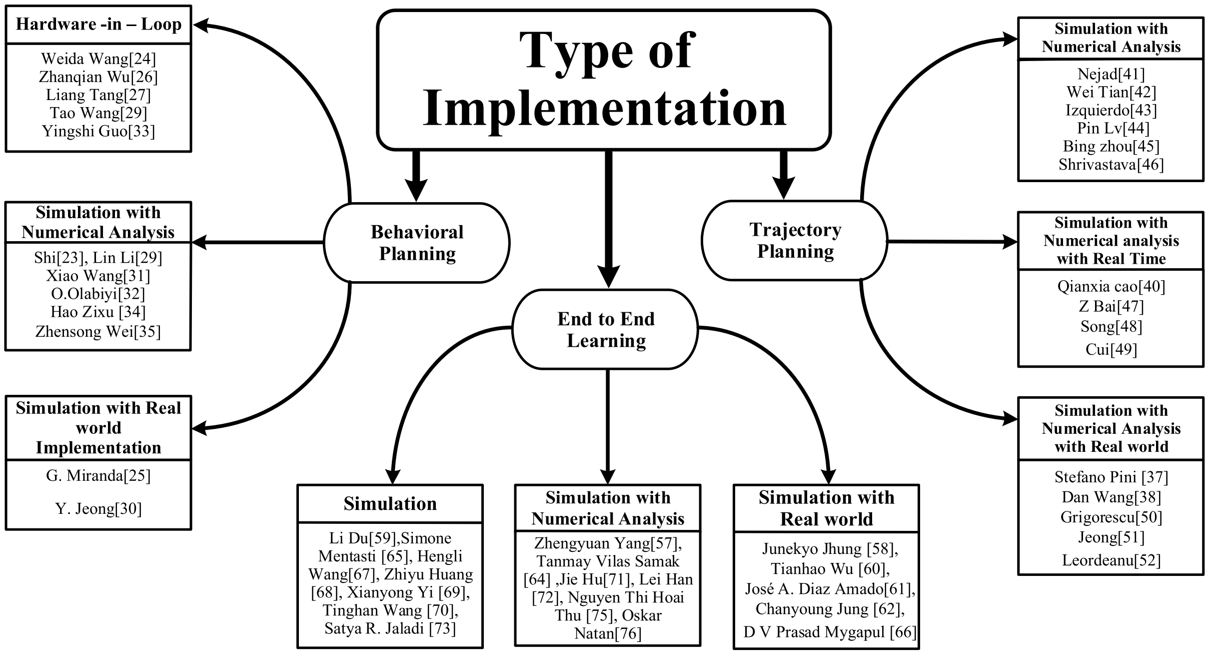


**FIGURE 20.** Type of implementation.

- Time: Conducting research in real-world environments can be time-consuming, especially when data collection requires long-term observation or follow-up. Simulations can be run faster, allowing researchers to evaluate and refine their hypotheses more quickly.
- Accessibility: Simulations can be accessed by researchers and scientists all over the world, making collaboration and sharing of results easier. This can lead to more diverse and robust research findings.

However, it is essential to remember that simulations don't always accurately reflect actual circumstances, and results obtained in simulations may not necessarily translate to the real world. Therefore, it is essential to validate simulation results with real-world experiments whenever possible.

Hence, the proficiency and effectiveness of the aforementioned motion planning and E2EL strategies indicate numerous difficulties. Although these approaches are relatively expensive to compute, they demonstrate promising outcomes for their intended task. Additionally, because they ignore crucial factors like energy consumption or forecast delay, mainstream approaches are only practical with cloud servers and high-end GPUs, which is an unrealistic situation for real application contexts. Further, we discuss several unresolved problems and their corresponding recommendations in the following section.

## VII. CHALLENGES AND FUTURE RECOMMENDATIONS

SDVs have advanced significantly, with validation on partially open roads in several cities. But there are challenges to full commercial implementation. Challenges include ensuring safety in diverse environments, navigating regulatory frameworks, building public trust, and upgrading infrastructure.

### A. CHALLENGES

Here are some of the important challenges in SDV development:

#### 1) HANDLING THE LONG TAIL OF RARE EVENTS

SDVs have shown remarkable proficiency in handling routine driving scenarios and well-maintained roads. However, the real world presents a myriad of unpredictable challenges. From sudden downpours to blowing debris, and even animals darting into traffic, rare and unexpected events continue to pose significant hurdles. While advancements in object recognition have enhanced the capabilities of SDVs AI, these uncommon occurrences can still befuddle the system. Additionally, navigating edge cases and ambiguous situations remains a formidable task. Traffic laws open to interpretation and human drivers relying on intuition to handle scenarios like unclear hand signals or merging lanes add layers of complexity. Teaching SDVs to emulate human judgment in such ambiguous circumstances remains an ongoing endeavour, underscoring the necessity for continued refinement and adaptation in SDV technology.

#### 2) DEPENDENCE ON HIGH-DEFINITION (HD) MAPS AND INFRASTRUCTURE

Driving SDVs rely on detailed, constantly updated HD maps to navigate. However, creating and maintaining these maps for every road everywhere is a massive undertaking. Additionally, poorly marked lanes, construction zones, or missing

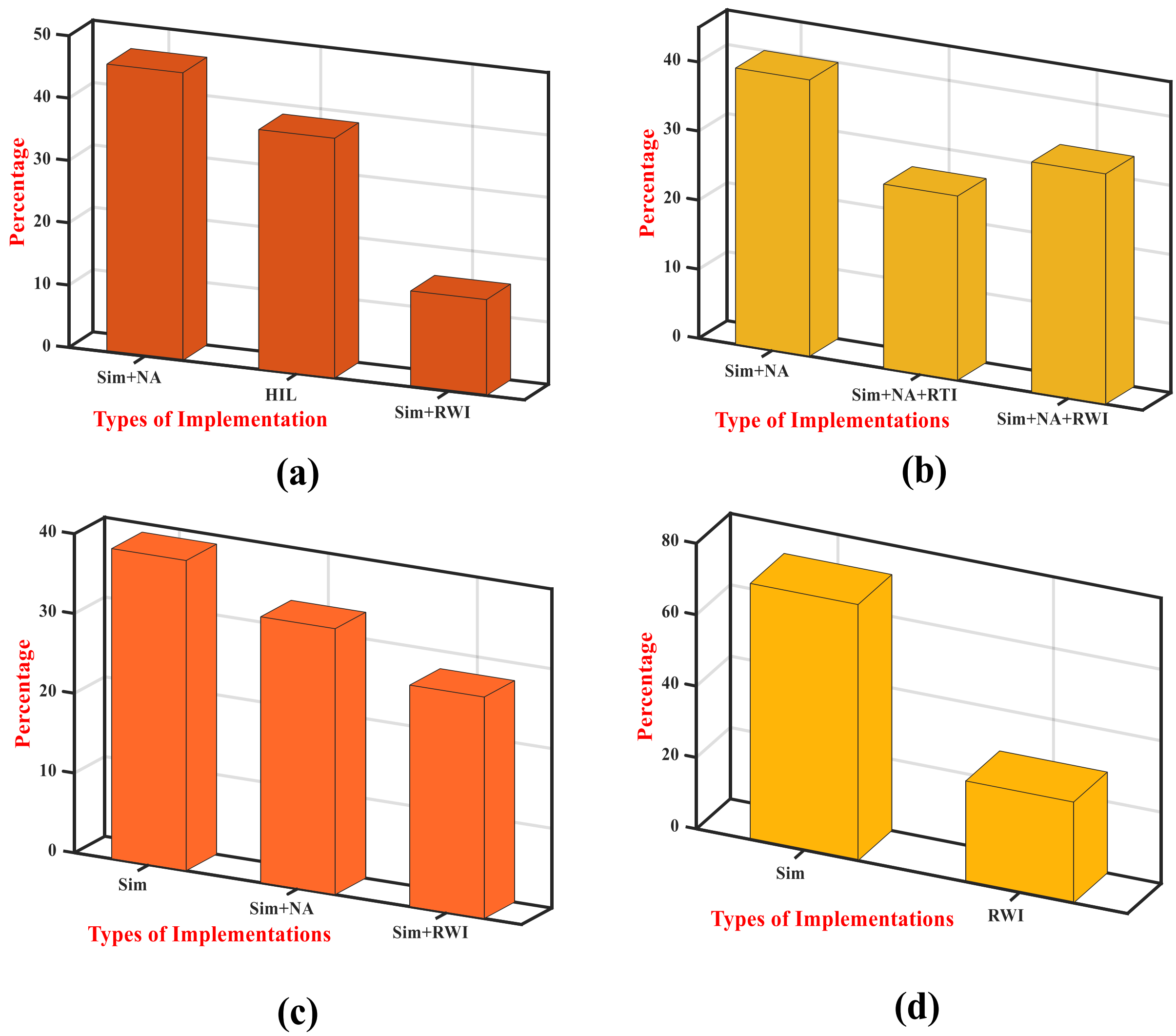


**FIGURE 21. Comparison of different types of implementation percentages among the survey papers. (a)Implementation percentage for behaviour Planning. (b)Implementation percentage for trajectory planning. (c) Implementation percentage for E2EL. (d) Overall percentage between simulation and Real-world Implementation. Simulation (Sim), Numerical Analysis (NA), Real-Time Implementation (RTI), Real World Implementation (RWI).**

signage can disrupt a SDV that relies too heavily on pre-programmed information.

### 3) SENSOR LIMITATIONS AND ADVERSE WEATHER

SDVs uses a complex suite of sensors to perceive their surroundings. However, these sensors can be fooled by things like fog, heavy rain, or even bright sunlight. Recent challenges include improving sensor performance in adverse weather conditions and ensuring they can't be easily confused by external factors.

### 4) CYBERSECURITY THREATS

SDVs are essentially computers on wheels, and like any computer system, they are vulnerable to hacking. A malicious actor could potentially take control of a SDV, causing accidents or privacy breaches. Ensuring robust cybersecurity measures are in place is crucial.

### 5) ETHICAL CONSIDERATIONS AND MORAL DILEMMAS

SDVs may someday face situations where an accident is unescapable. How the SDV is programmed to react in these ‘‘split-second’’ scenarios raises complex ethical questions. Engineers are grappling with how to program these vehicles to make the safest decisions possible, while considering factors like pedestrian safety and minimizing harm.

### 6) DATASET

High-quality datasets must be readily available in order to train and evaluate SDV algorithms. Although simulators are essential to this procedure, models that are exclusively trained in virtual environments frequently encounter difficulties when applied to real-world situations [172]. Thus, to effectively develop research in this subject, bridging the gap between simulated and realistic data is crucial.

## B. FUTURE RECOMMENDATIONS

Motion planning, the art of navigating a SDVS safely and efficiently, is on the cusp of significant advancements. Here's some of the future recommendation in the field of motion planning for SDVs.

### 1) INTERPRETABILITY: DEMYSTIFYING THE BLACK BOX

Currently, many AI models used for motion planning function as "black boxes." Their decision-making processes are opaque, making it difficult to understand why a car took a particular route or performed a specific maneuver. This lack of transparency is a major hurdle for gaining public trust and regulatory approval. The future of motion planning lies in **interpretable planning.** This means developing algorithms that can explain their reasoning in a way humans can understand. Imagine a system that could highlight factors like traffic flow, pedestrian presence, and signal interpretations when making a decision. This transparency will be crucial for building trust with regulators and the public, paving the way for wider adoption of SDV technology.

### 2) Sim2Real TRANSFER: BRIDGING THE SIMULATION GAP

A major challenge in developing SDV is the disparity between the controlled environment of simulations and the unpredictable nature of the real world [173], [174]. This gap can lead to situations where the SDV struggles to adapt its motion planning strategies when encountering unexpected obstacles or variations in road conditions. The future will see advancements in **Sim2Real transfer**. This involves developing algorithms that can effectively translate learnings from meticulously crafted simulations to the real world. Imagine a virtual environment that can realistically simulate not only ideal conditions but also diverse weather patterns, construction zones, and even erratic driver behavior. By training and validating motion planning algorithms in these nuanced simulations, developers can ensure a smoother and safer transition to real-world deployment.

### 3) DIGITAL TWIN INTEGRATION: A VIRTUAL PLAYGROUND FOR TESTING

virtual replicas of SDV and their environments will continue to play a vital role in refining motion planning algorithms. These virtual cities can be populated with millions of meticulously crafted scenarios, allowing researcher to test and refine the SDVs decision-making under a vast array of conditions. Imagine a digital twin simulating a busy intersection during rush hour with malfunctioning traffic lights and a jaywalking pedestrian. By testing motion planning algorithms in these complex situations, researcher can identify potential weaknesses and refine the SDVs ability to handle the unexpected, leading to a significant improvement in overall safety and performance.

### 4) RELIABILITY: BUILDING CONFIDENCE ON THE ENVIRONMENT

For SDVs to become a reliable mode of transportation, they need to demonstrate exceptional reliability. Current motion planning algorithms can sometimes struggle with unexpected situations like sudden sensor failures or previously unseen traffic patterns. The future will focus on developing **highly reliable motion planning algorithms**. This involves incorporating strategies for graceful degradation and fail-safe mechanisms. Imagine a SDV that can not only navigate flawlessly under normal conditions but also has backup plans or alternative routes in case of sensor malfunctions or unforeseen circumstances. Additionally, algorithms will be designed to continuously learn and adapt from real-world experiences, further enhancing their reliability over time.

### 5) GOVERNANCE: ESTABLISHING THE RULES OF THE ROAD

With the rise of SDVs, a clear and **comprehensive governance framework** will be essential. These frameworks will define how SDVs interact with human-driven vehicles and pedestrians, ensuring order and predictability on the roads. Imagine a set of regulations that govern communication protocols between SDVs, establish right-of-way rules in complex situations, and clearly define liability in case of accidents. Developing robust governance frameworks will be a collaborative effort between policymakers, developer, and industry leaders, fostering a safe and efficient transportation ecosystem.

# VIII. CONCLUSION

This review paper provides an extensive overview of innovative models in DL based motion planning and E2EL technologies within the field of SDV) It covers various performance metrics and challenges encountered in SDVs development. The primary approaches discussed include behavior planning, trajectory planning, and E2E learning (Imitation Learning and Reinforcement Learning). Each approach's state-of-the-art model is presented and compared. The survey also highlights the importance of practical enablers such as datasets and simulation deployment frameworks, along with their comparisons and reviews. Additionally, the survey offers insights into the implementation of state-of-the-art techniques and compares common performance metrics across behaviour planning, trajectory planning, and E2E learning for SDV. It analyses the distribution of different implementation types among reviewed papers. Furthermore, the survey identifies ongoing challenges in the development of SDV for real-world environment and provides future recommendations for addressing these challenges.

# REFERENCES

[1] A. Bachani, M. Peden, G. Gururaj, R. Norton, and A. Hyder, "Road traffic injuries," Tech. Rep., 2017.
[2] L. Chen, Y. Li, C. Huang, B. Li, Y. Xing, D. Tian, L. Li, Z. Hu, X. Na, Z. Li, S. Teng, C. Lv, J. Wang, D. Cao, N. Zheng, and F.-Y. Wang, "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 2, pp. 1046–1056, Feb. 2023.

[3] W. Wang, L. Wang, C. Zhang, C. Liu, and L. Sun, "Social interactions for autonomous driving: A review and perspectives," *Found. Trends Robot.*, vol. 10, nos. 3–4, pp. 198–376, 2022, doi: 10.1561/2300000078.

[4] L. Chen, Y. Zhang, B. Tian, D. Cao, and F.-Y. Wang, "Parallel driving os: Aubiquitous cyber-physical-socialsystem-based operating system for autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 4, pp. 801–803, Dec. 2022.

[5] R. Song, Y. Ai, B. Tian, L. Chen, F. Zhu, and F. Yao, "MSFANet: A light weight object detector based on context aggregation and attention mechanism for autonomous mining truck," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 3, pp. 2285–2295, Mar. 2023.

[6] L. Gong, Y. Wu, B. Gao, Y. Sun, X. Le, and C. Liu, "Real-time dynamic planning and tracking control of auto-docking for efficient wireless charging," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 3, pp. 2123–2134, Mar. 2023.

[7] J. Ziegler et al., "Making Bertha drive—An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Summer 2014.

[8] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[9] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mech. Syst. Signal Process.*, vol. 100, pp. 482–500, Feb. 2018.

[10] W. Zhang, W. Wang, J. Zhu, and D. Zhao, "Multi-vehicle interaction scenarios generation with interpretable traffic primitives and Gaussian process regression," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Las Vegas, NV, USA, Oct. 2020, pp. 1197–1204, doi: 10.1109/IV47402.2020.9304568.

[11] W. Zhang and W. Wang, "Learning V2V interactive driving patterns at signalized intersections," *Transp. Res. C, Emerg. Technol.*, vol. 108, pp. 151–166, Nov. 2019.

[12] J. Chen, C. Liu, and M. Tomizuka, "FOAD: Fast optimization-based autonomous driving motion planner," in *Proc. Annu. Amer. Control Conf. (ACC)*, Milwaukee, WI, USA, Jun. 2018, pp. 4725–4732, doi: 10.23919/ACC.2018.8431104.

[13] H. Zhou, J. Laval, A. Zhou, Y. Wang, W. Wu, Z. Qing, and S. Peeta, "Review of learning-based longitudinal motion planning for autonomous vehicles: Research gaps between self-driving and traffic congestion," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2676, no. 1, pp. 324–341, Jan. 2022, doi: 10.1177/03611981211035764.

[14] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826–1848, May 2020, doi: 10.1109/TITS.2019.2913998.

[15] K. Muhammad, A. Ullah, J. Lloret, J. D. Ser, and V. H. C. de Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4316–4336, Jul. 2021, doi: 10.1109/TITS.2020.3032227.

[16] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, Feb. 2022, doi: 10.1109/TITS.2020.3024655.

[17] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robot.*, vol. 37, no. 3, pp. 362–386, Apr. 2020.

[18] F. Ye, S. Zhang, P. Wang, and C.-Y. Chan, "A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Nagoya, Japan, Jul. 2021, pp. 1073–1080, doi: 10.1109/IV48863.2021.9575880.

[19] B. B. Elallid, N. Benamar, A. S. Hafid, T. Rachidi, and N. Mrani, "A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7366–7390, Oct. 2022.

[20] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu, and L. Chen, "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Trans. Intell. Vehicles*, pp. 1–21, Jun. 2023, doi: 10.1109/TIV.2023.3274536.

[21] P. S. Chib and P. Singh, "Recent advancements in end-to-end autonomous driving using deep learning: A survey," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 103–118, Jan. 2024, doi: 10.1109/TIV.2023.3318070.

[22] N. Wang, X. Li, K. Zhang, J. Wang, and D. Xie, "A survey on path planning for autonomous ground vehicles in unstructured environments," *Machines*, vol. 12, no. 1, p. 31, Jan. 2024, doi: 10.3390/machines12010031.

[23] K. Shi, Y. Wu, H. Shi, Y. Zhou, and B. Ran, "An integrated car-following and lane changing vehicle trajectory prediction algorithm based on a deep neural network," *Phys. A, Stat. Mech. Appl.*, vol. 599, Aug. 2022, Art. no. 127303.

[24] W. Wang, T. Qie, C. Yang, W. Liu, C. Xiang, and K. Huang, "An intelligent lane-changing behavior prediction and decision-making strategy for an autonomous vehicle," *IEEE Trans. Ind. Electron.*, vol. 69, no. 3, pp. 2927–2937, Mar. 2022.

[25] O. González-Miranda and J. M. Ibarra-Zannatha, "Behavior selector for autonomous vehicles using neural networks," in *Proc. 24th Robot. Mex. Congr. (COMRob)*, Nov. 2022, pp. 31–35.

[26] Z. Wu, K. Liang, D. Liu, and Z. Zhao, "Driver lane change intention recognition based on attention enhanced residual-MBi-LSTM network," *IEEE Access*, vol. 10, pp. 58050–58061, 2022.

[27] L. Tang, H. Wang, W. Zhang, Z. Mei, and L. Li, "Driver lane change intention recognition of intelligent vehicle based on long short-term memory network," *IEEE Access*, vol. 8, pp. 136898–136905, 2020.

[28] T. Wang, Z. Guan, T. Peng, D. Li, and R. Zhao, "Human-like decision making for autonomous lane changing using deep learning," in *Proc. IEEE 2nd Int. Conf. Inf. Technol., Big Data Artif. Intell. (ICIBA)*, vol. 2, Dec. 2021, pp. 950–955.

[29] L. Li, W. Zhao, C. Xu, C. Wang, Q. Chen, and S. Dai, "Lane-change intention inference based on RNN for autonomous driving on highways," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5499–5510, Jun. 2021.

[30] Y. Jeong, "Predictive lane change decision making using bidirectional long shot-term memory for autonomous driving on highways," *IEEE Access*, vol. 9, pp. 144985–144998, 2021.

[31] X. Wang, J. Wu, Y. Gu, H. Sun, L. Xu, S. Kamijo, and N. Zheng, "Human-like maneuver decision using LSTM-CRF model for on-road self-driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 210–216.

[32] O. Olabiyi, E. Martinson, V. Chintalapudi, and R. Guo, "Driver action prediction using deep (bidirectional) recurrent neural network," 2017, *arXiv:1706.02257*.

[33] Y. Guo, H. Zhang, C. Wang, Q. Sun, and W. Li, "Driver lane change intention recognition in the connected environment," *Phys. A, Stat. Mech. Appl.*, vol. 575, Aug. 2021, Art. no. 126057.

[34] Z. Hao, X. Huang, K. Wang, M. Cui, and Y. Tian, "Attention-based GRU for driver intention recognition and vehicle trajectory prediction," in *Proc. 4th CAA Int. Conf. Veh. Control Intell. (CVCI)*, Dec. 2020, pp. 86–91.

[35] Z. Wei, C. Wang, P. Hao, and M. J. Barth, "Vision-based lane-changing behavior detection using deep residual neural network," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3108–3113.

[36] M. R. Bachute and J. M. Subhedar, "Autonomous driving architectures: Insights of machine learning and deep learning algorithms," *Mach. Learn. Appl.*, vol. 6, Dec. 2021, Art. no. 100164.

[37] S. Pini, C. S. Perone, A. Ahuja, A. S. R. Ferreira, M. Niendorf, and S. Zagoruyko, "Safe real-world autonomous driving by learning to predict and plan with a mixture of experts," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 10069–10075.

[38] D. Wang, C. Wang, Y. Wang, H. Wang, and F. Pei, "An autonomous driving approach based on trajectory learning using deep neural networks," *Int. J. Automot. Technol.*, vol. 22, no. 6, pp. 1517–1528, Dec. 2021.

[39] T. Chen, C. Guo, H. Li, T. Gao, L. Chen, H. Tu, and J. Yang, "An improved multimodal trajectory prediction method based on deep inverse reinforcement learning," *Electronics*, vol. 11, no. 24, p. 4097, Dec. 2022.

[40] Q. Cao, Z. Zhao, Q. Zeng, Z. Wang, and K. Long, "Real-time vehicle trajectory prediction for traffic conflict detection at unsignalized intersections," *J. Adv. Transp.*, vol. 2021, pp. 1–15, Dec. 2021.

[41] Z. Salahshoori Nejad, H. Heravi, A. Rahimpour Jounghani, A. Shahrezaie, and A. Ebrahimi, "Vehicle trajectory prediction in top-view image sequences based on deep learning method," 2021, *arXiv:2102.01749*.

[42] W. Tian, S. Wang, Z. Wang, M. Wu, S. Zhou, and X. Bi, "Multi-modal vehicle trajectory prediction by collaborative learning of lane orientation, vehicle interaction, and intention," *Sensors*, vol. 22, no. 11, p. 4295, Jun. 2022.

[43] R. Izquierdo, Á. Quintanar, D. F. Llorca, I. G. Daza, N. Hernández, I. Parra, and M. Á. Sotelo, "Vehicle trajectory prediction on highways using bird eye view representations and deep learning," *Appl. Intell.*, vol. 53, no. 7, pp. 8370–8388, Apr. 2023.

[44] P. Lv, H. Liu, J. Xu, and T. Li, "Trajectory prediction with correction mechanism for connected and autonomous vehicles," *Electronics*, vol. 11, no. 14, p. 2149, Jul. 2022.

[45] B. Zhou, J. Zou, X. Wu, T. Chai, R. Zhou, Q. Pan, and R. Zhou, "Research on the improvement of the LaneGCN trajectory prediction algorithm," *Transp. Saf. Environ.*, vol. 4, no. 4, Oct. 2022, Art. no. tdac034.

[46] A. Shrivastava, J. P. V. Verma, S. Jain, and S. Garg, "A deep learning based approach for trajectory estimation using geographically clustered data," *Social Netw. Appl. Sci.*, vol. 3, no. 6, p. 597, Jun. 2021.

[47] Z. Bai, B. Cai, W. ShangGuan, and L. Chai, "Deep learning based motion planning for autonomous vehicle using spatiotemporal LSTM network," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2018, pp. 1610–1614.

[48] S. Song, X. Hu, J. Yu, L. Bai, and L. Chen, "Learning a deep motion planning model for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1137–1142.

[49] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2090–2096.

[50] S. M. Grigorescu, B. Trasnea, L. Marina, A. Vasilcoi, and T. Cocias, "NeuroTrajectory: A neuroevolutionary approach to local state trajectory learning for autonomous vehicles," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3441–3448, Oct. 2019.

[51] Y. Jeong, S. Kim, and K. Yi, "Surround vehicle motion prediction using LSTM-RNN for motion planning of autonomous vehicles at multi-lane turn intersections," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 2–14, 2020.

[52] M. Leordeanu and I. Paraicu, "Driven by vision: Learning navigation by visual localization and trajectory prediction," *Sensors*, vol. 21, no. 3, p. 852, Jan. 2021.

[53] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. M´uller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*.

[54] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 1, no. 1, pp. 187–210, 2018.

[55] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1364–1384, Apr. 2022.

[56] Y. Kang, H. Yin, and C. Berger, "Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 2, pp. 171–185, Jun. 2019.

[57] Z. Yang, Y. Zhang, J. Yu, J. Cai, and J. Luo, "End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2289–2294.

[58] J. Jhung, I. Bae, J. Moon, T. Kim, J. Kim, and S. Kim, "End-to-end steering controller with CNN-based closed-loop feedback for autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 617–622.

[59] L. Du, K. Ji, Z. Zhao, F. Su, and B. Zhuang, "An end-to-end future frame prediction method for vehicle-centric driving videos," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2019, pp. 1–4.

[60] T. Wu, A. Luo, R. Huang, H. Cheng, and Y. Zhao, "End-to-end driving model for steering control of autonomous vehicles with future spatiotemporal features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 950–955.

[61] J. A. Diaz Amado, I. P. Gomes, J. Amaro, D. F. Wolf, and F. S. Osório, "End-to-end deep learning applied in autonomous navigation using multi-cameras system with RGB and depth images," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1626–1631.

[62] C. Jung, H. Seong, and D. H. Shim, "Time-to-line crossing enhanced end-to-end autonomous driving framework," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–7.

[63] M.-J. Lee and Y.-G. Ha, "Autonomous driving control using end-to-end deep learning," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2020, pp. 470–473.

[64] T. Vilas Samak, C. Vilas Samak, and S. Kandhasamy, "Robust behavioral cloning for autonomous vehicles using end-to-end imitation learning," 2020, *arXiv:2010.04767*.

[65] S. Mentasti, M. Bersani, M. Matteucci, and F. Cheli, "Multi-state end-to-end learning for autonomous vehicle lateral control," in *Proc. AEIT Int. Conf. Electr. Electron. Technol. Automot.*, Nov. 2020, pp. 1–10.

[66] D. V. P. Mygapula, "CNN based end to end learning steering angle prediction for autonomous electric vehicle," in *Proc. 4th Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Sep. 2021, pp. 1–6.

[67] H. Wang, P. Cai, Y. Sun, L. Wang, and M. Liu, "Learning interpretable end-to-end vision-based motion planning for autonomous driving with optical flow distillation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13731–13737.

[68] Z. Huang, C. Lv, Y. Xing, and J. Wu, "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding," *IEEE Sensors J.*, vol. 21, no. 10, pp. 11781–11790, May 2021.

[69] X. Yi, H. Ghazzai, and Y. Massoud, "End-to-end neural network for autonomous steering using LiDAR point cloud data," in *Proc. IEEE 65th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2022, pp. 1–4.

[70] T. Wang, Y. Luo, J. Liu, R. Chen, and K. Li, "End-to-end self-driving approach independent of irrelevant roadside objects with auto-encoder," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 641–650, Jan. 2022.

[71] J. Hu, H. Kong, Q. Zhang, and R. Liu, "Enhancing scene understanding based on deep learning for end-to-end autonomous driving," *Eng. Appl. Artif. Intell.*, vol. 116, Nov. 2022, Art. no. 105474.

[72] L. Han, L. Wu, F. Liang, H. Cao, D. Luo, Z. Zhang, and Z. Hua, "A novel end-to-end model for steering behavior prediction of autonomous ego-vehicles using spatial and temporal attention mechanism," *Neurocomputing*, vol. 490, pp. 295–311, Jun. 2022.

[73] S. R. Jaladi, Z. Chen, N. R. Malayanur, R. M. Macherla, and B. Li, "End-to-end training and testing gamification framework to learn human highway driving," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 4296–4301.

[74] J. Kwon, A. Khalil, D. Kim, and H. Nam, "Incremental end-to-end learning for lateral control in autonomous driving," *IEEE Access*, vol. 10, pp. 33771–33786, 2022.

[75] N. T. Hoai Thu and D. Seog Han, "An end-to-end motion planner using sensor fusion for autonomous driving," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2023, pp. 678–683.

[76] O. Natan and J. Miura, "End-to-end autonomous driving with semantic depth cloud mapping and multi-agent," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 557–571, Jan. 2023.

[77] Z. Zhu and H. Zhao, "Learning autonomous control policy for intersection navigation with pedestrian interaction," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 5, pp. 3270–3284, May 2023.

[78] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4693–4700.

[79] Q. Wang, L. Chen, B. Tian, W. Tian, L. Li, and D. Cao, "End-to-end autonomous driving: An angle branched network approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11599–11610, Dec. 2019.

[80] M. Peng, Z. Gong, C. Sun, L. Chen, and D. Cao, "Imitative reinforcement learning fusing vision and pure pursuit for self-driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3298–3304.

[81] S. Teng, L. Chen, Y. Ai, Y. Zhou, Z. Xuanyuan, and X. Hu, "Hierarchical interpretable imitation learning for end-to-end autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 673–683, Jan. 2023.

[82] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2722–2730.

[83] A. Sauer, N. Savinov, and A. Geiger, "Conditional affordance learning for driving in urban environments," in *Proc. Conf. Robot Learn.*, 2018, pp. 237–252.

[84] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8652–8661.

[85] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, "Perceive, predict, and plan: Safe motion planning through interpretable semantic representations," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 414–430.

[86] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist. JMLR Workshop Conf.*, 2011, pp. 627–635.

[87] A. Attia and S. Dayan, "Global overview of imitation learning," 2018, *arXiv:1801.06503*.

[88] H. He, J. Eisner, and H. Daume, "Imitation learning by coaching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1–9.

[89] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end autonomous driving," 2016, *arXiv:1605.06450*.

[90] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg, "ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning," 2021, *arXiv:2109.08273*.

[91] C. Yan, J. Qin, Q. Liu, Q. Ma, and Y. Kang, "Mapless navigation with safety-enhanced imitation learning," *IEEE Trans. Ind. Electron.*, vol. 70, no. 7, pp. 7073–7081, Jul. 2023.

[92] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 663–670.

[93] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 1–8.

[94] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1–8.

[95] T. Phan-Minh, F. Howington, T.-S. Chu, S. Uk Lee, M. S. Tomov, N. Li, C. Dicle, S. Findler, F. Suarez-Ruiz, R. Beaudoin, B. Yang, S. Omari, and E. M. Wolff, "Driving in real life with inverse reinforcement learning," 2022, *arXiv:2206.03004*.

[96] M. Valko, M. Ghavamzadeh, and A. Lazaric, "Semi-supervised apprenticeship learning," in *Proc. Eur. Workshop Reinforcement Learn.*, 2013, pp. 131–142.

[97] B. Woodworth, F. Ferrari, T. E. Zosa, and L. D. Riek, "Preference learning in assistive robotics: Observational repeated inverse reinforcement learning," in *Proc. Mach. Learn. Healthcare Conf.*, 2018, pp. 420–439.

[98] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 2586–2591.

[99] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with Gaussian processes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1–9.

[100] D. Brown and S. Niekum, "Efficient probabilistic performance bounds for inverse reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1–9.

[101] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions by integrating human demonstrations and preferences," 2019, *arXiv:1906.08928*.

[102] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, Chicago, IL, USA, 2008, pp. 1433–1438.

[103] K. Lee, D. Isele, E. A. Theodorou, and S. Bae, "Spatiotemporal costmap inference for MPC via deep inverse reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3194–3201, Apr. 2022.

[104] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–9.

[105] M. Wulfmeier, D. Z. Wang, and I. Posner, "Watch this: Scalable cost-function learning for path planning in urban environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 2089–2095.

[106] C. Wang, C. Pérez-D'Arpino, D. Xu, L. Fei-Fei, K. Liu, and S. Savarese, "Co-GAIL: Learning diverse strategies for human–robot collaboration," in *Proc. Conf. Robot Learn.*, 2022, pp. 1279–1290.

[107] Y. Li, J. Song, and S. Ermon, "InfoGAIL: Interpretable imitation learning from visual demonstrations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–8.

[108] A. Sharma, M. Sharma, N. Rhinehart, and K. M. Kitani, "Directed-info GAIL: Learning hierarchical policies from unsegmented demonstrations using directed information," 2018, *arXiv:1810.01266*.

[109] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, May 1992.

[110] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[111] P. Wolf, C. Hubschneider, M. Weber, A. Bauer, J. Härtl, F. Dürr, and J. M. Zöllner, "Learning how to drive in a real world simulation with deep Q-networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 244–250.

[112] L. Chen, X. Hu, B. Tang, and Y. Cheng, "Conditional DQN-based motion planning with fuzzy logic for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 2966–2977, Apr. 2022.

[113] A. Alizadeh, M. Moghadam, Y. Bicer, N. K. Ure, U. Yavas, and C. Kurtulus, "Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1399–1404.

[114] M. P. Ronecker and Y. Zhu, "Deep Q-network based decision making for autonomous driving," in *Proc.3rd Int. Conf. Robot. Autom. Sci. (ICRAS)*, Jun. 2019, pp. 154–160.

[115] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 22–31.

[116] G. Li, Y. Yang, S. Li, X. Qu, N. Lyu, and S. E. Li, "Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness," *Transp. Res. C, Emerg. Technol.*, vol. 134, Jan. 2022, Art. no. 103452.

[117] Y. Chow, O. Nachum, A. Faust, E. Duenez-Guzman, and M. Ghavamzadeh, "Lyapunov-based safe policy optimization for continuous control," 2019, *arXiv:1901.10031*.

[118] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Phys. D, Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, Jul. 1985.

[119] Y. Yang, Y. Jiang, Y. Liu, J. Chen, and S. E. Li, "Model-free safe reinforcement learning through neural barrier certificate," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1295–1302, Mar. 2023.

[120] S. Mo, X. Pei, and C. Wu, "Safe reinforcement learning for autonomous vehicle using Monte Carlo tree search," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6766–6773, Jul. 2022.

[121] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8248–8254.

[122] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[123] G. Wang, J. Hu, Z. Li, and L. Li, "Harmonious lane changing via deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4642–4650, May 2022.

[124] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, "Driving in dense traffic with model-free reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 5385–5392.

[125] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[126] F. Ye, X. Cheng, P. Wang, C.-Y. Chan, and J. Zhang, "Automated lane change strategy using proximal policy optimization-based deep reinforcement learning," in *Proc. IEEE Intell. Veh. Symp.*, Jul. 2020, pp. 1746–1752.

[127] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12597–12608, Nov. 2020.

[128] Y. Wu, S. Liao, X. Liu, Z. Li, and R. Lu, "Deep reinforcement learning on autonomous driving policy with auxiliary critic network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 7, pp. 3680–3690, Jul. 2023.

[129] X. Liang, T. Wang, L. Yang, and E. Xing, "CIRL: Controllable imitative reinforcement learning for vision-based self-driving," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 584–599.

[130] Y. Tian, X. Cao, K. Huang, C. Fei, Z. Zheng, and X. Ji, "Learning to drive like human beings: A method based on deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6357–6367, Jul. 2022.

[131] Z. Huang, J. Wu, and C. Lv, "Efficient deep reinforcement learning with imitative expert priors for autonomous driving," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 26, 2022, doi: 10.1109/TNNLS.2022.3142822.

[132] J. Wu, Z. Huang, W. Huang, and C. Lv, "Prioritized experience-based reinforcement learning with human guidance for autonomous driving," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 10, 2022, doi: 10.1109/TNNLS.2022.3177685.

[133] W. Hu, Z. Deng, D. Cao, B. Zhang, A. Khajepour, L. Zeng, and Y. Wu, "Probabilistic lane-change decision-making and planning for autonomous heavy vehicles," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 12, pp. 2161–2173, Dec. 2022.

[134] Y. Chen, C. Dong, P. Palanisamy, P. Mudalige, K. Muelling, and J. M. Dolan, "Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 1–9.

[135] T. Shi, P. Wang, X. Cheng, C.-Y. Chan, and D. Huang, "Driving decision and control for autonomous lane change based on deep reinforcement learning," 2019, *arXiv:1904.10171*.

[136] J. Li, L. Sun, J. Chen, M. Tomizuka, and W. Zhan, "A safe hierarchical planning framework for complex driving scenarios based on reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 2660–2666.

[137] Y. Lu, X. Xu, X. Zhang, L. Qian, and X. Zhou, "Hierarchical reinforcement learning for autonomous decision making and motion planning of intelligent vehicles," *IEEE Access*, vol. 8, pp. 209776–209789, 2020.

[138] J. Duan, S. Eben Li, Y. Guan, Q. Sun, and B. Cheng, "Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data," *IET Intell. Transp. Syst.*, vol. 14, no. 5, pp. 297–305, May 2020.

[139] L. Gao, Z. Gu, C. Qiu, L. Lei, S. E. Li, S. Zheng, W. Jing, and J. Chen, "Cola-HRL: Continuous-lattice hierarchical reinforcement learning for autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 13143–13150.

[140] R. Xu, J. Li, X. Dong, H. Yu, and J. Ma, "Bridging the domain gap for multi-agent perception," 2022, *arXiv:2210.08451*.

[141] V. P. Tran, M. A. Garratt, K. Kasmarik, and S. G. Anavatti, "Dynamic frontier-led swarming: Multi-robot repeated coverage in dynamic environments," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 3, pp. 646–661, Mar. 2023.

[142] R. Xu, W. Chen, H. Xiang, L. Liu, and J. Ma, "Model-agnostic multi-agent perception framework," 2022, *arXiv:2203.13168*.

[143] M. Kaushik, N. Singhania, and K. M. Krishna, "Parameter sharing reinforcement learning architecture for multi agent driving," in *Proc. 4th Int. Conf. Adv. Robot.*, New York, NY, USA, 2020, pp. 1–10, doi: 10.1145/3352593.3352625.

[144] J. Wang, T. Shi, Y. Wu, L. Miranda-Moreno, and L. Sun, "Multi-agent graph reinforcement learning for connected automated driving," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1–6.

[145] W. Zhou, D. Chen, J. Yan, Z. Li, H. Yin, and W. Ge, "Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic," *Auto. Intell. Syst.*, vol. 2, no. 1, pp. 5–16, Dec. 2022.

[146] D. Chen, M. Hajidavalloo, Z. Li, K. Chen, Y. Wang, L. Jiang, and Y. Wang, "Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic," 2021, *arXiv:2105.05701*.

[147] S. Han, H. Wang, S. Su, Y. Shi, and F. Miao, "Stable and efficient Shapley value-based reward reallocation for multi-agent reinforcement learning of autonomous vehicles," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8765–8771.

[148] Z. Peng, Q. Li, K. M. Hui, C. Liu, and B. Zhou, "Learning to simulate self-driven particles system with coordinated policy optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 10784–10797.

[149] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. Hoang Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2: Audi autonomous driving dataset," 2020, *arXiv:2004.06320*.

[150] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The ApolloScape dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1–14.

[151] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8740–8749.

[152] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2633–2642.

[153] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[154] E. Santana and G. Hotz, "Learning a driving simulator," 2016, *arXiv:1608.01230*.

[155] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[156] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," in *Proc. Conf. Robot Learn.*, vol. 155, Nov. 2021, pp. 409–418.

[157] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multi-modal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[158] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, and B. Cain., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2443–2451.

[159] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2118–2125.

[160] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION dataset: An international, adversarial and cooperative moTION dataset in interactive driving scenarios with semantic maps," 2019, *arXiv:1910.03088*.

[161] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. Kaesemodel Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," 2023, *arXiv:2301.00493*.

[162] T. Choudhary, V. Dewangan, S. Chandhok, S. Priyadarshan, A. Jain, A. K. Singh, S. Srivastava, K. M. Jatavallabhula, and K. M. Krishna, "Talk2BEV: Language-enhanced Bird's-eye view maps for autonomous driving," 2023, *arXiv:2310.02251*.

[163] S. Dokania, A. H. A. Hafez, A. Subramanian, M. Chandraker, and C. V. Jawahar, "IDD-3D: Indian driving dataset for 3D unstructured road scenes," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, India, Jan. 2023, pp. 4471–4480, doi: 10.1109/WACV56688.2023.00446.

[164] D. Krajzewicz, "Traffic simulation with SUMO—Simulation of urban mobility," in *Fundamentals of Traffic Simulation*. Berlin, Germany: Springer, 2010, pp. 269–293.

[165] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, "LGSVL simulator: A high fidelity simulator for autonomous driving," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[166] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 287–296.

[167] Baidu. (Apr. 5, 2024). *Baidu Apollo*. [Online]. Available: https://github.com/ApolloAuto/apollo

[168] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.

[169] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software*, vol. 4, no. 6, pp. 2–7, 2000. [Online]. Available: http://torcs.sourceforge.net

[170] rFpro. (2024). *The World's Most Accurate Simulation Environment*. [Online]. Available: https://rfpro.com

[171] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Proc. Field Service Robot.*, Berlin, Germay, 2018, pp. 621–635.

[172] L. Chen, Q. Wang, X. Lu, D. Cao, and F.-Y. Wang, "Learning driving models from parallel end-to-end driving data set," *Proc. IEEE*, vol. 108, no. 2, pp. 262–273, Feb. 2020.

[173] X. Li, K. Wang, Y. Tian, L. Yan, F. Deng, and F.-Y. Wang, "The ParallelEye dataset: A large collection of virtual images for traffic vision research," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2072–2084, Jun. 2019.
[174] X. Li, P. Ye, J. Li, Z. Liu, L. Cao, and F.-Y. Wang, "From features engineering to scenarios engineering for trustworthy AI: I&I, C&C, and V&V," *IEEE Intell. Syst.*, vol. 37, no. 4, pp. 18–26, Jul. 2022.
[175] A. Khanum, C.-Y. Lee, and C.-S. Yang, "Involvement of deep learning for vision sensor-based autonomous driving control: A review," *IEEE Sensors J.*, vol. 23, no. 14, pp. 15321–15341, Jul. 2023, doi: 10.1109/JSEN.2023.3280959.

**MANIKANDAN GANESAN** received the B.E. degree in electrical and electronics engineering from the ARM College of Engineering and Technology, Anna University, Chennai, in 2012, and the M.E. degree in mechatronics engineering from Madras Institute of Technology, Anna University, Chennai, in 2014. With a cumulative eight years of teaching experience, he served as a Lecturer with the Department of Electromechanical Engineering, Faculty of Manufacturing, Hawassa University, Ethiopia, from 2016 to 2022. Prior to this, he was an Assistant Professor with the Department of Mechatronics Engineering, Bharath Institute of Higher Education and Research, Chennai, from 2014 to 2016. He is currently a Project Associate and a Research Scholar with the Center for Electric Mobility, a constituent of the Promotion of University Research and Scientific Excellence (PURSE) Project under the aegis with the Department of Science and Technology (DST), Government of India. This role is embedded with the Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai. Accumulating a wealth of knowledge, he has authored more than 15 articles in Scopus-indexed journals and conferences. His contributions have also earned him three international and Indian design patents, alongside copyrights. His research interests include mechatronics, robotics, autonomous vehicles, deep learning, motion planning, unmanned aerial vehicles, and underwater vehicles.

**SIVANATHAN KANDHASAMY** received the B.E. degree in electronics and communication engineering from the K. S. Rangasamy College of Technology, Periyar University, India, in 2003, the M.E. degree in mechatronics from Madras Institute of Technology, Anna University, India, in 2008, and the Ph.D. degree from the SRM Institute of Science and Technology, in 2022, specializing in online motion planning techniques for decentralized dynamic robot swarms. Since 2008, he has been an Assistant Professor with the Department of Mechatronics, SRM Institute of Science and Technology. With more than 17 years of teaching experience, he has also extended his expertise to provide training and consultancy services to industries through the SRM-BRIN Center of Excellence for Automation Technologies. In addition, he founded and led the Autonomous Systems Research Laboratory, driven by a visionary pursuit of cutting-edge research in the field of autonomous systems. Moreover, he serves as a Co-Principal Investigator with the Department of Science and Technology funded project under the PURSE scheme. His research interests include multi-robot SLAM, trajectory planning, optimization and control, autonomous cars, and bio-inspired algorithms for robots. He is an active member of various professional organizations, including IEEE RAS, ISTE, and ISCA.

**BHARATIRAJA CHOKKALINGAM** (Senior Member, IEEE) received the B.E. degree in electrical and electronics engineering from the Kumaraguru College of Engineering, Coimbatore, India, in 2002, the M.E. degree in power electronics and drives from the Government College of Technology, Coimbatore, in 2006, and the Ph.D. degree from the SRM Institute of Science and Technology, in 2015.

He completed the first Postdoctoral Fellowship with the Centre for Energy and Electric Power, Faculty of Engineering and the Built Environment, Tshwane University of Technology, South Africa, in 2016, through the National Research Foundation Funding, and the second Postdoctoral Fellowship with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. He was a Visiting Research Scientist with Northeastern University, in 2018, and a Visiting Researcher with the University of South Africa, in 2019 and 2020. He is currently a Professor with the Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology, Kattankulathur Campus, Chennai, India, where he is also the Centre Head of the Centre for E-Mobility, SRM. He is handling/handling government funded projects total worth Rs. 15.65 Cr, including SERB, DST, and TNSCST. He has authored more than 180 research papers and ten books chapters, and ten IPR patents in his name. He also made three Technologies transfers. He is a Senior Member of IEI and IET. He is also a member, mentor, and advisor of many government and private companies' bodies. He was a recipient of the DST, Indo-U.S. Bhaskara Advanced Solar Energy, in 2017, and the Young Scientists Fellowship, Tamil Nadu State Council for Science and Technology, in 2018, and the World's Top 2% Scientists 2021 and 2022 in the Energy category. For more information: https://www.srmist.edu.in/faculty/dr-c-bharatiraja/.

**LUCIAN MIHET-POPA** (Senior Member, IEEE) was born in 1969. He received the bachelor's degree in electrical engineering, the master's degree in electric drives and power electronics, and the Ph.D. and Habilitation degrees in electrical engineering from the Politehnica University of Timisoara, Romania, in 1999, 2000, 2002, and 2015, respectively. From 1999 to 2016, he was with the Politehnica University of Timisoara. He was also a Research Scientist with Danish Technical University, from 2011 to 2014, and Aalborg University, Denmark, from 2000 to 2002. He held a postdoctoral position with Siegen University, Germany, in 2004. Since 2016, he has been a Full Professor in energy technology with Østfold University College, Norway. He is also the Head of the Research Laboratory "Intelligent Control of Energy Conversion and Storage Systems." He has been granted and participated in more than 25 international projects. He has been awarded more than ten national research grants. He has published more than 200 papers in national and international journals and conference proceedings, and 15 books. His research interests include modeling, simulation, control, and testing of energy conversion systems; and distributed energy resources (DER components and systems, including battery storage systems for electric vehicles as well as hybrid cars and vanadium redox batteries), but also interactive buildings in smart grids.