# Motion Planning for Autonomous Driving: The State of the Art and Future Perspectives

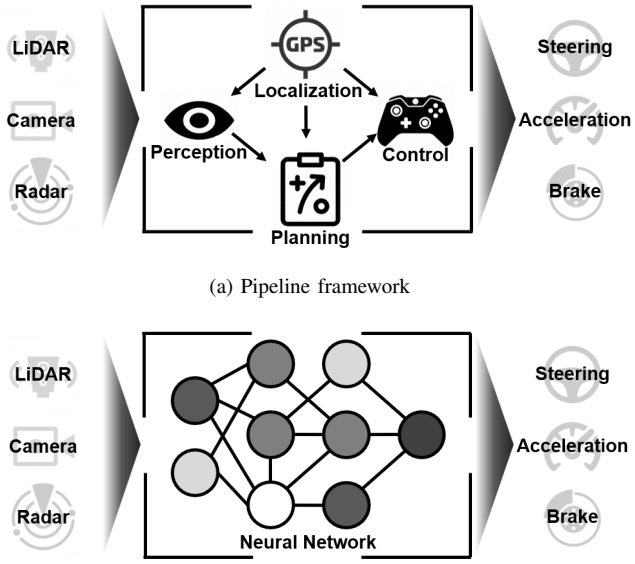
Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, *Senior Member, IEEE,* Long Chen, *Senior Member, IEEE,*

***Abstract*—Intelligent vehicles (IVs) have gained worldwide attention due to their increased convenience, safety advantages, and potential commercial value. Despite predictions of commercial deployment by 2025, implementation remains limited to small-scale validation, with precise tracking controllers and motion planners being essential prerequisites for IVs. This paper reviews state-of-the-art motion planning methods for IVs, including pipeline planning and end-to-end planning methods. The study examines the selection, expansion, and optimization operations in a pipeline method, while it investigates training approaches and validation scenarios for driving tasks in end-to-end methods. Experimental platforms are reviewed to assist readers in choosing suitable training and validation strategies. A side-by-side comparison of the methods is provided to highlight their strengths and limitations, aiding system-level design choices. Current challenges and future perspectives are also discussed in this survey.**

***Index Terms*—Motion planning, pipeline planning, end-to-end planning, imitation learning, reinforcement learning, parallel learning.**

## I. Introduction

INTELLIGENT vehicles (IVs) have attracted significant interest from governments, industries, academia, and the public, owing to their potential to transform transportation through advances in artificial intelligence and computer hardware [2]. The deployment of IVs holds great promise for reducing road accidents and alleviating traffic congestion, thereby improving mobility in densely populated urban areas [3]. Despite remarkable contributions by leading experts in the field, IVs remain primarily confined to limited trial programs due to concerns about their reliability and safety. To enhance situational awareness and improve safety, efficiency, and overall capabilities, IVs are equipped with a variety of sensors. However, even with an array of sensors, an IV stills face challenges in adequately detecting and responding to complex scenarios. Consequently, ensuring the safety, robustness, and adaptability of planning methods becomes crucial for the successful implementation of autonomous driving [4].




(a) Pipeline framework

(b) End-to-end framework

Fig. 1. Pipeline and end-to-end frameworks surveyed in [1]. The pipeline framework for autonomous driving consists of many interconnected modules, while the end-to-end method treats the entire framework as one learnable learning task.

### A. Background

The pipeline planning method, also known as the rule-based planning method, is a well-established category of planners. As depicted in 1a, this method serves as a core component of the pipeline framework and must be integrated with other methods, such as perception [5], localization, and control, to accomplish autonomous driving tasks. A significant

This work was supported in part by National Natural Science Foundation of China under Grant 62273135; Natural Science Foundation of Hubei Province in China under Grant 2021CFB460; National Natural Science Foundation of China under Grant 62103139; 2022 Opening Foundation of State Key Laboratory of Management and Control for Complex Systems under Grant E2S9021119; the Guangdong Provincial Key Laboratory of Interdisciplinary Research and Application for Data Science, BNU-HKBU United International College, project code 2022B1212010006. Guangdong Higher Education Upgrading Plan with UIC research grant R0400001-22 and R201902. (Siyu Teng and Xuemin Hu contributed equally to this work). (Corresponding authors: Zhe Xuanyuan, Fenghua Zhu and Long Chen).

Siyu Teng and Yuchen Li are with BNU-HKBU United International College, Zhuhai, 519087, China and Hong Kong Baptist University, Kowloon, Hong Kong, 999077, China (e-mail: siyuteng@ieee.org).

Xuemin Hu and Peng Deng are with the School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China.

Bai Li is with the State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, and also with the College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China.

Yunfeng Ai is with University of Chinese Academy of Sciences, Beijing, 100049, China.

Dongsheng Yang is with the School of Public Management/Emergency Management, Jinan University, Guangzhou 510632, China.

Lingxi Li is with the Purdue School of Engineering and Technology, Indiana University-Purdue University Indianapolis (IUPUI), Indianapolis, USA.

Zhe Xuanyuan is with the Guangdong provincial key lab of IRADS, BNU-HKBU United International College, Zhuhai, 519087, China.

Fenghua Zhu and Long Chen are with Institute of Automation, Chinese Academy of Sciences,Beijing, China, 100190, and Long Chen is also with Waytous Ltd. (e-mail:fenghua.zhu@ia.ac.cn; long.chen@ia.ac.cn).

Manuscript received April 19, 2021; revised August 16, 2021.

advantage of the pipeline framework is its interpretability, enabling the identification of defective modules when malfunctions or unexpected system behavior occur. In Section-II, the focus is solely on the planning method within the pipeline framework. The pipeline planning method comprises two primary components: global route planning, which generates a road-level path from the origin to the destination, and local behavior/trajectory planning, which generates a short-term trajectory. Although widely used in the industry, the pipeline planning method requires substantial computational resources and numerous manual heuristic functions [6]. This study specifically addresses the expansion and optimization mechanisms of the pipeline planning method.

The end-to-end planning method, also known as the learning-based approach, is the sole component in the end-to-end framework and has become a trend in autonomous vehicle research. As illustrated in 1b, the entire driving framework is treated as a single machine learning task that converts raw perception data into control commands. The driving model acquires knowledge through imitation learning, develops driving policies through reinforcement learning, and continuously self-optimizes via parallel learning. Despite its appealing concept, determining the reasons for model misbehavior can be challenging. Consequently, this study focuses on the network structure, training techniques, and deployment tasks of the end-to-end model.

### B. Comparison

In this subsection, we provide a concise overview of the distinctions between pipeline and end-to-end methods, particularly highlighting their respective advantages and disadvantages.

The pipeline framework, widely implemented in the industry, allows engineers to focus on well-defined sub-tasks and independently improve each sub-model within the entire pipeline. Due to its clear intermediate representations and deterministic decision-making rules, this framework facilitates pinpointing the root cause of errors when unexpected behavior occurs. Moreover, it enables reliable reasoning about how the system generates specific control signals. However, the pipeline framework has some drawbacks. Individual sub-models may not be optimal for all driving scenarios, posing a challenge to the generalization of the framework. Additionally, the concatenation of sub-modules and the numerous manual customization constraints in each sub-model can compromise the robustness and real-time capabilities of the method.

The end-to-end framework optimizes the entire driving task, from raw perception to control signals, as a single deep learning task. By learning optimal intermediate representations for the target task, the framework can attend to any implicit sources of raw data without human-defined information bottlenecks, enhancing its generalization for various scenarios. The end-to-end framework's streamlined architecture, consisting of one or a few networks, also offers superior robustness and real-time capabilities compared to the pipeline framework. However, as research progresses, the end-to-end optimization faces a critical interpretability issue. Without intermediate outputs, tracing the initial cause of an error and explaining why the model arrived at specific control commands or trajectories becomes more challenging.

### C. Paper Structure

In Section II, pipeline planning methods are reviewed, including global route planning and local behavior/trajectory planning, with a particular focus on the expansion and optimization mechanisms. In Section III, end-to-end planning methods are examined, encompassing imitation learning, reinforcement learning, and parallel learning, while exploring network architecture, generalizability, robustness, and validation & verification methods. Additionally, large datasets, simulation platforms, and physical platforms play auxiliary roles in the development of autonomous driving with higher levels of intelligence and mobility. Therefore, other aspects of autonomous driving are summarized in Section IV, including datasets, simulation platforms, and physical platforms. Finally in Section V, current challenges and future directions of autonomous driving are reviewed.

### D. Contributions

This paper presents a comprehensive analysis of the general planning methods for autonomous driving. Broadly speaking, planning methods for autonomous driving can be classified into two categories: pipeline and end-to-end.

There have been numerous state-of-the-art works on motion planning for IVs, however, a comprehensive review encompassing both pipeline and end-to-end methods has yet to be conducted. The pipeline is a classical planning method commonly used in the industry, with general categories outlined in previous research [7], [8]. In this paper, we propose a new classification of pipeline methods that captures the extensively deployed approaches in a manner more relevant to industry selection, based on the expansion and optimization mechanisms of each method. Our proposed classification includes state grid identification, primitive generation, and other approaches. The end-to-end approach has emerged as a popular research direction in recent years, as demonstrated by previous work [1], [9], which illustrates methods for mapping raw perception inputs to control command outputs. In this survey, we not only review the latest achievements in imitation learning (IL) and reinforcement learning (RL) but also introduce a novel category called parallel planning. This category proposes a virtual-real interaction confusion learning method for a reliable end-to-end planning method. Furthermore, we provide a thorough analysis and summary of the latest datasets, simulation platforms, and semi-open real-world testing scenarios, which serve as essential auxiliary elements for the advancement of IVs. To the best of our knowledge, this survey presents the first comprehensive analysis of motion planning methods in various scenarios and tasks.

## II. Pipeline Planning Methods

The pipeline method, also known as the modular approach, is widely used in the industry and has become the conventional approach. This method originates from architectures that

primarily evolved for autonomous mobile robots and consists of self-contained, interconnected modules such as perception, localization, planning, and control.

Planning methods are responsible for calculating a sequence of trajectory points for the ego vehicle's low-level controller to track, typically consisting of three functions: global route planning, local behavior planning, and local trajectory planning [7], [10]. Global route planning provides a road-level path from the start point to the end point on a global map. Local behavior planning decides on a driving action type (e.g., car-following, nudge, side pass, yield, and overtake) for the next several seconds. Local trajectory planning generates a short-term trajectory based on the decided behavior type. In fact, the boundary between local behavior planning and local trajectory planning is somewhat blurred [10], as some behavior planners do more than just identify the behavior type. For the sake of clarity, this paper does not strictly distinguish between the two functions, and the related methods are simply regarded as trajectory planning methods.

This section categorizes the related algorithms into two functions: global route planning and local behavior/trajectory planning. To provide a more detailed analysis and discussion, local behavior/trajectory planning is divided into three components: state grid identification, primitive generation, and other approaches, based on their respective extension methods and optimization theories

### A. Global Route Planning

Global route planning is responsible for finding the best road-level path in a road network, which is presented as a directed graph containing millions of edges and nodes. A route planner searches in the directed graph to find the minimal-cost sequence that links the starting and destination nodes. Herein, the cost is defined based on the query time, preprocessing complexity, memory occupancy, and solution robustness considered. Edsger Wybe Dijkstra is a pioneer in this field and innovatively proposes the Dijkstra algorithm [11], [12] named after him. Lotfi et al. [13] construct a Dijkstra-based intelligent scheduling framework that computes the optimal scheduling for each agent, including maximum speed, minimum movement, and minimum consumption cost. A-star algorithm [14], [15] is another famous algorithm in road-level navigation tasks, it leverages the advantages of the heuristic function to streamline research space. All of these algorithms substantially alleviate the problem of transportation efficiency and garnered significant attention in the field of intelligent transportation systems.

### B. Local Behavior/Trajectory Planning

Local behavior planning and local trajectory planning functions work together to compute a safe, comfortable and continuous local trajectory based on the identified global route from route planning. Since the resultant trajectory is local, the two functions have to be implemented in a receding-horizon way unless the global destination is not far away [16]. It deserves to emphasize that the output of the two functions should be a trajectory rather than a path [17], [18], and the trajectory interacts with other dynamic traffic participants, otherwise, extra efforts are needed for the ego vehicle to evade the moving obstacles in the environment.

Nominally, local planning is done by solving an optimal control problem (OCP), which minimizes a predefined cost function with multiple types of hard or soft constraints satisfied [19], [20]. The solution to the OCP is presented as time-continuous control and state profiles, wherein the desired trajectory is reflected by a part of the state profiles. As shown in Equ. 1, the state space of the vehicle is denoted as $\boldsymbol{z} \in \mathbb{R}^{n_z}$, the control space is presented as $\boldsymbol{u} \in \mathbb{R}^{n_u}$. $\Upsilon$ shows the workspace. The obstacle space as $\Upsilon_{OBS} \subset \Upsilon$and the free space is described as $\Upsilon_{FREE} \subset \Upsilon \setminus \Upsilon_{OBS}$.

$$
\begin{aligned}
&\min_{\boldsymbol{z}(t),\boldsymbol{u}(t),T} J(\boldsymbol{z}(t),\boldsymbol{u}(t)),\\
&\quad s.t.,\dot{\boldsymbol{z}}(t) = f(\boldsymbol{z}(t),\boldsymbol{u}(t));\\
&\qquad \boldsymbol{z} \le \boldsymbol{z}(t) \le \bar{\boldsymbol{u}}, \boldsymbol{u} \le \boldsymbol{u}(t) \le \bar{\boldsymbol{z}}, , t \in [0,T];\\
&\qquad \boldsymbol{z}(0) = \boldsymbol{z}_{init}, \boldsymbol{u}(0) = \boldsymbol{u}_{init};\\
&\qquad g_{end}(\boldsymbol{z}(T),\boldsymbol{z}(T)) \le 0;\\
&\qquad fp(\boldsymbol{z}(t)) \subset \Upsilon_{FREE}, t \in [0,T];
\end{aligned}
\tag{1}
$$

The planning process duration in seconds is described as $T$, where $t \subset T$, and the cost function to be minimized is denoted as $J$. We use the common shorthand $\dot{\boldsymbol{z}}$ to denote the derivative with respect to time, $\dot{\boldsymbol{z}} = \partial \boldsymbol{z}/\partial \boldsymbol{t}$, $\dot{\boldsymbol{u}} = \partial \boldsymbol{u}/\partial \boldsymbol{t}$. The vehicle kinematic is described by the function $f$ and the allowable intervals where $\boldsymbol{z}(t)$ and $\boldsymbol{u}(t)$ are denoted by $[\boldsymbol{z}, \bar{\boldsymbol{z}}]$ and $[\boldsymbol{u}, \bar{\boldsymbol{u}}]$ respectively, where $\boldsymbol{z}$ and $\boldsymbol{u}$ representing the initial values. The inequality $g_{end} \le 0$ models the implicit end-point conditions at $t = T$. Finally, $f$ is a mapping from the vehicle state to its footprints, and $fp(\cdot) : \mathbb{R}^{n_z} \to \mathbb{R}^2$ represents the collision-avoidance constraints. In the following context of this section, we provide a detailed review of the motion planning method based on this scheme.

Since the analytical solution to such an OCP is generally not available, two types of operations are needed to construct a trajectory. Concretely, local planning is divided into three parts, the first type of operation is to identify a sequence of state grids, the second type is to generate primitives between adjacent state grids, and The third is an organic combination of the first two.

*1) State Grid Identification:* State grid identification can be done by search, selection, optimization, or potential minimization. Search-based methods abstract the continuous state space related to the aforementioned OCP into a graph and find a link of states there. Prevalent search-based methods include A* search [21] and dynamic programming (DP) [22]. Many advanced applications of these algorithms have pushed its influence to the top of the heap, such as Hybrid A* [23], Bi-direction A*, Semi-optimization A* [24], and LQG framework [22]. Selection-based methods decide the state grids in the next one or several steps by seeking the candidate with the optimal cost function. Greedy selection [25] and Markov decision process (MDP) series methods typically [26], [27] fall into this category.

An optimization-based method discretizes the original OCP into a mathematical program (MP), the solution of which

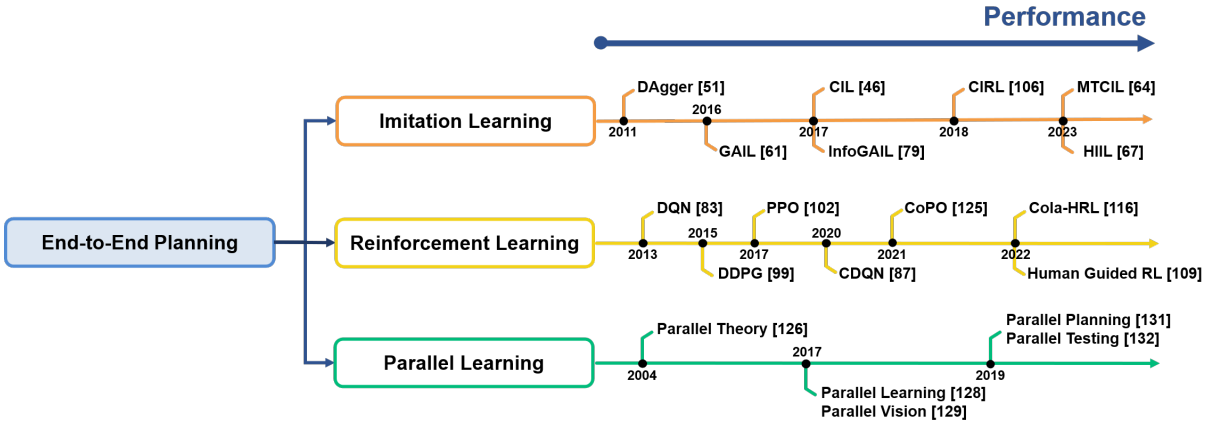



Fig. 2. The critical method survived in End-to-End Planning Section. The time axis (dark blue) represents the progressiveness of the survived methods, and the performance of the methods is better with the latter proposed time.

are high-resolution state grids [28], [29]. MP solvers are further classified as gradient-based and non-gradient-based ones; gradient-based solvers typically solve nonlinear programs [20], quadratic programs [28], [30], [31], quadratically constrained quadratic programs [32] and mix-integer programs; non-gradient-based solvers are typically represented by metaheuristics [33]. Multiple previous methods could be combined to provide a coarse-to-fine local behavior/motion planning strategy.

*2) Primitive Generation:* Primitive generation commonly manifests as closed-form rules, simulation, interpolation, and optimization. Closed-form rules stand for methods that compute primitives by analytical methods with closed-form solutions. Typical methods include the Dubins/Reeds-Shepp curves [34], polynomials [25], and theoretical optimal control methods [35], [36]. Simulation-based methods generate trajectory/path primitives by forwarding simulation, which runs fast because it has no degree of freedom [21]. Interpolation-based methods are represented by splines or parameterized polynomials [37]. Optimization-based methods solve a small-scale OCP numerically to connect two state grids [38], [39].

*3) Other Approaches:* State grid identification and primitive generation are two fundamental operations to construct a trajectory. Both operations may be organized in various ways. For example, Kuwata et al. [40] integrate both operations in an iterative loop; Hu et al. [38] build a graph of primitives offline before online state grid identification; Fan et al. [30] identify the state grids before generating connective primitives. If a planner only finds a path rather than a trajectory, then a time course should be attached to the planned path as a post-processing step [39]. This strategy, denoted as path velocity decomposition (PVD), has been commonly used because it converts a 3D problem into two 2D ones, which largely facilitates the solution process. Conversely, non-PVD methods directly plan trajectories, which has the underlying merit to improve the solution optimality [22], [41]–[43].

Recent studies in this research domain include how to develop specific planners that fit specific scenarios/tasks particularly [16], [42], and how to plan safe trajectories with imperfect upstream/downstream modules [42]. The past decades have seen increasingly rapid progress in the autonomous driving field. In addition to the advances in computing hardware, this rapid progress has been enabled by major theoretical progress in the computational aspects of mobile robot motion planning theory. Research efforts have undoubtedly been spurred by the improved utilization and safety of road networks that intelligent vehicles would provide.

## III. End-to-End Planning Methods

End-to-end stands for the direct mapping from raw sensor data into trajectory points or control signals. Because of its ability to extract task-specific policies, it has achieved great success in a variety of fields [9]. Compared with the pipeline method, there is no external gap between the perception and control modules, and seldom human-customized heuristics are embedded, so the end-to-end method deals with vehicle-environment interactions more efficiently. End-to-end has a higher ceiling, with the potential to achieve expert performance in the autonomous driving field. This section categorizes the end-to-end method into three distinct types from learning methods: imitation learning using supervised learning, reinforcement learning utilizing unsupervised learning, and parallel learning incorporating confusion learning. Fig. 2 further clarification of the structural relationships fo end-to-end planner, highlighting the performance and progressiveness of the reviewed methods.

### A. Imitation Learning

Imitation learning (IL) refers to the agent learning policy based on expert trajectory, which generally provides expert decisions and control information [44]. Each expert trajectory contains a sequence of states and actions, and all "state-action" pairs are extracted to construct datasets. In the IL task, the model leverages constructed dataset to learn the latent relationship between state and action, the state stands for a feature and the action demonstrates labels. Thus, the specific objective of IL is to appraise the most fitness mapping between state and action, so that the agent achieves the expert

TABLE I
THE CRUCIAL REVIEWS AND RELATIVE INFORMATION OF EACH FAMOUS END-TO-END MODELS IN AUTONOMOUS DRIVING.

| Article | Category | Input | Output | Implement Tasks | Auxiliary Method | Dataset |
|---|---|---|---|---|---|---|
| Bojarski et al. [45] | **BC** | monocular image | steering angle | lane Keeping | CNN is the only component of end-to-end model | physical & simulate platform |
| Codevilla et al. [46] | **BC** | monocular image | control information | simulation navigation task | High-level commands as a switch to select the branch | Carla |
| Chen et al. [47] | **BC** | monocular image | control information | simulation navigation task | Affordance is used to predict control actions | TORCS Dataset & KITTI |
| Sauer et al. [48] | **BC** | monocular video & directional input | control information | physical navigation task | Conditional affordance is trained to calculate intermediate representations | Carla |
| Zeng et al. [49] | **BC** | Lidar data & HD Map | trajectory, scenario representation | physical navigation task | The intermediate representation is used to improve the model's interpretability | physical dataset collected in North America |
| Sadat et al. [50] | **BC** | Lidar data & HD Map | trajectory, scenario representation | physical navigation task | A joint system with interpretable intermediate representations for E2E planner | physical dataset collected in North America |
| Ross et al. [51] | **DPL** | monocular image | control information | autonomous racing competition | An iterative algorithm is proposed to guarantee the performance in corner cases | 3D racing simulator |
| Zhang et al. [52] | **DPL** | monocular image | control information | autonomous racing competition | Embedded query-efficient model to reduce the request for expert trajectories | racing car simulator |
| Yan et al. [53] | **DPL** | LiDAR, ego-vehicle speed, Sub-goal | control information | physical & simulation navigation task | The novice and the expert policy is fused to control the robot | physical and simulate platform |
| Li et al. [54] | **DPL** | monocular image & sub-goal | waypoint, control information | autonomous racing | A reward-based online method learns from multiple experts | Sim4CV |
| Ohn-Bar et al. [55] | **IRL** | monocular image | control information | simulation navigation task | Scenario context is embedded into the policy learning network | Carla |
| Levine et al. [56] | **IRL** | BEV image | control information | keep the lane, change lanes & takeover | The Gaussian algorithm is used to learn the relevance of features in expert trajectories. | Highway driving simulator |
| Brown et al. [57] | **IRL** | monocular image | control information | keep the lane, change lanes & takeover | The high-confidence upper bounds on the $alpha$-worst-case are embedded into the policy network. | Highway driving simulator |
| Palan et al. [58] | **IRL** | monocular image | control information | keep the lane, change lanes & takeover | A globally normalized reward function is constructed. | Lunar lander simulator |
| Ziebart et al. [59] | **IRL** | Road network, Sub-goal, & GPS Data | control information | long range autonomous navigation task | A probabilistic approach is proposed for maximum entropy | Driver route modeling |
| Lee et al. [60] | **IRL** | monocular image | control information, costmap | keep the lane, change lanes & takeover | The query generation process is used to improve the generalization | NGSIM & Carla |
| Ho et al. [61] | **IRL** | monocular image | control information | keep the lane, change lanes & takeover | GAN is integrated into the end-to-end model | Carla |
| Phan et al. [62] | **IRL** | BEV image, HD map, obstacle information | Control information | physical navigation task | A three-step IRL planner is proposed | physical dataset from the Las Vegas Strip |

trajectories as much as possible. The formulation of IL is summarized in summarized as follows:

Given a dataset $\mathbb{D}$ comprising "state-action" pairs $(s, a)$ generated from expert trajectories $\pi^*$, the primary objective of IL is to learn a policy $\pi_\theta(s)$ that closely approximates the expert trajectories in any input state $s$, as determined by Equ. 2:

$$\arg\min_{\theta} \mathbb{E}_{s\sim P(s|\theta)} L(\pi^*(s), \pi_\theta(s)), \tag{2}$$

where $P(s|\theta)$ stands for the distribution of the current state from the trained policy $\pi_\theta$.

Based on this formulation, three widely used training methods are survived in this part [63], first manifests as a negative method, named behavioral cloning (BC); The second builds on BC, named direct policy learning (DPL); The last is a task-dependent method, named inverse reinforcement learning (IRL) method. Table I presents all famous imitation learning methods reviewed in this part.

*1) Behavioral Cloning:* Behavioral Cloning (BC) manifests as the primary method of IL in autonomous driving [45], [64]. The agent leverages expert trajectories to the training model and then replicates the policy using a classifier/regressor. BC is a passive method, where the objective is to learn the target policy by passively observing the complete execution of commands. This requires the premise that the

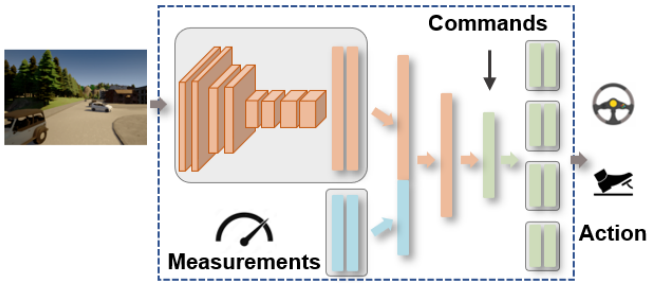
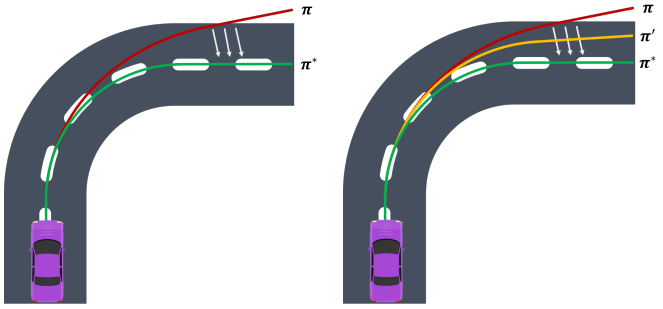



Fig. 3. The model proposed in [46]. Measurements stand for the velocity of ego-vehicle. The high-level command, including straight, left, right, and lane following. Actions are control signals including steering, accretion, and brake.




(a) DAgger (b) DAgger by coaching

Fig. 4. The DAgger method [51] for Autonomous Driving Navigation Task.

state-action pairs in all trajectories are independent.

Bojarski et al. [45] construct a pioneering framework for BC, which trains a convolutional neural network to only compute steering from a front-view monocular camera. This method exclusively outputs lateral control while ignoring longitudinal commands, rendering it can only be implemented in a limited number of uncomplicated scenarios. Codevilla et al. [46] proposed a famous IL model, named conditional imitation learning (CIL), which contains both lateral and longitudinal control, as shown in Fig. 3. Monocular images, velocity measurement of ego-vehicle, and high-level commands (straight, left, right and lane following) are used as input to CIL, and both predicted longitude and latitude control commands as output. Each command acts as a switch to select a specialized sub-module. CIL is a milestone for the CL method in autonomous driving and demonstrates that the convolutional neural network (CNN) can learn to perform lane and road tracking tasks autonomously.

Based on CIL [46], many researchers include additional information such as global route, location information, or point cloud in the input stage [65]–[67]. These methods demonstrate strong generalization and robustness in various conditions, because of sufficient perception data input.

Because of its novel structure, IL methods exclude uncertainty estimation among different sub-systems and lead to fewer feedback milliseconds. However, this characteristic leads to a significant drawback, lack of interpretability, which does not provide sufficient reasons to explain the decisions. Many researchers try to address this pain point by inserting the intermediate representation layer. Chen et al. [47] propose a novel paradigm, named direct perception method, to predict an affordance for urban autonomous driving scenarios. The affordance represents a BEV format that clearly displays features about the surrounding environment and then is fed to a low-level controller to generate steering and acceleration. Sauer et al. [48] further propose an advanced direct perception model, which leverages video and high-level commands to intermediate representations and computes control signals as output. Compared with [47], this model can handle complex scenarios in urban traffic scenarios. Urtasun and her team also propose two interpretable end-to-end planners [49], [50], which leverage raw LiDAR data and a High-Definition Map (HD Map) to predict safe trajectories and intermediate representations. The representations demonstrate how the policy responds to surrounding scenarios. Compared with previous methods that only use the monocular images as inputs, these planners can predict trajectories more safely, owing to the resource-rich input information.

The main feature of the BC method is that only experts can generate training examples, which directly leads to the training set being a subset of the states accessed during the execution of the learned policy [68]. Therefore, when the dataset is biased or overfitted, the method is limited to generalize. Moreover, when the agent is guided to an unknown state, it is hard to learn the correct recovery behavior.

*2) Direct Policy Learning:* Direct Policy Learning (DPL), a training method based on BC, evaluates the current policy and then obtains more suitable training data for self-optimization. Compared with BC, the main advantage of DPL leverages expert trajectories to instruct the agent how to recover from current errors [63]. In this way, DPL alleviates the limitation of BC due to insufficient data. In this section, we summarize a series of DPL methods.

Ross et al. [51] construct a classical online IL method named Dataset Aggregation (DAgger) method. This is an active method based on the Follow-the-Leader algorithm [63], each validation iteration is an online learning example. The method modifies the main classifier or regressor on all state−action pairs experienced by the agent. DAgger is a novel solution for sequential prediction problems, however, its learning efficiency might be suppressed by the far distance between policy space and learning space. In reply, He et al. [69] propose a DAgger by coaching algorithm which employs a coach to demonstrate easy-to-learn policies for the learner and the demonstrated policies gradually converge to label. To better instruct the agent, the coach establishes a compromised policy which is not much worse than a ground truth control signal and much better than novice predicted action. As shown in Fig. 4, $\pi$ is the predicted command, $\pi^*$ shows the expert trajectory, and $\pi^{'}$ presents the compromised trajectory. $\pi^{'}$ is much easier than $\pi^*$ for agent to learn sub-optimal policy in each iteration, and the policy is asymptotically optimal.

Other researchers also point out some drawbacks of DAgger methods [51], [69]: inefficient query, inaccurate data collector, and poor generalization. In reply, Zhang et al. [52] propose the SafeDAgger algorithm, which intends to improve the query

efficiency of DAgger and can further reduce the dependence on label accuracy. Hoque et al. [70] propose a ThriftyDAgger model, which integrates human feedback on corner cases, Yan et al. [53] propose a novel DPL training scheme for navigation tasks in mapless scenarios, both of them improve the generalization and robustness of the model.

DAgger-based methods reduce dataset dependency and improve learning efficiency, however, these methods cannot distinguish between good or bad expert trajectories and ignore the learning opportunity from unfitness behaviors. In reply, Li et al. [54] propose the observational imitation learning (OIL) method, which predicts the control commands from the monocular image and embeds waypoints as intermediate representations. OIL manifests as an online learning policy based on a reward function, it could learn from multi-experts and abandon the wrong policies.

To fine-tune the agent policy in perception-to-action methods, Ohn-Bar et al. [55] propose a method for optimizing situational driving policies which effectively captures reasoning in different scenarios, shown in Fig. 5. The training policy is divided into three parts. First, the model learns sub-optimal policies by the BC method. Second, context embedding is trained to learn scenario features. Third, refined the integrated model by online interaction with the simulation and collect better data by a DAgger-based method.

DPL is an iterative online learning policy that alleviates the requirements for the volume and distribution of dataset, while facilitating the continuous improvement of policies by effectively eliminating incorrect ones.

*3) Inverse Reinforcement Learning:* Inverse reinforcement learning (IRL) is designed to circumvent the drawbacks of the aforementioned methods by inferring the latent reasons between input and output. Similar to the prior methods, IRL needs to collect a set of expert trajectories at the beginning. However, instead of simply learning a state-action mapping, these expert trajectories are first inferred and then the behavioral policy is optimized based on the elaborate reward function. IRL method can be classified into three distinct categories, max-margin methods, Bayesian methods, and maximum entropy methods.

The max-margin method leverages expert trajectories to evaluate a reward function that maximizes the margin between the optimal policy and estimates sub-optimal policies. these methods represent reward functions with a group of features utilizing a linear combination algorithm, where all features are considered independent.

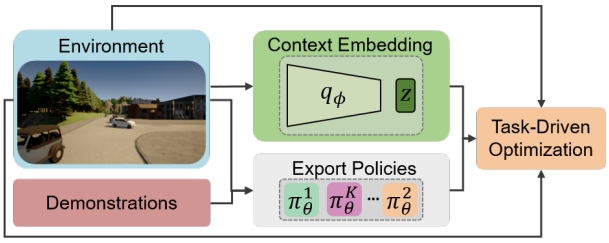



Fig. 5. Training policies propose in [55]. Export policies learn sub-optimal policy. Context Embedding is trained to learn scenarios. Both Context Embedding and Export Policies are fine-tuning in Task-Driven Optimization by Online method.

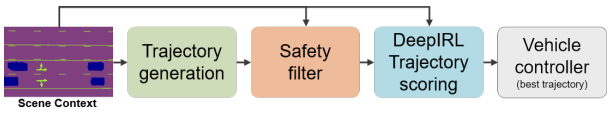



Fig. 6. The system proposed in [62] is divided into three stages: trajectory generation, safety filtering, and trajectory scoring.

Andrew Wu [71] is a pioneer in this field, he introduces the first max-margin IRL method, which puts forward three algorithms for computing the refined reward function. Furthermore, Pieter et al. [72] devise an optimized algorithm based on [71], which assumes that an expert reward function can be expressed as a manually crafted linear combination of known features, with the objective of uncovering the latent relationships between weights and features.

The limitation of prior methods is that the quality and distribution of the expert trajectories sets an upper bound on the performance of the method. In reply, Umar et al. [73] propose a game-theoretic-based IRL method named multiplicative weights for apprenticeship learning, it has the capability to import prior policy to the agent about the weight of each feature and leverages a linear programming algorithm to modify the reward function so that its policy is stationary.

In addition, Phan-Minh et al. [62] propose an interpretable planning system, as shown in Fig. 6. The trajectory generation module leverages perception information to compute a set of future trajectories. The safety filter is used to guarantee the basic safety with an interpretable method. DeepIRL trajectory scoring the predicted trajectories, which is the core contribution of this system. Furthermore, [74] and [75] propose preference-inference formulation, users can choose actions according to their personal preferences, which indeed improves the performance of the model.

The second part of IRL is Bayesian methods, which often leverage the optimized trajectory or the prior distribution of the reward to maximize the posterior distribution of the reward. The first Bayesian IRL is proposed by Ramachandran et al. [76], which references the IRL model from a Bayesian perspective and inferences a posterior distribution of the estimated reward function from a prior distribution. Levine et al. [56] integrate a kernel function into the Bayesian IRL model [76] to improve the accuracy of estimating reward and promote the performance in unseen driving.

Furthermore, Brown et al. [57] construct a sampling-based Bayesian IRL model, which utilizes expert trajectories to calculate practical high-confidence upper bounds on the $\alpha$-worst-case difference in expected return under the unseen scenarios without a reward function. Palan et al. [58] propose DemPref model, which utilizes the expert trajectory to learn a coarse reward function, the trajectory is used to ground the (active) query generation process, to improve the quality of the generated queries. DemPref alleviates the efficiency problems faced by standard preference-based learning methods and does not exclusively depend on high-quality expert trajectories.

The third part of IRL is the maximum entropy method, which is defined by using maximum entropy in the opti-

mization routine to estimate the reward function. Compared with the previous IRL method, Maximum entropy methods are preferable for continuous spaces and have the potential ability to address the sub-optimal impact of expert trajectories. The first Maximum Entropy IRL model is proposed by Ziebart [59], which leverages the same method as [71] and could alleviate both noises and imperfect behavior in the expert trajectory. The agent attempts to optimize the reward function under supervision by linearly mapping features to rewards.

And then, many researchers [60], [61], [77] implement the maximum entropy IRL to physical end-to-end autonomous driving. Among them, [61] propose Generative Adversarial Imitation Learning (GAIL), which has become a classical algorithm in this field. GAIL leverages a generative adversarial network (GAN) to generate the distribution of expert trajectories with a model-free method in order to alleviate the problem of state drift caused by insufficient datasets. Because of sufficient reconstruction expert trajectories and competitive policies, GAIL achieves performance comparable to that of human drivers in specific scenarios. Based on [61], many works have been proposed, such as InfoGAIL [78], Directed-InfoGAIL [79], Co-GAIL [80], all of them achieve competitive results in their implement fields.

IRL provides several excellent works for autonomous driving, however, like the aforementioned methods, it also has long tail problems in corner cases. How to effectively improve the robustness and interpretability of IRL is also a future direction.

The objective of IL methods is to acquire state-to-action mapping from expert trajectories. However, the generalizability of the method may be compromised when the dataset is intrinsically flawed (e.g., overfitting or uneven distribution) [68]. Additionally, when the agent is guided to an unknown state, predicting the correct behavior becomes a formidable challenge. To overcome these limitations, many researchers [46], [64], [67] have significantly enriched the dataset distribution using data augmentation and the combination of real and virtual data. These efforts ensure the generalizability of the methods and obtain competitive results.

### B. Reinforcement Learning

IL methods require large amounts of manually labeled data, and diverse drivers may arrive at entirely distinct decisions when presented with identical situations, which leads to uncertainty quandaries during training. In order to obviate the hunger for labeled data, some researchers have endeavored to utilize reinforcement learning (RL) algorithms for autonomous decision planning. Reinforcement learning refers to the agent learning policy by interacting with an environment. Rather than imitating expert behavior, the goal of an RL agent is to maximize the cumulative numerical rewards from its environment via trial and error. By consistently interacting with the environment, the agent gradually acquires knowledge of the optimal policy to attain the target.

Markov decision processes (MDPs) are typically used to formulate the RL problem. An MDP consists of a state space $\mathcal{S}$, and an action space $\mathcal{A}$, a reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a transition function $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, and a discount factor $\gamma$ that trades instantaneous over future rewards, i.e. a tuple $(\mathcal{S}, \mathcal{A}, R, T, \gamma)$. At each time step $t$, the agent finds itself in a state $s \in \mathcal{S}$ and selects an action $a \in \mathcal{A}$ according to its policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Then the environment enters a new state $s' \in \mathcal{S}$ with a transition probability $T$, where a new action is selected, and so on. Along with the state transition, the environment also gives rise to rewards $r$, special numerical values that the agent seeks to maximize over time through its choice of actions. The goal is to find the optimal policy $\pi^*$, which results in the highest expected sum of discounted rewards [81]:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[ \sum_{t=0}^{N-1} \gamma^t r_{t+1} \mid s_0 = s \right], \tag{3}$$

where the initial states $s \in \mathcal{S}$. The horizon $N$ is the number of time steps, and reward $r_t = R(s_t, a_t)$, and $\gamma \in [0, 1]$ is the discount factor. The expectation means that the agent takes actions following the policy $\pi$ and gets corresponding discounted total rewards.

Based on this formulation, two main RL approaches to achieve optimal policies are developed, e.g., value-based reinforcement learning and policy-based reinforcement learning. Furthermore, based on those approaches, hierarchical reinforcement learning (HRL) and multi-agent reinforcement learning (MARL) are promising ways to solve more complex problems and better fit real driving scenarios. Training autonomous vehicles with RL methods has become a growing trend in end-to-end autonomous driving research.

*1) Value-based Reinforcement Learning:* Value-based methods try to estimate the value of different actions in a given state and learn to assign a value to each action based on the expected reward that can be obtained by taking that action in that state. The agent learns to associate the rewards with the states and actions taken in the environment and leverages this information to make optimal decisions [82].

Among value-based methods, Q-Learning [83] stands out as the most prominent. The framework for implementing Q-Learning in end-to-end planning is illustrated in Fig. 7. Mnih et al. [84] propose the first deep learning method by a Q-learning based approach that learns directly from screenshots to control signals. Furthermore, Wolf et al. [85] introduce the Q-learning method into the intelligent vehicle field, they define five different driving maneuvers in the Gazebo simulator [86], and the vehicle chooses a corresponding maneuver based on the image information. For the purpose of alleviating the problem of poor stability with high-dimensional perception input. The Conditional DQN [87] method is proposed, which leverages a defuzzification algorithm to enhance the predictive stability of distinct motion commands. The proposed model achieves a performance comparable to human driving in specific scenarios

In order to perform high-level decision-making for IVs on specific scenarios, Alizadeh et al. [88] train a DQN agent combined with DNN which outputs two discrete actions. The safety and agility of the ego vehicle can be balanced on-the-go, indicating that the RL agent can learn an adaptive behavior. Furthermore, Ronecker et al. [89] propose a safer navigating

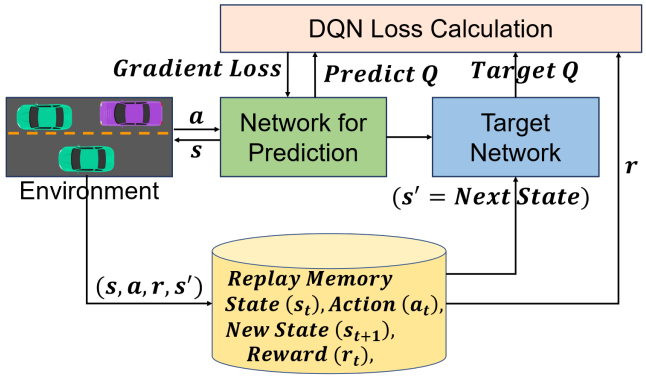



Fig. 7. The architecture of DQN-based end-to-end autonomous driving method.

method for IVs in highway scenarios by combining Deep Q-Networks from control theory. The proposed network is trained in simulation for central decision-making by proposing targets for a trajectory planner, which shows that the value-based RL can produce efficient and safe driving behavior in highway traffic scenarios.

The security of end-to-end autonomous driving also raises significant apprehension. Constrained Policy Optimization (CPO) [90] is a pioneering general-purpose policy exploit algorithm for constrained reinforcement learning with guarantees for near-constraint satisfaction at each iteration. Building on this, [91] and [92] present the Safety Gym benchmark suite and validate several constrained deep RL algorithms under constrained conditions. Li et al. [93] introduce a risk awareness algorithm into DRL frameworks to learn a risk-aware driving decision policy for lane-changing tasks with the minimum expected risk. Chow et al. [94] propose safe policy optimization algorithms that employ a Lyapunov-based approach [95] to address CMDP problems. Furthermore, Yang et al. [96] construct a model-free safe RL algorithm that integrates policy and neural barrier certificate learning in a stepwise state constraint scenario. Mo et al. [97] leverage Monte Carlo Tree Search to reduce unsafe behaviors on overtaking subtasks at highway scenarios.

*2) Policy-based Reinforcement Learning:* The value-based approach is limited to providing discrete commands. However, autonomous driving is a continuous process, continuous commands within an uninterrupted span can be controlled at a fine-grained level. Therefore, the continuous approach is better for vehicle control. Policy-based methods hold the potential for high ceilings in high-dimensional action spaces with continuous control commands. These methods exhibit superior convergence and exploration than value-based methods.

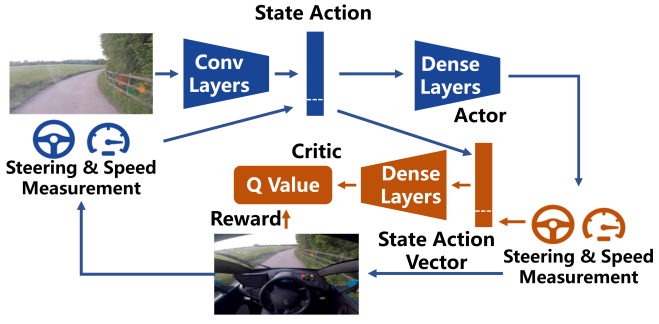



Fig. 8. The actor-critic algorithm used to learn a policy and value function for driving proposed in [98].

The execution of RL on real-world IVs is a challenging assignment. Kendall et al. [98] implement the Deep Deterministic Policy Gradient (DDPG) [99] algorithm on an actual intelligent vehicle, performing all exploration and optimization on-board, as shown in Fig. 8. Monocular images are the only input, the agent learns the lane-following policy and achieves human-level performance in a 250m road test. This work marks the first application of implementing deep reinforcement learning on a full-sized autonomous vehicle. To further enhance driving safety and comfort, Wang et al. [100] introduce an innovative method for IVs based on the lane-change policy of human experts. This method can be executed on single or multiple vehicles, facilitating smooth lane changes without the need for V2X communication support.

To alleviate the challenge of autonomous driving on congested roads, Saxena et al. [101] employ the proximal policy optimization (PPO) algorithm [102] to learn a control policy in a continuous motion planning space. Their model implicitly simulates interaction with other vehicles to avoid collisions and enhance passenger comfort. Building on this work, Ye et al. [103] leverage PPO to learn an automated lane change policy on real highway scenarios. Taking the ego vehicle and its surrounding vehicle states as input, the agent learns to avoid collisions and to drive in a smooth manner. Several other studies [104], [105] have also demonstrated the efficacy of PPO-based RL algorithms in end-to-end autonomous driving, since PPO can provide better performance for both the efficiency of policy learning and the diversity during trajectory exploring.

Training a policy from scratch in RL is frequently time-consuming and difficult. Combining RL with other methods such as imitation learning (IL) and curriculum learning may serve as a viable solution. Liang et al. [106] combine IL and DDPG together to alleviate the problem of low efficiency in exploring the continuous space, an adjustable gating mechanism is introduced to selectively activate four different control signals, which allows the model to be controlled by a central one. Tian et al. [107] leverage an RL method of learning from expert experience to implement trajectory-tracking tasks, which are trained in two steps, an IL method adopted in [66] and a continuous, deterministic, model-free RL algorithm to further fine-turn the method.

To address the learning efficiency limitations of RL methods, Huang et al. [108] devise a novel method, which incorporates human prior knowledge in RL methods. When confronted with the long-tail problem of autonomous driving, many researchers have turned their perspective to the exploitation of expert human experience. Wu et al. [109] propose a human guidance-based RL method which leverages a novel prioritized experience replay mechanism to improve the efficiency and performance of the RL algorithm in extreme scenarios, the framework of the proposed method is shown in Fig 9. This method is validated in two challenging autonomous driving tasks and achieves a competitive result. Therefore, improving

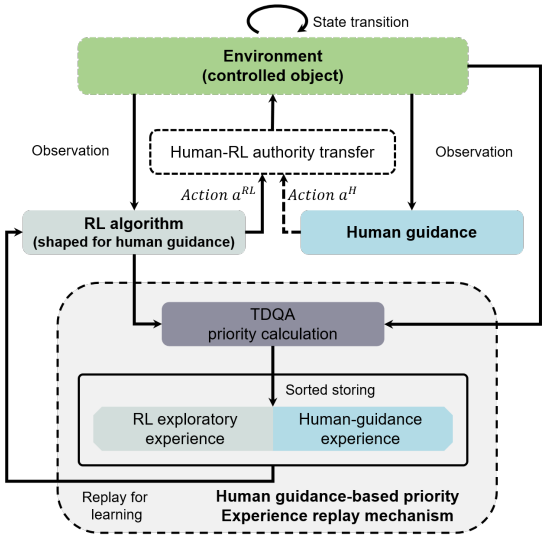



Fig. 9. Framework of the proposed human guidance-based RL algorithm [109].

the performance of driving tasks may require the combination of multiple methods and the design of task-specific training methods.

*3) Hierarchical Reinforcement Learning:* RL methods have shown great promise in various domains, however, these methods are often criticized for difficult training. Especially in the autonomous driving field, non-stationary scenarios and high-dimensional input data cause intolerable training hours and resource usage [110]. Hierarchical reinforcement learning (HRL) decomposes the total problem into a hierarchy of subtasks, and each subtask has its own goal and policy. The subtasks are organized in a hierarchical manner, with higher-level subtasks providing context and guidance for lower-level ones. This hierarchical organization allows the agent to focus on smaller subproblems, reducing the complexity of the learning problem and making it more tractable.

Forcing the lane-changing task, Chen et al. [111] propose a two-level method. The high-level network learns policies for deciding whether to execute a lane change action, while the low-level network learns policies for executing the chosen commands. [112] and [113] also present a two-stage HRL methodology based on [111], where [112] needs to employ the pure pursuit to track the output trajectory points, and [113] integrates position, velocity and heading of ego-vehicle to further improve the performance of the low-level controller. All these proposed methods provide a promising solution for developing robust and safe autonomous driving systems.

The generalizability of HRL is a hot research point. Lu et al. [115] propose an HRL approach for autonomous decision-making and motion planning in complex dynamic traffic scenarios, as shown in Fig. 10. The approach consists of a high-level layer and a low-level planning layer, the high-level layer leverages a kernel-based least-squares policy iteration algorithm with uneven sampling and pooling strategy (USP-KLSPI) to solve the decision-making problems. Duan et al. [114] divide the whole navigation task into three models. The master policy network is trained to select the appropriate driving task, this policy greatly enhances the generalizability and effectiveness of the model. For the purpose of further improving decision quality in complex scenarios, Cola-HRL [116] is presented based on [114], this method consists of three main components: a high-level planner, a low-level controller, and a continuous-lattice representation of the state space. The results show that the Cola-HRL outperforms other SOTA methods for making high-quality decisions in various scenarios.

*4) Multi-Agent Reinforcement Learning:* In real scenarios, diverse traffic participants are commonly present, and their interactions can have a significant impact on the policy of each other [117]. In the single-agent system, the behavior of other participants is usually controlled based on pre-defined rules, and the predicted behavior of the agent may overfit the other participants, thus leading to a more deterministic policy other than in a multi-agent one [118]. Multi-Agent Reinforcement Learning (MARL) is designed to learn the decision-making policies of multiple agents in the environment. Decentralized partially observable Markov decision processes (DEC-POMDPs) are a typical formalization of MARL, as in many real-world domains, it is not possible for agents to observe all features of the environment state, and all agents interact with the environment in a decentralized way. Furthermore, the state space expands exponentially with the number of agents, making it more challenging and slow to train a multi-agent system (MAS) [119].

To reduce the impact of "the dimension explosion", some effective learning schemes are proposed. Kaushik et al. [120] use a simple parameter-sharing DDPG to train the agent for two distinct tasks. By injecting the task into the observation space as a command, the same agent can act both competitively or cooperatively. Wang et al. [121] train autonomous agents in three scenarios: a ring network, a figure-of-eight network, and a mini city with various scenarios. Graph information sharing between each agent is integrated in the approach with PPO for continuous action generation, and vehicle communication is permitted within a certain range.

Although RL has been widely studied for lane-changing decision makings, those studies are mainly focused on a single-agent system. MARL methods provide a global perspective on multi-vehicle control. Zhou et al. [122] formulate

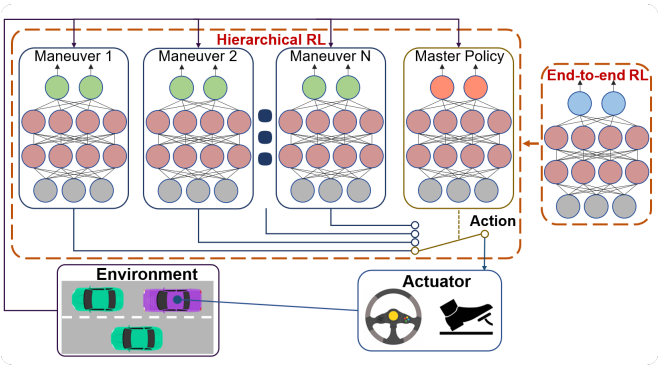



Fig. 10. The framework of hierarchical reinforcement learning (HRL) for self-driving decision-making proposed in [114].

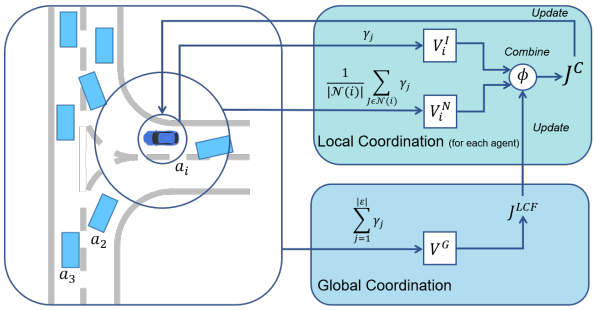



Fig. 11. The framework of the CoPO method proposed in [125]: the Local Coordination Factor (LCF) describes an agent's preference of being selfish, cooperative, or competitive. A Local Coordination for each policy and a Global Coordination to update global LCF are both performed during training.

the lane-changing decision-making of multiple autonomous vehicles coexisting with human-driven vehicles in a mixed-traffic highway scenario. Beyond simple tasks, MARL approaches have great potential to solve decision and planning problems in complex scenarios. Chen et al. [123] train agents to evade collisions in a time-critical merging highway scenario. The agents observe the locations and the velocities of the surrounding vehicles and then select corresponding actions.

Credit assignment is vital for policy learning in cooperative multi-agent scenarios. Han et al. [124] introduce an effective reward reallocating mechanism to motivate stable cooperation among IVs using a cooperative policy learning algorithm with Shapley value reward reallocation. Each vehicle's states include position, velocity, acceleration, image and point clouds captured by the onboard camera and LiDAR sensors respectively. The experimental results demonstrate significant improvement of the mean episode system reward in connected autonomous vehicles. Instead of reallocating rewards between agents, Peng et al. [125] incorporate the social psychology principle to learn the neural controller of Self-Driven Particles (SDP) system, in which each constituent agent is self-interested and the relationship between them is constantly changing. The proposed method, Coordinated Policy Optimization (CoPO), performs local coordination between the agent and its neighbor vehicles global coordination, as shown in Fig 11. Taking raw LiDAR data as input and continuous actuator signals as output, experiments demonstrate that the proposed method outperforms other MARL methods across three main metrics: success rate, safety, and efficiency. This work adopts 5 typical simulated traffic scenarios, which are still far from emulating the complexity of real-world traffic scenes and lack other traffic participants, e.g., pedestrians and traffic lights. Compared with the monocular camera, LiDAR can provide sufficient range information, making agents interact within a safer distance. so these works [120], [124], [125] achieve more generalization results.

Although RL is an appealing way to make the agent learn by trial-and-error in the environment without expert instructions, most RL methods suffer from poor sample efficiency. With the use of neural networks for deep representation learning and function approximation in the domain of RL, interpretability still remains a challenge.

### C. Parallel Learning

Planning methods in autonomous driving are constrained by several challenges. Pipeline planning methods couple numerous human-customized heuristics, which leads to inefficient computation and low generalization. Imitation learning (IL) methods require considerable volume and diverse distribution of expert trajectories, while reinforcement learning (RL) methods demand significant computational resources. Consequently, the presence of these limitations impedes the widespread implementation of autonomous driving.

In response to the various problems in planning methods, virtual-real interaction provides a proven solution [133]. Cyber-physical-systems (CPS) based intelligent control can facilitate interactions and integration between physical and cyberspaces but are not considering human and social factors in systems. In reply, many researchers have added social factors and artificial information to the CPS to form the cyber-physical-social systems (CPSS). In CPSS, the 'C' stands for two dimensions: the information system in the real world and the virtual artificial system defined by software. The 'P' refers to the traditional real system. The 'S' includes not only the human social system but also the artificial system based on the real world.

CPSS enables virtual and real systems to interact, feedback, and promote each other. The real system provides valuable datasets for the construction and calibration of the artificial system, while the artificial system directs and supports the operation of the real system, thus achieving self-evolution. Due to the advantages of virtual-real interaction, CPSS provides a new verification method for end-to-end autonomous driving.

Based on CPSS, Fei-Yue Wang [126] proposes the concept of parallel system theory in 2004, as shown in Fig. 12, the core concept of which is the ACP method, an organic combination of artificial societies (A), computational experiments (C) and parallel execution (P). Over the past two decades, the research system of parallel system theory has been enriched and improved by a large number of implementations in practice [134], such as parallel intelligence [135], parallel control [136], [137], parallel management [138], parallel transportation [139], parallel driving [130], [140], parallel tracking [141], parallel testing [132], parallel vision [129] and so on. The survey about the methods proposed in this section is shown in Table III-C.

In order to further expand the learning capabilities of neural networks, and to address the challenges of IL and RL, Li et al. [128] propose a basic framework for parallel learning based on the parallel system theory as shown in

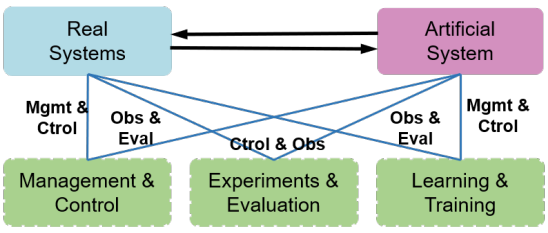



Fig. 12. The framework of parallel system theory proposed in [126].

TABLE II
MAIN APPROACHES FOR MOTION PLANNING IN AUTONOMOUS DRIVING BASED ON DEEP REINFORCEMENT LEARNING.

| Article | Method | Observation | Output | Scenario | Simulator |
|---|---|---|---|---|---|
| Wolf et al. [85] | **Value-based**, DQN | front cam | discrete Steering angle | lane keeping | Gazebo |
| Alizadeh et al. [88] | **Value-based**, DQN | relative distance & velocity value | trajectory points | lane change | Self-made environment |
| Ronecker et al. [89] | **Value-based**, DQN | relative distance & velocity value | trajectory points | lane change, highway strategy | Self-made environment |
| Li et al. [93] | **Value-based**, DQN | front cam | discrete lane change action | lane change & city strategy | CARLA |
| Mo et al. [97] | **Value-based**, DQN | front cam | discrete acceleration & lane change action | overtakeing & highway strategy | SUMO |
| Kendall et al. [98] | **Policy-based**, DDPG | front cam | continuous steering angle & speed setpoint | lane keeping | Unreal Engine 4 |
| Wang et al. [100] | **Policy-based**, DDPG, DQN | front cam | discrete lane change action | lane change | Self-made environment |
| Saxen et al. [101] | **Policy-based**, PPO | lane based grid | continuous acceleration & steering angle | highway kinematic | Open source simulator |
| Ye et al. [103] | **Policy-based**, PPO | relative distance & velocity | discrete lane change action | lane change | SUMO |
| Liang et al. [106] | **Policy-based**, DDPG | front cam, Speed | continuous steering angle, acceleration, braking | navigation | CARLA |
| Tian et al. [107] | **Policy-based**, BC, DDPG | vehicle kinematic | continuous steering angle & vehicle speed | path tracking | Carsim/Simulink |
| Huang et al. [108] | **Policy-based**, BC, AC | BEV images | continuous target speed & discrete lane change action | unprotected left turn, roundabout | SMARTS |
| Wu et al. [109] | **Policy-based**, PHIL-TD3 | BEV semantic graph | continuous steering angle & accelerating | left-turn, congestion | CARLA |
| Chen et al. [111] | **HRL**, AC, DQN | front cam | trajectory points | lane change | TORCS |
| Shi et al. [112] | **HRL**, DQN | relative distance & velocity | discrete lane change action & continuous acceleration | lane change | Self-made environment |
| Li et al. [113] | **HRL**, DQN | scenario state | discrete speed & steering angle | INTERACTION dataset | OpenAI GYM toolkit |
| Duan et al. [114] | **HRL** | policy-specific dynamics | discrete speed & steering increment | lane change | Self-made environment |
| Lu et al. [115] | **HRL**, USP-KLSPI | 14-DOF dynamics | discrete speed & steering action | lane merging | Matlab |
| Gao et al. [116] | **HRL**, DDPG, CNN | BEV perception data, HD-Map | continuous speed & steering angle | navigation | Real-world HD-maps |
| Kaushik et al. [120] | **MARL**, DDPG | vehicle kinematics, LiDAR | continuous speed & steering angle | highway navigation | TORCS |
| Wang et al. [121] | **MARL**, PPO | relative distance & velocity | continuous acceleration | road networks | Flow |
| Zhou et al. [122] | **MARL**, MA2C | relative distance & velocity | discrete acceleration | lane change action | Highway-env |
| Chen et al. [123] | **MARL**, MA2C | relative distance & velocity | discrete acceleration & lane change action | lane merging | Highway-env |
| Han et al. [124] | **MARL**, Reward Reallocation | front cam, LiDAR, vehicle kinematic | discrete lane change action | mixed traffic | CARLA |
| Peng et al [125] | **MARL**, CoPO | continuous ego state & LiDAR | continuous acceleration & steering angle values | multi scenarios | MetaDrive |

Fig. 13. In the action phase, parallel learning [128] follows the RL paradigm, employing state transfer to represent the movement of the model, learning from big data, and storing the learned policy in the state-transition function. Notably, parallel learning capitalizes on computational experimentation to refine the policy. Through feature extraction methods, small knowledge can be applied to specific scenarios or tasks, and used for parallel control. Here, "small" refers to specific and intelligent knowledge for the particular problem, rather than denoting the magnitude of knowledge.

An innovative training approach based on parallel learning [128] presents an alternative solution for problem-solving in fully end-to-end autonomous stacks. As shown in Fig. 14, Wang et al. [142] introduce a parallel driving framework, a unified approach for ITS and IVs. The framework directly bridges expert trajectories and control commands to compute the most optimal policy for specific scenarios. Plenty of expert trajectories are collected from real scenarios, and a neural network is employed to learn all these trajectories, inputs and outputs of this network are destination state and control signals. From the viewpoint of parallel learning, this is a self-labeling process, and the process significantly alleviates the

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2023.3274536



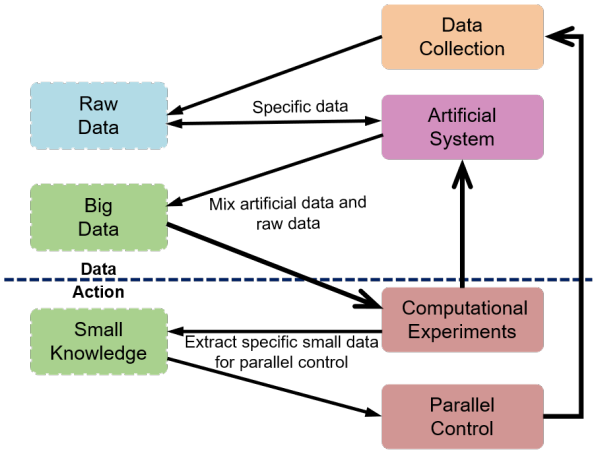



Fig. 13. The theoretical framework of parallel learning proposed in [128]. (The part above the dashed line focuses on big data preprocessing using artificial systems; the part beneath the dashed line focuses on computational experiments. The thin arrows represent either data generation or data learning; the thick arrows present interactions between data and actions.)

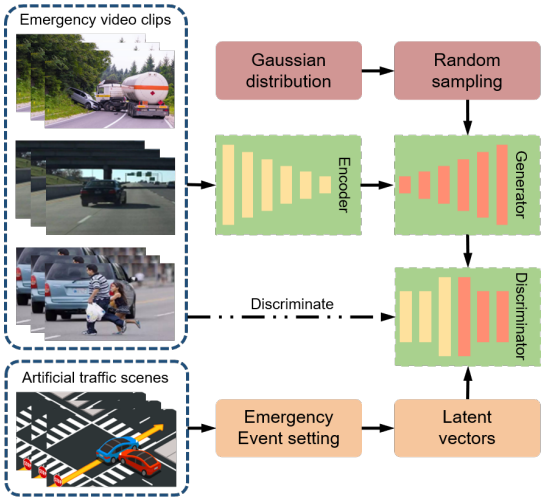



Fig. 15. Hybrid model of combining the variational auto-encoder (VAE) and the generative adversarial network (GAN) for predicting and generating potential emergency image sequences proposed in [131].

data hunger of end-to-end methods.

In order to handle the integrated data from the artificial system and computational experiment, a new theory is proposed, named parallel reinforcement learning (PRL), which combines the parallel learning and deep reinforcement learning approaches. Liu et al. [130] integrate digital quadruplets with

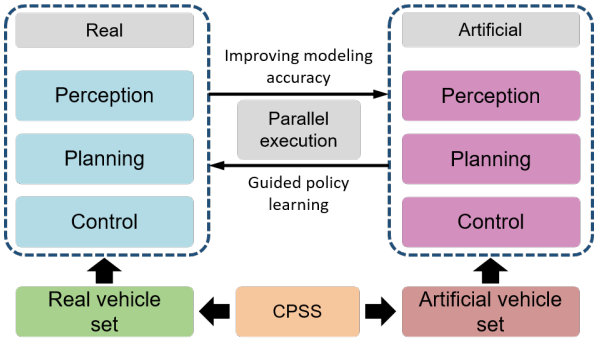



Fig. 14. The theoretical framework of the parallel driving proposed in [142].

parallel driving. This framework defines the physical vehicle, the descriptive vehicle, the predictive vehicle, and the prescriptive vehicle. Based on the description of digital quadruplets, three virtual vehicles can be defined as three "guardian angels" for the physical vehicle, playing different roles to make the IVs safer and more reliable in complex scenarios.

Planning is one of the most significant components of autonomous driving. As a concrete implementation of parallel driving, Chen et al. [130], [142] propose a parallel planning framework for end-to-end planning, which constructs two customized approaches to solve emergency planning problems in specific scenarios. For the data-insufficient problem, parallel planning leverages artificial traffic scenarios to generate expert trajectories based on the pretrained knowledge from reality, as shown in Fig. 15. For the non-robustness problem, parallel planning utilizes a variational auto-encoder (VAE) and a generative adversarial network (GAN) to learn from virtual emergencies generated in artificial traffic scenes. For the learning inefficient problem, parallel planning learning policy from both virtual and real scenarios, and the final decision is

TABLE III
THE SURVEY ABOUT THE PARALLEL SYSTEM THEORY AND ITS SOURCES AND DERIVED ALGORITHMS.

| Method | Year | Detail |
|---|---|---|
| CPS | 1990s | Proposing a multi-dimensional intelligent technology framework, based on big data, internet of things, and large computing, the organic integration and deep collaboration of computing, communication and control (3C). |
| CPSS [127] | 2000 | Integrating social signals and relationships into CPS, leveraging the human, data and information of the social network to break through the various limitations of the real world. |
| Parallel System Theory [126] | 2004 | Integrating artificial societies (A), computational experiments (C) and parallel execution (P), and provide effective tools for control and management of complex systems. |
| Parallel Learning [128] | 2017 | Proposing a new framework of machine learning theory, parallel learning, which incorporates and inherits many elements from various existing machine learning theories. |
| Parallel Vision [129] | 2017 | Introducing the parallel system theory into the computer vision area and constructing a novel research method for perception and understanding of complex driving scenarios. |
| Parallel Driving [130] | 2019 | Constructing an advanced and unified framework for autonomous driving that includes operation management, online condition management and emergency disengagement. |
| Parallel Planning [131] | 2019 | Constructing a deep planning method that integrates a convolutional neural network and a Long short-term memory module to improve the generalization and robustness of planning models in intelligent vehicles. |
| Parallel Testing [132] | 2019 | Proposing a closed-loop testing framework, which implements more challenging scenarios to accelerate evaluation and development of autonomous vehicles. |

TABLE IV
DATASETS AND RELATED DESCRIPTIONS FOR THE AUTONOMOUS DRIVING DATASET.

| Dataset | Year | Sensors | Scenarios |
|---|---|---|---|
| KITTI [143] | 2013 | 4 cameras; 1 LiDAR | City; Countryside; Highway |
| Comma.ai [144] | 2016 | 1 monocular camera; 1 point grey camera | Highway scenarios |
| Oxford RobotCar [145] | 2016 | 6 Cameras; 3 LiDARS; Speed; GPS; INS | City; Contain weather changed |
| Mapillary Vistas [146] | 2017 | Image devices | Street Scenarios |
| nuScenes [147] | 2019 | 6 Cameras; 5 Radars; 1 LiDAR | Street Scenarios |
| ApolloScape [148] | 2019 | 2 Cameras; 2 LiDAR; GPS; IMU | Street Scenarios |
| Waymo Open Dataset [149] | 2019 | 5 Cameras; 5LiDAR; | 1150 Street Scenarios |
| BDD100K [150] | 2020 | 1 Camera; GPS; IMU | Street scenarios in 4 cities |
| A2D2 [151] | 2020 | 6 Cameras; 5 LiDAR; GPS; IMU | 360° Street Scenarios |
| Automine [152] | 2021 | 2 Cameras; 1 LiDAR; GPS; IMU | The first open-pit mine dataset |
| AI4MARS [153] | 2021 | 2 Cameras | The first large-scale dataset in Mars |
| SODA10M [154] | 2021 | 1 Camera | City Scenarios in 31 cities with all kinds of weathers |
| SUPS [155] | 2022 | 6 Cameras; 1 LiDAR; GPS; IMU | Underground parking scenarios |
| DRIVERTRUTH [156] | 2022 | 1 Camera; 1 LiDAR; GPS; IMU; Control signal | City Scenarios based-on CARLA |
| ROAD [157] | 2023 | 1 Camera | Scenarios in [145] for road event detection |

determined by analysis of real observations. Parallel planning is able to make rational decisions without a heavy calculation burden when an emergency occurs.

The parallel system theory provides an effective tool for the control and management of complex systems, especially in the autonomous control field, parallel driving effectively alleviates the shortage of data, inefficient learning, and poor robustness for end-to-end planning models.

## IV. EXPERIMENT PLATFORM

Testing IVs in real systems often comes with potentially fatal safety risks. Therefore, algorithms in autonomous driving are often evaluated in artificial systems with the utilization of open-source datasets and simulation platforms [158].

### A. Dataset

The end-to-end method leverages widely available large-scale datasets of human driving to be trained to approximate human standards. Consequently, the training process requires a large amount of data from driving scenarios. The magnitude, abundance, and distribution of the dataset directly affect the safety, robustness, and generalization of the trained model. Though constructing and assembling novel datasets for IVs is time-consuming, numerous generic and influential datasets are available for research, such as Comma.ai [144], Bdd100K [150], A2D2 [151], Automine [152], DriverTruth [156] and Sups [155], most of the famous dataset is shown in Table. IV.

KITTI [143] is a pioneer in this field and also the most famous autonomous driving dataset. Thanks to its good task scaling, KITTI now covers a wide range of basking perception tasks, such as object detection, sceneflow, depth estimation, tracking and so on.

Comma.ai [144] enriches the diversity of data by additionally collecting localization information and control signals, so it can be implemented for more tasks, for example, localization and planning.

BDD100K [150] and SODA10M [154] alleviate diversity and volume problems by constructing large-scale simulation scenarios, both of them collect several urban scenarios under various weather conditions in more than 31 cities, they also come with a rich set of labels: scene tagging, object bounding box, lane marking, drivable area, full-frame semantic and instance segmentation, multiple object tracking, and multiple objects tracking.

A2D2 [151] is a commercial-grade dataset that is well-suited for diverse perception tasks, bridging the gap between public datasets which are deficient in comprehensive vehicular information. Compared with previous datasets, it provides a 360° point cloud perception field by 5 LiDARs to enable full scene perception for autonomous driving.

The following dataset provides traffic scenarios that are distinct from previous ones. Automine [152] constructs the pioneering open-pit mine dataset for IVs, comprising 18 hours of transportation videos and localization information gathered from 6 open-pit mines. The distinctive features of open-pit mines, such as uneven and rough terrain, intense light, and copious dust, pose significant challenges. The Automine serves as a valuable resource to address the gaps in the open-pit mine dataset, and supports the advancement of autonomous mining technology. AI4MARS [153] proposes another interesting large-scale dataset, which consists of 35,000 semantic segmentation full images of the surface of Mars.

Currently, datasets play a crucial role in training and validating IV methods, supporting the fundamental groundwork necessary for implementing autonomous driving.

### B. Simulation Platform

Testing autonomous driving algorithms in real-world scenarios is often accompanied by significant potential risks, simulation testing shows a smart method to validate algorithms that can speed up testing due to its low cost and high safety.

Many autonomous driving simulation platforms have been developed with open-source code and protocols, which are available for the testing of algorithms in autonomous driving. SUMO [159], an open-source and microscopic traffic simulation platform, developed by the German Aerospace Center, offers a powerful validation platform for large-scale transportation algorithms. It is equipped with a well-designed interface that supports a broad range of data formats. Owing to its superior features, SUMO has been one of the earliest and most extensively utilized simulators. Moreover, Apollo [148] and Autoware [160] not only provide a simulation

TABLE V
SIMULATION PLATFORMS AND RELATED DESCRIPTIONS FOR AUTONOMOUS DRIVING BASED ON VISUAL PERCEPTION.

| Platform | Latest Version | Description |
|---|---|---|
| PTV Vissim | V2023 | Traffic simulation platform focused on complex intersection design and active traffic management. |
| VTD | V2.2 (19.01) | Provides a complete bottom-up simulation platform, including ADAS and automation systems. |
| SUMO [159] | V1.15.0 (22.11) | Provides a purely microscopic traffic model that can be defined to customize each vehicle. |
| TORCS | V1.3.8 (17.03) | Support for running a large number of agents at the same time, allowing for scheduling functions in dense vehicle areas. |
| SVL Simulator [164] | V2021.3 (21.05) | Enables developers to simulate billions of miles and arbitrary corner cases to accelerate algorithm development and system integration. |
| V-Rep | V3.6.2 (19.01) | With a driving actions assessment function, which indicates the agent behavior based on the result. |
| CarMaker | V10.0 (21.10) | Specifically designed for the development and seamless testing of cars and light-duty vehicles in all development stages. |
| CARLA [161] | V0.9.13 (21.11) | Various city maps are provided for autonomous driving algorithms, as well as support for customized sensor types and weather conditions. |
| AriSim [165] | V1.8.1 (22.06) | The capability to quickly complete autonomous driving tests, and build various scenarios (urban, countryside, highway, field, etc.) |
| Apollo [148] | V8.0 (22.12) | Support for learning and validation of single and multi-vehicle autonomous driving algorithms on urban scenarios. |
| Autoware [160] | V1.11.0 (21.05) | An open-source autonomous driving platform, which include all component of autonomous function for intelligent vehicle. |
| Drive Constellation | V6.05 (22.10) | Provides a computing platform based on two different servers that can undertake large-scale vehicle data interaction services. |
| MetaDrive [163] | V0.2.6.0 (22.11) | A wide range of road segments are available, which can be customized to generate a variety of complex scenarios, more suitable for reinforcement learning. |

platform for validating algorithms but also equip open-source algorithms for each task, providing developers with a complete development-validation-deployment chain.

In the context of the ego-vehicle autonomous driving method, CARLA [161] offers a suitable answer. It is an open-source simulator for urban autonomous driving scenarios, which facilitates the development, training, and validation of the underlying urban autonomous driving system.

In the field of the multi-vehicle interaction method, TORCS [162] provides an open racing car simulator with over 50 diverse vehicle models and more than 20 racing tracks. Furthermore, it has the ability to race against 50 vehicles simultaneously, making it a valuable tool for research in this field. MetaDrive [163] proposes an open-source platform to support the research of generalizable reinforcement learning algorithms for machine autonomy. It is highly compositional and capable of generating an infinite number of diverse driving scenarios through both procedural generation and real data importing. The other simulation platforms and their related descriptions are shown in Table. V.

### C. Physical Platform

With the increase in computer computing capabilities, simulation testing has become increasingly capable of meeting the testing requirements for various scenarios and has proven effective in solving the long-tail problem associated with such systems. However, pre-trained models used in a simulator typically require fine-tuning prior to implementation in the real world. Moreover, while simulation testing can cover a wide range of scenarios, it can't account for all corner cases. Consequently, a professional and safe semi-open autonomous driving validation site is essential [ [166].

Autonomous driving technology achieved significant development over the past few decades, and several countries adopt policies permitting the testing of robotic taxis on public roads. In the United States, Waymo is now permitted to test robotaxis on the streets of San Francisco from 2022. Nuro recently begins to deploy autonomous delivery vehicles in Arizona, California, and Texas. In England, Aurrigo is conducting a trial of an autonomous shuttle at Birmingham airport. Wayve is authorized to test autonomous vehicles over long distances between five cities. In China, the commercialization of autonomous driving is rapidly progressing, with companies such as Apollo, Pony, and Momenta already implementing IVs in several cities. Additionally, Waytous is working on unmanned transport in unstructured and closed scenarios and has already provided driverless solutions for several open-pit mines.

## V. CHALLENGES AND FUTURE PERSPECTIVES

Considerable milestones have been achieved in autonomous driving, as evidenced by its successful validation on semi-open roads in various cities. However, its complete commercial deployment is yet to be realized due to numerous obstacles and impending challenges that need to be surmounted.

### A. Challenges

The challenges in IVs are summarised below:

1) Perception: autonomous driving frameworks heavily rely on perception data, however, most sensors are vulnerable to environmental effects and suffer from partial perception issues. As a result, potential hazards may be ignored, and these drawbacks present security challenges for autonomous driving.
2) Planning: both pipeline and end-to-end planning have intrinsic limitations, and ensuring the production of high-quality outputs under uncertain and complex scenarios is an indispensable research objective.
3) Safety: hacking for autonomous driving systems is increasing, even minor disruptions have possibly triggered significant deviations. Therefore, the deployment of autonomous driving methods on a massive scale necessitates robust measures to counter adversarial attacks.

4) Dataset: simulators are essential for training and testing autonomous driving models, however, models well-trained in virtual environments often cannot be directly implemented in reality [167]. Thus, bridging the gap between virtual and real data is imperative for advancing research in this field.

### B. Future Perspectives

The mechanism of the end-to-end planner is the closest to the human driver, according to the input state to calculate the output space. However, due to challenges in data, interpretability, generalization, and policies, end-to-end planners are still scarcely implemented in the real world. Herein, we propose some future perspectives in the field of end-to-end planning.

- Interpretability: Machine learning receives criticism due to its black-box properties. The current intermediate feature representations are insufficient to explain the causality of its inference process. In the case of IV, the consequences of lacking interpretability could be catastrophic. Thus, providing clear and understandable interpretations for the motion planner is crucial in enhancing trust in intelligent vehicles (IVs). Moreover, this approach could assist in predicting and rectifying potential issues that may jeopardize the safety of the passengers.
- Sim2Real: The simulation and the real environment have obvious differences in scenario diversity and environment complexity, making it challenging to align simulation data with real data [168], [169]. Consequently, the well-trained models in simulators may not optimally perform in real settings. Developing a model to bridge the gap between simulated and real environments is critical to address the challenges about data diversity and fairness, which is also a crucial research direction in end-to-end planning.
- Reliability: One critical bottleneck that impedes the development and deployment of IVs is the prohibitively high economic and time costs required to validate their reliability. Constructing an artificial-intelligence-based algorithm that can identify the corner cases in a short time is a key direction for the validation of IVs.
- Governance: IV is not only a technical issue, the sound policy is also crucial. Designing a framework that includes safety standards, data privacy regulations, and ethical guidelines is necessary to govern the development and deployment of IVs. This framework will promote accountability and transparency, reduce risks, and ensure that the public interest is defended.

## REFERENCES


[1] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[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 Transactions on Intelligent Vehicles*, pp. 1–13, 2022.

[3] W. Wang, L. Wang, C. Zhang, C. Liu, and L. Sun, "Social interactions for autonomous driving: A review and perspectives," *Foundations and Trends® in Robotics*, vol. 10, no. 3-4, pp. 198–376, 2022. [Online]. Available: http://dx.doi.org/10.1561/2300000078

[4] L. Chen, Y. Zhang, B. Tian, D. Cao, and F.-Y. Wang, "Parallel driving os: A ubiquitous cyber-physical-socialsystem-based operating system for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2022.

[5] R. Song, Y. Ai, B. Tian, L. Chen, F. Zhu, and Y. Fei, "Msfanet: A light weight object detector based on context aggregation and attention mechanism for autonomous mining truck," *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2022.

[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 Transactions on Intelligent Vehicles*, pp. 1–11, 2022.

[7] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.

[8] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.

[9] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. de Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2020.

[10] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[11] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[12] A. Charnes and W. M. Raike, "One-pass algorithms for some generalized network problems," *Operations Research*, vol. 14, no. 5, pp. 914–924, 1966.

[13] M. Lotfi, G. J. Osório, M. S. Javadi, A. Ashraf, M. Zahran, G. Samih, and J. P. S. Catalão, "A dijkstra-inspired graph algorithm for fully autonomous tasking in industrial applications," *IEEE Transactions on Industry Applications*, vol. 57, no. 5, pp. 5448–5460, 2021.

[14] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[15] N. J. Nilsson, "A mobile automaton: An application of artificial intelligence techniques," Sri International Menlo Park Ca Artificial Intelligence Center, Tech. Rep., 1969.

[16] B. Li, Z. Yin, Y. Ouyang, Y. Zhang, X. Zhong, and S. Tang, "Online trajectory replanning for sudden environmental changes during automated parking: A parallel stitching method," *IEEE Transactions on Intelligent Vehicles*, 2022.

[17] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, and C. Hu, "A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1376–1386, 2019.

[18] L. Chen, Y. Shan, W. Tian, B. Li, and D. Cao, "A fast and efficient double-tree rrt*-like sampling-based planner applying on mobile robotic systems," *IEEE/ASME transactions on mechatronics*, vol. 23, no. 6, pp. 2568–2578, 2018.

[19] F. Gao, W. Wu, J. Pan, B. Zhou, and S. Shen, "Optimal time allocation for quadrotor trajectory generation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4715–4722.

[20] B. Li, T. Acarman, Y. Zhang, Y. Ouyang, C. Yaman, Q. Kong, X. Zhong, and X. Peng, "Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[21] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.

[22] W. Xu, J. Pan, J. Wei, and J. M. Dolan, "Motion planning under uncertainty for on-road autonomous driving," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2507–2512.

[23] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

[24] S. M. Bagheri, H. Taghaddos, A. Mousaei, F. Shahnavaz, and U. Hermann, "An a-star algorithm for semi-optimization of crane location and

configuration in modular construction," *Automation in Construction*, vol. 121, p. 103447, 2021.

[25] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," *IEEE/ASME Transactions on mechatronics*, vol. 21, no. 2, pp. 740–753, 2015.

[26] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online pomdp planning for autonomous driving in a crowd," in *2015 ieee international conference on robotics and automation (icra)*. IEEE, 2015, pp. 454–460.

[27] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "Despot: Online pomdp planning with regularization," *Advances in neural information processing systems*, vol. 26, 2013.

[28] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1: 43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[29] B. Li, Y. Ouyang, L. Li, and Y. Zhang, "Autonomous driving on curvy roads without reliance on frenet frame: A cartesian-based trajectory planning method," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[30] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu apollo em motion planner," *arXiv preprint arXiv:1807.08048*, 2018.

[31] P. Scheffe, T. M. Henneken, M. Kloock, and B. Alrifaee, "Sequential convex programming methods for real-time optimal trajectory planning in autonomous vehicle racing," *IEEE Transactions on Intelligent Vehicles*, 2022.

[32] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 613–626, 2018.

[33] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[34] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific journal of mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[35] R. Bai and H.-B. Wang, "Robust optimal control for the vehicle suspension system with uncertainties," *IEEE Transactions on Cybernetics*, 2021.

[36] 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," *Mechanical systems and signal processing*, vol. 100, pp. 482–500, 2018.

[37] A. Botros and S. L. Smith, "Tunable trajectory planner using g3 curves," *IEEE Transactions on Intelligent Vehicles*, 2022.

[38] J. Hu, Y. Zhang, and S. Rakheja, "Adaptive lane change trajectory planning scheme for autonomous vehicles under various road frictions and vehicle speeds," *IEEE Transactions on Intelligent Vehicles*, 2022.

[39] Y. Guo, D. D. Yao, B. Li, H. Gao, and L. Li, "Down-sized initialization for optimization-based unstructured trajectory planning by only optimizing critical variables," *IEEE Transactions on Intelligent Vehicles*, 2022.

[40] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[41] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4889–4895.

[42] F. Tian, R. Zhou, Z. Li, L. Li, Y. Gao, D. Cao, and L. Chen, "Trajectory planning for autonomous mining trucks considering terrain constraints," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 772–786, 2021.

[43] B. Li, Y. Ouyang, X. Li, D. Cao, T. Zhang, and Y. Wang, "Mixed-integer and conditional trajectory planning for an autonomous mining truck in loading/dumping scenarios: A global optimization approach," *IEEE Transactions on Intelligent Vehicles*, 2022.

[44] Z. Zhang, R. Tian, R. Sherony, J. Domeyer, and Z. Ding, "Attention-based interrelation modeling for explainable automated driving," *IEEE Transactions on Intelligent Vehicles*, pp. 1–10, 2022.

[45] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[46] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4693–4700.

[47] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2722–2730.

[48] A. Sauer, N. Savinov, and A. Geiger, "Conditional affordance learning for driving in urban environments," in *Conference on Robot Learning*. PMLR, 2018, pp. 237–252.

[49] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8660–8669.

[50] 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 *European Conference on Computer Vision*. Springer, 2020, pp. 414–430.

[51] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

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

[53] C. Yan, J. Qin, Q. Liu, Q. Ma, and Y. Kang, "Mapless navigation with safety-enhanced imitation learning," *IEEE Transactions on Industrial Electronics*, pp. 1–9, 2022.

[54] G. Li, M. Mueller, V. Casser, N. Smith, D. L. Michels, and B. Ghanem, "Oil: Observational imitation learning," *arXiv preprint arXiv:1803.01129*, 2018.

[55] E. Ohn-Bar, A. Prakash, A. Behl, K. Chitta, and A. Geiger, "Learning situational driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 296–11 305.

[56] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with gaussian processes," *Advances in neural information processing systems*, vol. 24, 2011.

[57] D. Brown and S. Niekum, "Efficient probabilistic performance bounds for inverse reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

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

[59] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[60] K. Lee, D. Isele, E. A. Theodorou, and S. Bae, "Spatiotemporal costmap inference for mpc via deep inverse reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3194–3201, 2022.

[61] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[62] T. Phan-Minh, F. Howington, T.-S. Chu, S. U. Lee, M. S. Tomov, N. Li, C. Dicle, S. Findler, F. Suarez-Ruiz, R. Beaudoin *et al.*, "Driving in real life with inverse reinforcement learning," *arXiv preprint arXiv:2206.03004*, 2022.

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

[64] Z. Zhu and H. Zhao, "Multi-task conditional imitation learning for autonomous navigation at crowded intersections," *IEEE Transactions on Intelligent Vehicle*, 2023.

[65] Q. Wang, L. Chen, B. Tian, W. Tian, L. Li, and D. Cao, "End-to-end autonomous driving: An angle branched network approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11 599–11 610, 2019.

[66] M. Peng, Z. Gong, C. Sun, L. Chen, and D. Cao, "Imitative reinforcement learning fusing vision and pure pursuit for self-driving," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3298–3304.

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

[68] X. Hu, B. Tang, L. Chen, S. Song, and X. Tong, "Learning a deep cascaded neural network for multiple motion commands prediction in

autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7585–7596, 2020.

[69] H. He, J. Eisner, and H. Daume, "Imitation learning by coaching," *Advances in neural information processing systems*, vol. 25, 2012.

[70] 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," *arXiv preprint arXiv:2109.08273*, 2021.

[71] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[72] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[73] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," *Advances in neural information processing systems*, vol. 20, 2007.

[74] M. Valko, M. Ghavamzadeh, and A. Lazaric, "Semi-supervised apprenticeship learning," in *European workshop on reinforcement learning*. PMLR, 2013, pp. 131–142.

[75] B. Woodworth, F. Ferrari, T. E. Zosa, and L. D. Riek, "Preference learning in assistive robotics: Observational repeated inverse reinforcement learning," in *Machine learning for healthcare conference*. PMLR, 2018, pp. 420–439.

[76] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," in *IJCAI*, vol. 7, 2007, pp. 2586–2591.

[77] M. Wulfmeier, D. Z. Wang, and I. Posner, "Watch this: Scalable cost-function learning for path planning in urban environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2089–2095.

[78] Y. Li, J. Song, and S. Ermon, "Infogail: Interpretable imitation learning from visual demonstrations," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

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

[80] 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 *Conference on Robot Learning*. PMLR, 2022, pp. 1279–1290.

[81] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction, 2nd ed*. Cambridge, MA, US: The MIT Press, 2018.

[82] L. Yue and H. Fan, "Dynamic scheduling and path planning of automated guided vehicles in automatic container terminal," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 11, pp. 2005–2019, 2022.

[83] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

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

[85] 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 *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 244–250.

[86] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[87] L. Chen, X. Hu, B. Tang, and Y. Cheng, "Conditional dqn-based motion planning with fuzzy logic for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 2966–2977, 2020.

[88] 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 *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1399–1404.

[89] M. P. Ronecker and Y. Zhu, "Deep q-network based decision making for autonomous driving," in *2019 3rd International Conference on Robotics and Automation Sciences (ICRAS)*, 2019, pp. 154–160.

[90] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning*. PMLR, 2017, pp. 22–31.

[91] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," *arXiv preprint arXiv:1910.01708*, vol. 7, no. 1, p. 2, 2019.

[92] E. Marchesini, D. Corsi, and A. Farinelli, "Benchmarking safe deep reinforcement learning in aquatic navigation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5590–5595.

[93] 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," *Transportation research part C: emerging technologies*, vol. 134, p. 103452, 2022.

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

[95] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica D: nonlinear phenomena*, vol. 16, no. 3, pp. 285–317, 1985.

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

[97] S. Mo, X. Pei, and C. Wu, "Safe reinforcement learning for autonomous vehicle using monte carlo tree search," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6766–6773, 2022.

[98] 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 *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8248–8254.

[99] 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," *arXiv preprint arXiv:1509.02971*, 2015.

[100] G. Wang, J. Hu, Z. Li, and L. Li, "Harmonious lane changing via deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4642–4650, 2022.

[101] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, "Driving in dense traffic with model-free reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5385–5392.

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

[103] 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 *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1746–1752.

[104] 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 Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, 2020.

[105] Y. Wu, S. Liao, X. Liu, Z. Li, and R. Lu, "Deep reinforcement learning on autonomous driving policy with auxiliary critic network," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2021.

[106] X. Liang, T. Wang, L. Yang, and E. Xing, "Cirl: Controllable imitative reinforcement learning for vision-based self-driving," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 584–599.

[107] 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 Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6357–6367, 2022.

[108] Z. Huang, J. Wu, and C. Lv, "Efficient deep reinforcement learning with imitative expert priors for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2022.

[109] J. Wu, Z. Huang, W. Huang, and C. Lv, "Prioritized experience-based reinforcement learning with human guidance for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[110] 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 Journal of Automatica Sinica*, vol. 9, no. 12, pp. 2161–2173, 2022.

[111] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 1–9.

[112] 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," *arXiv preprint arXiv:1904.10171*, 2019.

[113] J. Li, L. Sun, J. Chen, M. Tomizuka, and W. Zhan, "A safe hierarchical planning framework for complex driving scenarios based on reinforce-

ment learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2660–2666.

[114] 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 Intelligent Transport Systems*, vol. 14, no. 5, pp. 297–305, 2020.

[115] 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. 209 776–209 789, 2020.

[116] 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 *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 143–13 150.

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

[118] 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 Journal of Automatica Sinica*, vol. 10, no. 3, pp. 1–16, 2023.

[119] R. Xu, W. Chen, H. Xiang, L. Liu, and J. Ma, "Model-agnostic multi-agent perception framework," *arXiv e-prints*, pp. arXiv–2203, 2022.

[120] M. Kaushik, N. Singhania, P. S., and K. M. Krishna, "Parameter sharing reinforcement learning architecture for multi agent driving," in *Proceedings of the Advances in Robotics 2019*, ser. AIR 2019. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3352593.3352625

[121] J. Wang, T. Shi, Y. Wu, L. Miranda-Moreno, and L. Sun, "Multi-agent graph reinforcement learning for connected automated driving," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020, pp. 1–6.

[122] 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," *Autonomous Intelligent Systems*, vol. 2, no. 1, p. 5, 2022.

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

[124] 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 *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8765–8771.

[125] Z. Peng, Q. Li, K. M. Hui, C. Liu, and B. Zhou, "Learning to simulate self-driven particles system with coordinated policy optimization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 784–10 797, 2021.

[126] F.-Y. Wang, "Parallel system methods for management and control of complex systems," *Control and Decision*, vol. 19, no. 5, pp. 485–491, 2004.

[127] X. Wang, J. Yang, J. Han, W. Wang, and F.-Y. Wang, "Metaverses and demetaverses: From digital twins in cps to parallel intelligence in cpss," *IEEE Intelligent Systems*, vol. 37, no. 4, pp. 97–102, 2022.

[128] L. Li, L. Yilun, C. Dongpu, Z. Nanning, and W. Fei-Yue, "Parallel learning — a new framework for machine learning," *ACTA AUTOMATICA SINICA*, vol. 43, no. 1, pp. 1–8, 2017.

[129] K. Wang, C. Gou, N. Zheng, J. M. Rehg, and F.-Y. Wang, "Parallel vision for perception and understanding of complex scenes: methods, framework, and perspectives," *Artificial Intelligence Review*, vol. 48, no. 3, pp. 299–329, 2017.

[130] L. Teng, W. Xiao, X. Yang, G. Yu, T. Bin, and C. Long, "Research on digital quadruplets in cyber-physical-social space-based parallel driving," *Chinese Journal of Intelligent Science and Technology*, vol. 1, no. 1, pp. 485–491, March 2019.

[131] L. Chen, X. Hu, B. Tang, and D. Cao, "Parallel motion planning: Learning a deep planning model against emergencies," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 1, pp. 36–41, 2018.

[132] L. Li, X. Wang, K. Wang, Y. Lin, J. Xin, L. Chen, L. Xu, B. Tian, Y. Ai, J. Wang *et al.*, "Parallel testing of vehicle intelligence via virtual-real interaction," *Science robotics*, vol. 4, no. 28, p. eaaw4106, 2019.

[133] J. Yang, X. Wang, and Y. Zhao, "Parallel manufacturing for industrial metaverses: A new paradigm in smart manufacturing," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 12, pp. 2063–2070, 2022.

[134] K. Liu, L. Li, Y. Lv, D. Cao, Z. Liu, and L. Chen, "Parallel intelligence for smart mobility in cyberphysical social system-defined metaverses: A report on the international parallel driving alliance," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 18–25, 2022.

[135] X. Wang, L. Li, Y. Yuan, P. Ye, and F.-Y. Wang, "Acp-based social computing and parallel intelligence: Societies 5.0 and beyond," *CAAI Transactions on Intelligence Technology*, vol. 1, no. 4, pp. 377–393, 2016.

[136] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630–638, 2010.

[137] J. Lu, Q. Wei, T. Zhou, Z. Wang, and F.-Y. Wang, "Event-triggered near-optimal control for unknown discrete-time nonlinear systems using parallel control," *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 1890–1904, 2023.

[138] J. Lu, X. Wang, X. Cheng, J. Yang, O. Kwan, and X. Wang, "Parallel factories for smart industrial operations: From big ai models to field foundational models and scenarios engineering," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 12, pp. 2079–2086, 2022.

[139] F. Zhu, Y. Lv, Y. Chen, X. Wang, G. Xiong, and F.-Y. Wang, "Parallel transportation systems: Toward iot-enabled smart urban traffic control and management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4063–4071, 2019.

[140] J. Yang, X. Wang, and Y. Zhao, "Parallel manufacturing for industrial metaverses: A new paradigm in smart manufacturing," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 12, pp. 2063–2070, 2022.

[141] J. Lu, Q. Wei, Y. Liu, T. Zhou, and F.-Y. Wang, "Event-triggered optimal parallel tracking control for discrete-time nonlinear systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 6, pp. 3772–3784, 2022.

[142] F.-Y. Wang, N.-N. Zheng, D. Cao, C. M. Martinez, L. Li, and T. Liu, "Parallel driving in cpss: A unified approach for transport automation and vehicle intelligence," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 577–587, 2017.

[143] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

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

[145] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[146] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4990–4999.

[147] 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 multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[148] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 954–960.

[149] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[150] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[151] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn *et al.*, "A2d2: Audi autonomous driving dataset," *arXiv preprint arXiv:2004.06320*, 2020.

[152] Y. Li, Z. Li, S. Teng, Y. Zhang, Y. Zhou, Y. Zhu, D. Cao, B. Tian, Y. Ai, Z. Xuanyuan *et al.*, "Automine: An unmanned mine dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 308–21 317.

[153] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.

[154] J. Han, X. Liang, H. Xu, K. Chen, L. Hong, J. Mao, C. Ye, W. Zhang, Z. Li, X. Liang, and C. Xu, "Soda10m: A large-scale 2d self/semi-supervised object detection dataset for autonomous driving," 2021.

[155] J. Hou, Q. Chen, Y. Cheng, G. Chen, X. Xue, T. Zeng, and J. Pu, "Sups: A simulated underground parking scenario dataset for autonomous

driving," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 2265–2271.
[156] R. Muller, Y. Man, Z. B. Celik, M. Li, and R. Gerdes, "Drivetruth: Automated autonomous driving dataset generation for security applications," in *International Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2022.
[157] G. Singh, S. Akrigg, M. D. Maio, V. Fontana, R. J. Alitappeh, S. Khan, S. Saha, K. Jeddisaravi, F. Yousefi, J. Culley, T. Nicholson, J. Omokeowa, S. Grazioso, A. Bradley, G. D. Gironimo, and F. Cuzzolin, "Road: The road event awareness dataset for autonomous driving," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1036–1054, 2023.
[158] R. Xu, H. Xiang, X. Han, X. Xia, Z. Meng, C.-J. Chen, C. Correa-Jullian, and J. Ma, "The opencda open-source ecosystem for cooperative driving automation research," *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2023.
[159] D. Krajzewicz, "Traffic simulation with sumo–simulation of urban mobility," in *Fundamentals of traffic simulation*. Springer, 2010, pp. 269–293.
[160] 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 *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, 2018, pp. 287–296.
[161] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.
[162] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software available at http://torcs. sourceforge. net*, vol. 4, no. 6, p. 2, 2000.
[163] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE transactions on pattern analysis and machine intelligence*, 2022.
[164] 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 *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.
[165] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*. Springer, 2018, pp. 621–635.
[166] B. Li, L. Fan, Y. Ouyang, S. Tang, X. Wang, D. Cao, and F.-Y. Wang, "Online competition of trajectory planning for automated parking: Benchmarks, achievements, learned lessons, and future perspectives," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 16–21, 2023.
[167] L. Chen, Q. Wang, X. Lu, D. Cao, and F.-Y. Wang, "Learning driving models from parallel end-to-end driving data set," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 262–273, 2019.
[168] 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 Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2072–2084, 2019.
[169] 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 Intelligent Systems*, vol. 37, no. 4, pp. 18–26, 2022.

**Siyu Teng** received M.S. degree from Jilin University in 2021. Now he is a PhD Student at Department of Computer Science, Hong Kong Baptist University. His main interests are parallel planning, end-to-end autonomous driving and interpretable deep learning.

**Xuemin Hu** is currently an Associate Professor with School of Artificial Intelligence, Hubei University, Wuhan, China. He received the B.S. degree from Huazhong University of Science and Technology and the Ph.D. degree from Wuhan University in 2007 and in 2012, respectively. He was a visiting scholar in the University of Rhode Island, Kingston, RI, US from November 2015 to May 2016. His areas of interest include computer vision, machine learning, motion planning, and autonomous driving.

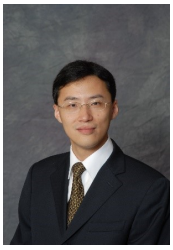
**Peng Deng** received the B.E. degree in vehicle engineering from China Agricultural University, Beijing, China. He is currently pursuing the M.S. degree with the School of Artificial Intelligence, Hubei University, Wuhan, China. His areas of interest include reinforcement learning and autonomous driving.

**Bai Li** (SM'13–M'18) received his B.S. degree in 2013 from the School of Advanced Engineering, Beihang University, China, and his Ph.D. degree in 2018 from the College of Control Science and Engineering, Zhejiang University, China. From Nov. 2016 to June 2017, he visited the Department of Civil and Environmental Engineering, University of Michigan (Ann Arbor), USA, as a joint training Ph.D. student. He is currently an associate professor in Hunan University. Before teaching at Hunan University, he worked in JDX R&D Center of Automated Driving, JD Inc., China from 2018 to 2020 as an algorithm engineer. Prof. Li has been the first author of more than 70 journal/conference papers and two books related to numerical optimization, motion planning, and robotics. He was a recipient of the International Federation of Automatic Control (IFAC) 2014–2016 Best Journal Paper Prize from Engineering Applications of Artificial Intelligence. He is currently an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT VEHICLES. He was a recipient of the 2022 TIV Best Associate Editor Award. His research interest is rule-based motion planning methods for IVs.

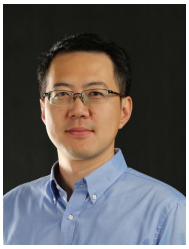
**Yuchen Li** received the B.E. degree from the University of Science and Technology Beijing in 2016, and the M.E. degrees from Beihang University in 2020. He is pursuing the Ph.D. degree in Hong Kong Baptist University. He is an intern at Waytous. His research interest covers computer vision, 3D object detection, and autonomous driving.

**Yunfeng Ai** received the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China in 2006. He is Associate Professor at University of Chinese Academy of Sciences. He was a research fellow at Carnegie Mellon University. His current research interest covers computer vision, machine learning, robots, and autonomous driving.

**Dongsheng Yang** received the Ph.D. degree in information system engineering from the National University of Defense Technology, Changsha, China, in 2004. He is currently a Professor with the School of Public Management/Emergency Management (The Laboratory for Military- Civilian Integration Emergency Command and Control), Jinan University, Guangzhou, China. His research interests include intelligent emergency response of complex systems, multiscale emergency command and control mode and mechanism, and parallel intelligent technology of emergency management.



**Lingxi Li** (S'04-M'08-SM'13) is currently a full professor in the Department of Electrical and Computer Engineering at Purdue School of Engineering and Technology, Indiana University-Purdue University Indianapolis (IUPUI), USA. Dr. Li received his Ph.D. degree in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2008. Dr. Li's current research focuses on modeling, analysis, control, and optimization of complex systems, connected and automated vehicles, intelligent transportation systems, digital twins and parallel intelligence, and human-machine interaction. He has authored/co-authored one book and over 130 research articles in refereed journals and conferences. Dr. Li was the recipient of five best paper awards, 2021 IEEE ITSS outstanding application award, 2017 outstanding research contributions award, 2012 T-ITS outstanding editorial service award, and several university research/teaching awards. He is currently serving as an associate editor for five international journals and has served as the General Chair, Program Chair, Program Co-Chair, etc., for 20+ international conferences.



**Zhe XuanYuan** received the B.S. degree in electronic engineering from Peking University, Beijing, China, in 2005, and the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2012. He is now an Associate Professor of Data Science with Beijing Normal University-Hong Kong Baptist University United International College, Zhuhai, China. His research interests include robot mapping and navigation, autonomous driving, and vehicular networks.

**Fenghua Zhu** (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2008. He is currently an Associate Professor with the State Key Laboratory of Multimodal Artificial Intelligence Systems, China. His research interests include artificial transportation systems and parallel transportation management systems.

**Long Chen** (Senior Member, IEEE) received the Ph.D. degree from Wuhan University in 2013, he is currently a Professor with State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include autonomous driving, robotics, and artificial intelligence, where he has contributed more than 100 publications. He serves as an Associate Editor for the IEEE Transaction on Intelligent Transportation Systems, the IEEE/CAA Journal of Automatica Sinica, the IEEE Transaction on Intelligent Vehicle and the IEEE Technical Committee on Cyber-Physical Systems.