End-to-end Autonomous Driving: Challenges and Frontiers

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger and Hongyang Li

Abstract—The autonomous driving community has witnessed a rapid growth in approaches that embrace an end-to-end algorithm framework, utilizing raw sensor input to generate vehicle motion plans, instead of concentrating on individual tasks such as detection and motion prediction. End-to-end systems, in comparison to modular pipelines, benefit from joint feature optimization for perception and planning. This field has flourished due to the availability of large-scale datasets, closed-loop evaluation, and the increasing need for autonomous driving algorithms to perform effectively in challenging scenarios. In this survey, we provide a comprehensive analysis of more than 270 papers, covering the motivation, roadmap, methodology, challenges, and future trends in end-to-end autonomous driving. We delve into several critical challenges, including multi-modality, interpretability, causal confusion, robustness, and world models, amongst others. Additionally, we discuss current advancements in foundation models and visual pre-training, as well as how to incorporate these techniques within the end-to-end driving framework. We maintain an active repository that contains up-to-date literature and open-source projects at https://github.com/OpenDriveLab/End-to-end-Autonomous-Driving.

Index Terms—Autonomous Driving, End-to-end System Design, Policy Learning, Simulation.

1 Introduction

Conventional autonomous driving systems adopt a modular design strategy, wherein each functionality, such as perception, prediction, and planning, is individually developed and integrated into onboard vehicles. The planning or control module, responsible for generating steering and acceleration outputs, plays a crucial role in determining the driving experience. The most common approach for planning in modular pipelines involves using sophisticated rule-based designs, which are often ineffective in addressing the vast number of situations that occur on road. Therefore, there is a growing trend to leverage large-scale data and to use learning-based planning as a viable alternative.

We define end-to-end autonomous driving systems as fully differentiable programs that take raw sensor data as input and produce a plan and/or low-level control actions as output. Fig. 1 (a)-(b) illustrates the difference between the classical and end-to-end formulation. The conventional approach feeds the output of each component, such as bounding boxes and vehicle trajectories, directly into subsequent units (dashed arrows). In contrast, the end-to-end paradigm propagates feature representations across components (gray solid arrow). The optimized function is set to be, for example, the planning performance, and the loss is minimized via back-propagation (red arrow). Tasks are jointly and globally optimized in this process.

In this survey, we conduct an extensive review of this emerging topic. Fig. 1 provides an overview of our work. We begin by discussing the motivation and roadmap for end-to-end autonomous driving systems. End-to-end approaches can be broadly classified into imitation and reinforcement learning, and we give a brief review of these methodologies. We cover datasets and benchmarks for both closed and open-loop evaluation. We summarize a series of critical challenges, including interpretability, generalization, world models, causal confusion, *etc*. We conclude by discussing future trends that we think should be embraced by the community to incorporate the latest developments from data engines, and large foundation models, amongst others. Note that this review is mainly orchestrated from a theoretical perspective. Engineering efforts such as version control, unit testing, data servers, data cleaning, software-hardware co-design, *etc*., play crucial roles in deploying the end-to-end technology. Publicly available information regarding the latest practices on these topics is limited. We invite the community towards more openness in future discussions.

1.1 Motivation of an End-to-end System

In the classical pipeline, each model serves a standalone component and corresponds to a specific task (*e.g.*, traffic light detection). Such a design is beneficial in terms of interpretability and ease of debugging. However, since the optimization objectives across modules are different, with detection pursuing mean average precision (mAP) while planning aiming for driving safety and comfort, the entire system may not be aligned with a unified target, *i.e.*, the ultimate planning/control task. Errors from each module, as the sequential procedure proceeds, could be compounded and result in an information loss. Moreover, compared to one end-to-end neural network, the multi-task, multi-model deployment which involves multiple encoders and message transmission systems, may increase the computational burden and potentially lead to sub-optimal use of compute.

In contrast to its classical counterpart, an end-to-end autonomous system offers several advantages. (a) The most apparent merit is its simplicity in combining perception,

- *L. Chen and H. Li are with OpenDriveLab, Shanghai AI Lab, Shanghai, China, and the University of Hong Kong, Hong Kong, China. P. Wu is with OpenDriveLab, Shanghai AI Lab, Shanghai, China. K. Chitta, B. Jaeger and A. Geiger are with University of Tübingen and Tübingen AI Center, Germany. Primary contact:* lihongyang@pjlab.org.cn

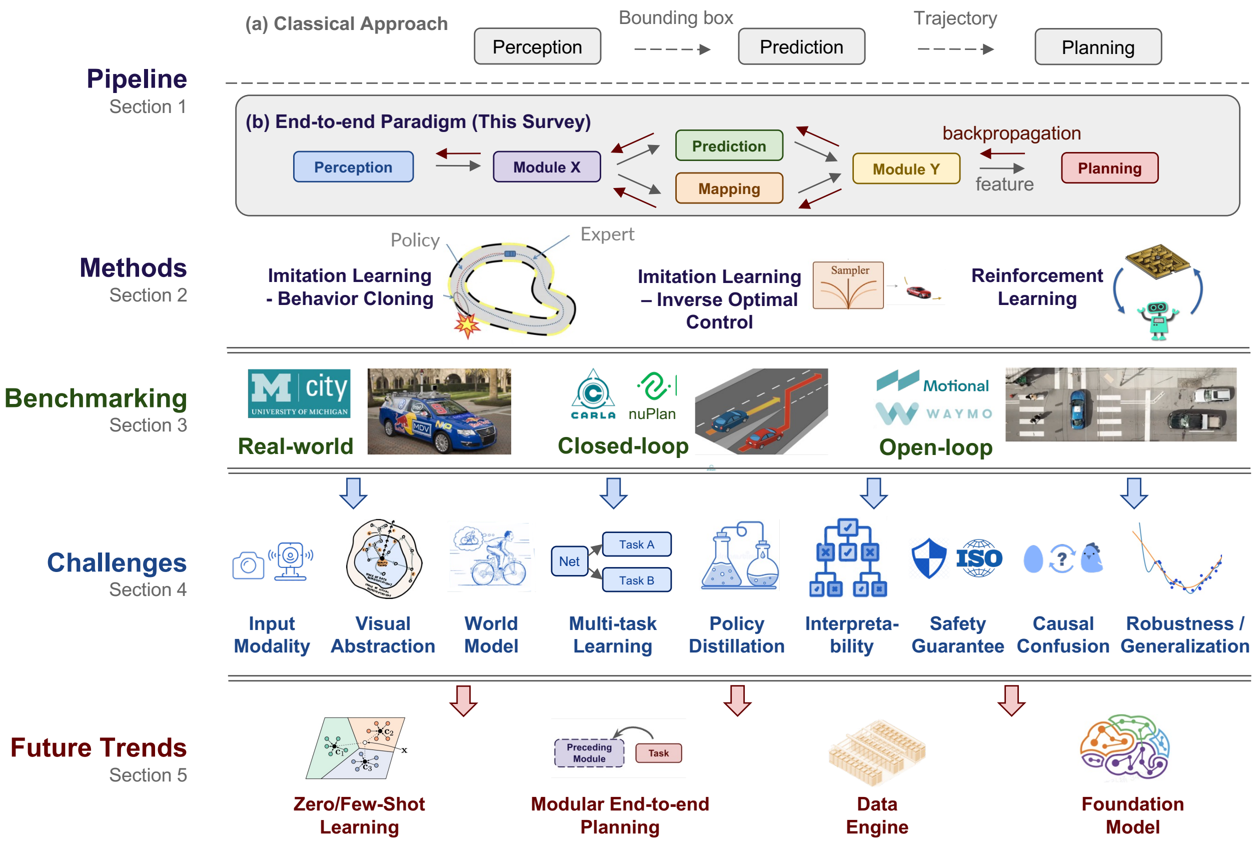

Fig. 1: **Survey at A Glance. (a) Pipeline and Methods.** We define end-to-end autonomous driving as a learning-based algorithm framework with raw sensor input and planning/control output. We deepdive into 270+ papers and categorize into imitation learning (IL) and reinforcement learning (RL). **(b) Benchmarking.** We group popular benchmarks into closed-loop and open-loop evaluation, respectively. We cover various aspects of closed-loop simulation and the limitations of open-loop evaluation for this problem. **(c) Challenges.** This is the main section of our work. We list key challenges from a wide range of topics and extensively analyze why these concerns are crucial. Promising resolutions to these challenges are covered as well. **(d) Future Trends.** We discuss how end-to-end paradigm could benefit by aid of the rapid development of foundation models, visual pre-training, *etc.* Partial photos by courtesy of online resources.

prediction, and planning into a single model that can be jointly trained. (b) The whole system, including its intermediate representations, is optimized towards the ultimate task. (c) Shared backbones increase computational efficiency. (d) Data-driven optimization has the potential to improve the system by simply scaling training resources.

Note that the end-to-end paradigm does not necessarily indicate one black box with only planning/control outputs. It could have intermediate representations and outputs (Fig. 1 (b)) as in classical approaches. In fact, several state-of-the-art systems [1, 2] propose a modular design but optimize all components together to achieve superior performance.

1.2 Roadmap

Fig. 2 depicts a chronological roadmap of critical achievements in end-to-end autonomous driving, where each part indicates an essential paradigm shift or performance boost. The history of end-to-end autonomous driving dates back to 1988 with ALVINN [3], where the input was two "retinas" from a camera and a laser range finder, and a simple neural network generated steering output. NVIDIA designed a prototype end-to-end CNN system, which reestablished this idea in the new era of GPU computing [8]. Notable progress has been achieved with the development of deep neural networks, both in imitation learning [15, 16] and reinforcement learning [4, 17, 18, 19]. The policy distillation paradigm proposed in LBC [5] and related approaches [20, 21, 22, 23] has significantly improved closed-loop performance by mimicking a well-behaved expert. To enhance generalization ability due to the discrepancy between the expert and learned policy, several papers [10, 24, 25] have proposed aggregating on-policy data [26] during training.

A significant turning point occurred around 2021. With diverse sensor configurations available within a reasonable computational budget, attention was focused on incorporating more modalities and advanced architectures (*e.g.*, Transformers [27]) to capture global context and representative features, as in TransFuser [6, 28] and many variants [29, 30, 31]. Combined with more insights about the simulation environment, these advanced designs resulted in a substantial performance boost on the CARLA benchmark [13]. To improve the interpretability and safety of autonomous systems, approaches [11, 32, 33] explicitly involve various auxiliary modules to better supervise the learning process or utilize attention visualization. Recent works prioritize generating safety-critical data [7, 34, 35], pre-training a

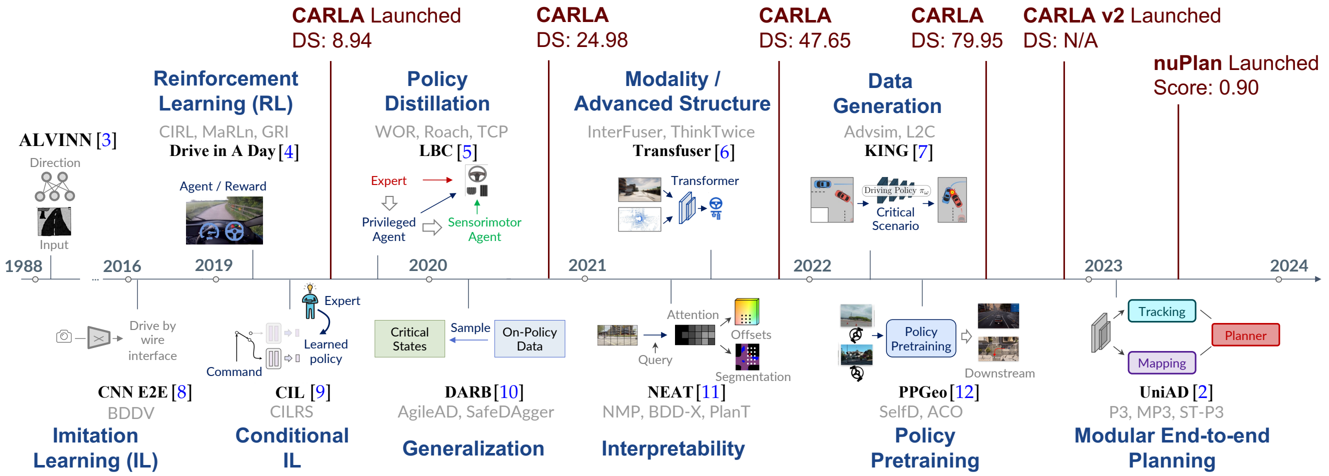

Fig. 2: **Roadmap of End-to-end Autonomous Driving.** We present the key milestones chronologically, grouping similar works under the same theme. The representative or first work is shown in bold with an illustration, while the date of the rest of the literature in the same theme may vary. We also display the score for each year's top entry in the CARLA leaderboard [13] (DS, ranging from 0 to 100) and the recent nuPlan challenge [14] (Score ranging from 0 to 1).

foundation model or backbone curated for policy learning [12, 36, 37], and advocating a modular end-to-end planning philosophy [1, 2, 38, 39]. Meanwhile, the new and challenging CARLA v2 [13] and nuPlan [14] benchmarks have been introduced to facilitate research into this area.

1.3 Comparison to Related Surveys

We would like to clarify the difference between our survey and previous related surveys [40, 41, 42, 43, 44, 45, 46, 47, 48]. Some prior surveys [40, 41, 42, 43] cover content similar to ours in the sense of an end-to-end system. However, they do not cover new benchmarks and approaches that arose with the significant recent transition in the field, and place a minor emphasis on frontiers and challenges. The others focus on specific topics in this domain, such as imitation learning [44, 45, 46] or reinforcement learning [47, 48]. In contrast, our survey provides up-to-date information on the latest developments in this field, covering a wide span of topics and providing in-depth discussions of critical challenges.

1.4 Contributions

To summarize, this survey has three key contributions:
(a) We provide a comprehensive analysis of end-to-end autonomous driving for the first time, including high-level motivation, methodologies, benchmarks, and more. Instead of optimizing a single block, we advocate for a philosophy to design the algorithm framework as a whole, with the ultimate target of achieving safe and comfortable driving.
(b) We extensively investigate the critical challenges that concurrent approaches face. Out of the more than 270 papers surveyed, we summarize major aspects and provide in-depth analysis, including topics on generalizability, language-guided learning, causal confusion, *etc*.
(c) We cover the broader impact of how to embrace large foundation models and data engines. We believe that this line of research and the large scale of high-quality data it provides could significantly advance this field. To facilitate future research, we maintain an active repository updated with new literature and open-source projects.

2 Methods

This section reviews fundamental principles behind most existing end-to-end self-driving approaches. Sec. 2.1 discusses methods using imitation learning and provides details on the two most popular sub-categories, namely behavior cloning and inverse optimal control. Sec. 2.2 summarizes methods that follow the reinforcement learning paradigm.

2.1 Imitation Learning

Imitation learning (IL), also referred to as learning from demonstrations, trains an agent to learn the policy by imitating the behavior of an expert. IL requires a dataset $D = \{\xi_i\}$ containing trajectories collected under the expert's policy π_β, where each trajectory is a sequence of state-action pairs. The goal of IL is to learn an agent policy π that matches π_β.

The policy π can output planned trajectories or control signals. Early works usually adopt control outputs, due to the ease of collection. However, predicting controls at different steps could lead to discontinuous maneuvers and the network inherently specializes to the vehicle dynamics which hinders generalization to other vehicles. Another genre of works predicts waypoints. It considers a relatively longer time horizon. Meanwhile, converting trajectories for vehicles to track into control signals needs additional controllers, which is non-trivial and involves vehicle models and control algorithms. Since no clear performance gap has been observed between these two paradigms, we do not differentiate them explicitly in this survey. An interesting and more in-depth discussion can be found in [22].

One widely used category of IL is behavior cloning (BC) [49], which reduces the problem to supervised learning. Inverse Optimal Control (IOC), also known as Inverse Reinforcement Learning (IRL) [50] is another type of IL method that utilizes expert demonstrations to learn a reward function. We elaborate on these two categories below.

2.1.1 Behavior Cloning

In BC, matching the agent's policy with the expert's is accomplished by minimizing planning loss as supervised

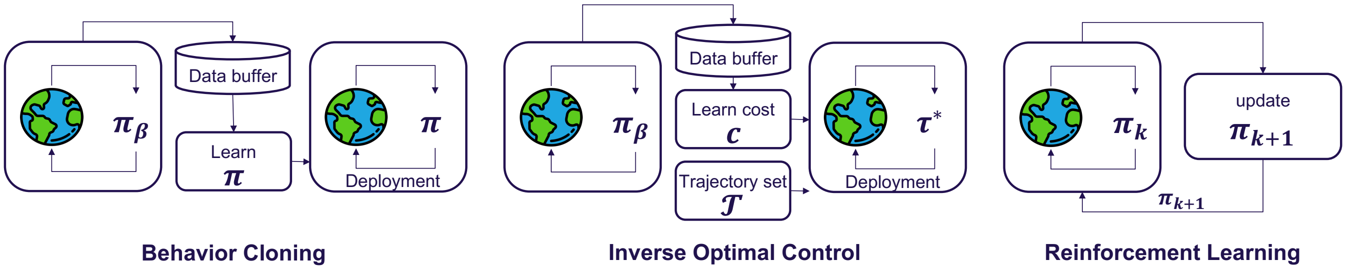

Fig. 3: **Overview of methods in end-to-end autonomous driving.** We illustrate three popular paradigms, including two imitation learning frameworks (behavior cloning and inverse optimal control), as well as online reinforcement learning.

learning over the collected dataset: $\mathbb{E}_{(s,a)}\,\ell(\pi_\theta(s), a)$. Here, $\ell(\pi_\theta(s), a)$ represents a loss function that measures the distance between the agent action and the expert action.

Early applications of BC for driving [3, 8, 51] utilized an end-to-end neural network to generate control signals from camera inputs. Further enhancements, such as multi-sensor inputs [6, 52], auxiliary tasks [16, 28], and improved expert design [21], have been proposed to enable BC-based end-to-end driving models to handle challenging urban scenarios.

BC is advantageous due to its simplicity and efficiency, as it does not require hand-crafted reward design, which is crucial for RL. However, there are some common issues. During training, it treats each state as independently and identically distributed, resulting in an important problem known as covariate shift. For general IL, several on-policy methods have been proposed to address this issue [26, 53, 54, 55]. In the context of end-to-end autonomous driving, DAgger [26] has been adopted in [5, 10, 25, 56]. Another common problem with BC is causal confusion, where the imitator exploits and relies on false correlations between certain input components and output signals. This issue has been discussed in the context of end-to-end autonomous driving in [57, 58, 59, 60]. These two challenging problems are further discussed in Sec. 4.9 and Sec. 4.8, respectively.

2.1.2 *Inverse Optimal Control*

Traditional IOC algorithms learn an unknown reward function $R(s, a)$ from expert demonstrations, where the expert's reward function can be represented as a linear combination of features [50, 61, 62, 63, 64]. However, in continuous, high-dimensional autonomous driving scenarios, the definition of the reward is implicit and difficult to optimize.

Generative adversarial imitation learning [65, 66, 67] is a specialized approach in IOC that designs the reward function as an adversarial objective to distinguish the expert and learned policies, similar to the concept of generative adversarial networks [68]. Recently, several works propose optimizing a cost volume or cost function with auxiliary perceptual tasks. Since a cost is an alternative representation of the reward, we classify these methods as belonging to the IOC domain. We define the cost learning framework as follows: end-to-end approaches learn a reasonable cost $c(\cdot)$ and use algorithmic trajectory samplers to select the trajectory τ^* with the minimum cost, as illustrated in Fig. 3.

Regarding cost design, it has representations including a learned cost volume in a bird's-eye-view (BEV) [32], joint energy calculated from other agents' future motion [69], or a set of probabilistic semantic occupancy or freespace layers [39, 70, 71]. On the other hand, trajectories are typically sampled from a fixed expert trajectory set [1, 72] or processed by parameter sampling with a kinematic model [32, 38, 39, 70]. Then, a max-margin loss is adopted as in classic IOC methods to encourage the expert demonstration to have a minimal cost while others have high costs.

Several challenges exist with cost learning approaches. In particular, in order to generate more realistic costs, HD maps, auxiliary perception tasks, and multiple sensors are typically incorporated, which increases the difficulty of learning and constructing datasets for multi-modal multi-task frameworks. Nevertheless, the aforementioned cost learning methods significantly enhance the safety and interpretability of decisions (see Sec. 4.6), and we believe that the industry-inspired end-to-end system design is a viable approach for real-world applications.

2.2 Reinforcement Learning

Reinforcement learning (RL) [73, 74] is a field of learning by trial and error. The success of deep Q networks (DQN) [75] in achieving human-level control on the Atari benchmark [76] has popularized deep RL. DQN trains a neural network called the critic (or Q network), which takes as input the current state and an action, and predicts the discounted return of that action. The policy is then implicitly defined by selecting the action with the highest predicted return.

RL requires an environment that allows potentially unsafe actions to be executed, to collect novel data (*e.g.*, via random actions). Additionally, RL requires significantly more data to train than IL. For this reason, modern RL methods often parallelize data collection across multiple environments [77]. Meeting these requirements in the real world presents great challenges. Therefore, almost all papers that use RL in driving have only investigated the technique in simulation. Most use different extensions of DQN. The community has not yet converged on a specific RL algorithm.

RL has successfully learned lane following on a real car on an empty street [4]. Despite this encouraging result, it must be noted that a similar task was already accomplished by IL three decades prior [3]. To date, no report has shown results for end-to-end training with RL that are competitive with IL. The reason for this failure likely is that the gradients obtained via RL are insufficient to train deep perception architectures (*i.e.*, ResNet) required for driving. Models used in benchmarks like Atari, where RL succeeds, are relatively shallow, consisting of only a few layers [78].

RL has been successfully applied in end-to-end driving when combined with supervised learning (SL). Implicit affordances [18, 19] pre-train the CNN encoder using SL with tasks like semantic segmentation. In the second stage, this encoder is frozen, and a shallow policy head is trained on the features from the frozen encoder with a modern version of Q-learning [79]. RL can also be used to finetune full networks that were pre-trained using IL [17, 80].

RL can also been effectively applied, if the network has access to privileged simulator information. [48, 81, 82]. Privileged RL agents can be used for dataset curation. Roach [21] trains an RL agent on privileged BEV semantic maps and uses the policy to automatically collect a dataset with which a downstream IL agent is trained. WoR [20] employs a Q-function and tabular dynamic programming to generate additional or improved labels for a static dataset.

A challenge in the field is to transfer the findings from simulation to the real world. In RL, the objective is expressed as reward functions, and many algorithms require them to be dense and provide feedback at each environment step. Current works typically use simple objectives, such as progress and collision avoidance. These simplistic designs potentially encourage risky behaviors [81]. Devising or learning better reward functions remains an open problem. Another direction would be to develop RL algorithms that can handle sparse rewards, enabling the optimization of relevant metrics directly. RL can be effectively combined with world models [83, 84, 85], though this presents specific challenges (See Sec. 4.3). Current RL solutions for driving rely heavily on low-dimensional representations of the scene, and this issue is further discussed in Sec. 4.2.2.

3 BENCHMARKING

Autonomous driving systems require a comprehensive evaluation to ensure safety. Researchers must benchmark these systems using appropriate datasets, simulators, metrics, and hardware to accomplish this. This section delineates three approaches for benchmarking end-to-end autonomous driving systems: (1) real-world evaluation, (2) online or closed-loop evaluation in simulation, and (3) offline or open-loop evaluation on driving datasets. We focus on the scalable and principled online simulation setting and summarize real-world and offline assessments for completeness.

3.1 Real-world Evaluation

Early efforts on benchmarking self-driving involved real-world evaluation. Notably, DARPA initiated a series of races. The first event offered $1M in prize money for autonomously navigating a 240km route through the Mojave desert, which no team achieved [86]. The final series event, called the DARPA Urban Challenge, required vehicles to navigate a 96km mock-up town course, adhering to traffic laws and avoiding obstacles [87]. These races fostered important developments in self-driving, such as LiDAR sensors. Following this spirit, the University of Michigan established MCity [88], a large controlled real-world environment to facilitate testing autonomous vehicles. However, such academic ventures have not been widely employed for end-to-end systems due to a lack of data and vehicles. In contrast, industries with the resources to deploy fleets of driverless vehicles could rely on real-world evaluation to benchmark improvements in their algorithms.

3.2 Online/Closed-loop Simulation

Conducting tests of self-driving systems in the real world is costly and risky. To address this challenge, simulation is a viable alternative [14, 89, 90, 91, 92, 93]. Simulators facilitate rapid prototyping and testing, enable the quick iteration of ideas, and provide low-cost access to diverse scenarios for unit testing. In addition, simulators offer tools for measuring performance accurately. However, their primary disadvantage is that the results obtained in a simulated environment do not necessarily generalize to the real world (Sec. 4.9.3).

Closed-loop evaluation involves building a simulated environment that closely mimics a real-world driving environment. The evaluation entails deploying the driving system in simulation and measuring its performance. The system has to navigate safely through traffic while progressing toward a designated goal location. There are four main sub-tasks involved in developing such simulators: parameter initialization, traffic simulation, sensor simulation, and vehicle dynamics simulation. We briefly describe these subtasks below, followed by a summary of currently available open-source simulators for closed-loop benchmarks.

3.2.1 Parameter Initialization

Simulation offers the benefit of a high degree of control over the environment, including weather, maps, 3D assets, and low-level attributes such as the arrangement of objects in a traffic scene. While powerful, the number of these parameters is substantial, resulting in a challenging design problem. Current simulators tackle this in two ways:

Procedural Generation: Traditionally, initial parameters are hand-tuned by 3D artists and engineers [89, 90, 91, 92]. This limits scalability. Recently, some of the simulation properties can be sampled from a probabilistic distribution with computer algorithms, which we refer to as procedural generation [94]. Procedural generation algorithms combine rules, heuristics, and randomization to create diverse road networks, traffic patterns, lighting conditions, and object placements [95, 96]. Due to its efficiency compared to fully manual design, it has become one of the most commonly used methods of initialization for video games and simulations. Nevertheless, the process still needs pre-defined parameters and algorithms to control generation reliability, which is time-consuming and requires a lot of expertise.

Data-Driven: Data-driven approaches for simulation initialization aim to learn the required parameters. Arguably, the simplest way is to sample from real-world driving logs [14, 93], where parameters such as road maps or traffic patterns are directly extracted from pre-recorded datasets. The advantage of log sampling is its ability to capture the natural variability present in real-world data, leading to more realistic simulation scenarios. However, it may not encompass rare situations that are critical for testing the robustness of autonomous driving systems. The initial parameters can be optimized to increase the representation of such scenarios [7, 34, 35]. Another advanced data-driven approach to initialization is generative modeling, where machine learning algorithms are utilized to learn the underlying structure and distributions of real-world data. They can then generate novel scenarios that resemble the real world but were not included in the original data [97, 98, 99, 100].

3.2.2 Traffic Simulation

Traffic simulation involves generating and positioning virtual entities in the environment with realistic motion [98, 101]. These entities often include vehicles (such as cars, motorcycles, bicycles, *etc.*) and pedestrians. Traffic simulators must account for the effects of speed, acceleration, braking, obstructions, and the behavior of other entities. Moreover, traffic light states must be periodically updated to simulate realistic city driving. There are two popular approaches for traffic simulation, which we describe below.

Rule-Based: Rule-based traffic simulators use predefined rules to generate the motion of traffic entities. The most prominent implementation of this concept is the Intelligent Driver Model (IDM) [102]. IDM is a car-following model that computes acceleration for each vehicle based on its current speed, the speed of the leading vehicle, and a desired safety distance. Although widely used and straightforward, this approach may be inadequate to simulate realistic motion and complex interactions in urban environments.

Data-Driven: Realistic human traffic behavior is highly interactive and complex, including lane changing, merging, sudden stopping, *etc*. To model such behavior, data-driven traffic simulation utilizes data collected from real-world driving. These models can capture more nuanced, realistic behavior but require significant amounts of labeled data for training. A wide variety of learning-based techniques have been proposed for this task [98, 99, 101, 103, 104, 105].

3.2.3 Sensor Simulation

Sensor simulation is crucial for evaluating end-to-end self-driving systems. This involves generating simulated raw sensor data, such as camera images or LiDAR scans that the driving system would receive from different viewpoints in the simulator [106, 107, 108]. This process needs to take into account noise and occlusions to realistically assess the autonomous system. There are two main branches of ideas concerning sensor simulation, as described below.

Graphics-Based: Recent computer graphics simulators use 3D models of the environment, along with traffic entity models, to generate sensor data via approximations of physical rendering processes in the sensors [90, 91]. For example, this can involve occlusions, shadows, and reflections present in real-world environments while simulating camera images. However, the realism of graphics-based simulation is often subpar or comes at the cost of heavy computation, making parallelization non-trivial [109]. It is closely tied to the quality of the 3D models and the approximations used in modeling the sensors. A comprehensive survey of graphics-based rendering for driving data is provided in [110].

Data-Driven: Data-driven sensor simulation leverages real-world sensor data to create the simulation where both the ego vehicle and background traffic may move differently from the way they did in recordings [111, 112, 113]. Popular methods are Neural Radiance Fields (NeRF) [114] and 3D Gaussian Splatting [115], which can generate novel views of a scene by learning an implicit representation of the scene's geometry and appearance. These methods can produce more realistic sensor data visually than graphics-based approaches, but they have limitations such as high rendering times or requiring independent training for each scene being reconstructed [108, 116, 117, 118, 119]. Another approach to data-driven sensor simulation is domain adaptation, which aims to minimize the gap between real and graphics-based simulated sensor data [120]. Deep learning techniques such as GANs can be employed to improve realism (Sec. 4.9.3).

Simulator	Benchmarks
CARLA	CoRL [91], noCrash [124], Town05 [6], LAV [52], Roach [21], Longest6 [28], Leaderboard v1 and v2 [13]
nuPlan	NAVSIM [125], Val14 [126], Leaderboard [14]

TABLE 1: **Open-source Simulators** with active benchmarks for closed-loop evaluation of autonomous driving.

3.2.4 Vehicle Dynamics Simulation

The final aspect of driving simulation pertains to ensuring that the simulated vehicle adheres to physically plausible motion. Most existing publicly available simulators use highly simplified vehicle models, such as the unicycle model [121] or the bicycle model [122]. However, in order to facilitate seamless transfer of algorithms from simulation to the real world, it is essential to incorporate more accurate physical modeling of vehicle dynamics. For instance, CARLA adopts a multi-body system approach, representing a vehicle as a collection of sprung masses on four wheels. For a comprehensive review, please refer to [123].

3.2.5 Benchmarks

We give a succinct overview of end-to-end driving benchmarks available up to date in Table 1. In 2019, the original benchmark released with CARLA [91] was solved with near-perfect scores [5]. The subsequent NoCrash benchmark [124] involves training on a single CARLA town under specific weather conditions and testing generalization to another town and set of weathers. Instead of a single town, the Town05 benchmark [6] involves training on all available towns while withholding Town05 for testing. Similarly, the LAV benchmark trains on all towns except Town02 and Town05, which are both reserved for testing. Roach [21] uses a setting with 3 test towns, albeit all seen during training, and without the safety-critical scenarios in Town05 and LAV. Finally, the Longest6 benchmark [28] uses 6 test towns. Two online servers, the leaderboard (v1 and v2) [13], ensure fair comparisons by keeping evaluation routes confidential. Leaderboard v2 is highly challenging due to the long route length (over 8km on average, as opposed to 1-2km on v1) and a wide variety of new traffic scenarios.

The nuPlan simulator is currently accessible for evaluating end-to-end systems via the NAVSIM project [125]. Further, there are two benchmarks on which agents input maps and object properties via the data-driven parameter initialization for nuPlan (Sec. 3.2.1). Val14, proposed in [126], uses a validation split of nuPlan. The leaderboard, a submission server with the private test set, was used in the 2023 nuPlan challenge, but it is no longer public for submissions.

3.3 Offline/Open-loop Evaluation

Open-loop evaluation mainly assesses a system's performance against pre-recorded expert driving behavior. This method requires evaluation datasets that include (1) sensor readings, (2) goal locations, and (3) corresponding future driving trajectories, usually obtained from human drivers. Given sensor inputs and goal locations as inputs, performance is measured by comparing the system's predicted future trajectory against the trajectory in the driving log. Systems are evaluated based on how closely their trajectory predictions match the human ground truth, as well as auxiliary metrics such as the collision probability with other agents. The advantage of open-loop evaluation is that it is easy to implement using realistic traffic and sensor data, as it does not require a simulator. However, the key disadvantage is that it does not measure performance in the actual test distribution encountered during deployment. During testing, the driving system may deviate from the expert driving corridor, and it is essential to verify the system's ability to recover from such drift (Sec. 4.9.2). Furthermore, the distance between the predicted and the recorded trajectories is not an ideal metric in a multi-modal scenario. For example, in the case of merging into a turning lane, both the options of merging immediately or later could be valid, but open-loop evaluation penalizes the option that was not observed in the data. Therefore, besides measuring collision probability and prediction errors, a few metrics were proposed to cover more comprehensive aspects such as traffic violations, progress, and driving comfort [126].

This approach requires comprehensive datasets of trajectories to draw from. The most popular datasets for this purpose include nuScenes [127], Argoverse [128], Waymo [129], and nuPlan [14]. All of these datasets comprise a large number of real-world driving traversals with varying degrees of difficulty. However, open-loop results do not provide conclusive evidence of improved driving behavior in closed-loop, due to the aforementioned drawbacks [124, 126, 130, 131]. Overall, a realistic closed-loop benchmarking, if available and applicable, is recommended in future research.

4 Challenges

Following each topic illustrated in Fig. 1, we now walk through current challenges, related works or potential resolutions, risks, and opportunities. We start with challenges in handling different input modalities in Sec. 4.1, followed by a discussion on visual abstraction for efficient policy learning in Sec. 4.2. Further, we introduce learning paradigms such as world model learning (Sec. 4.3), multi-task frameworks (Sec. 4.4), and policy distillation (Sec. 4.5). Finally, we discuss general issues that impede safe and reliable end-to-end autonomous driving, including interpretability in Sec. 4.6, safety guarantees in Sec. 4.7, causal confusion in Sec. 4.8, and robustness in Sec. 4.9.

4.1 Dilemma over Sensing and Input Modalities

4.1.1 Sensing and Multi-sensor Fusion

Sensing: Though early work [8] successfully achieved following a lane with a monocular camera, this single input modality cannot handle complex scenarios. Therefore, various sensors in Fig. 4 have been introduced for recent self-driving vehicles. Particularly, RGB images from cameras replicate how humans perceive the world, with abundant semantic details; LiDARs or stereo cameras provide accurate 3D spatial knowledge. Emerging sensors like mmWave radars and event cameras excel at capturing objects' relative movement. Additionally, vehicle states from speedometers and IMUs, together with navigation commands, are other lines of input that guide the driving system. However, various sensors possess distinct perspectives, data distributions, and huge price gaps, thereby posing challenges in effectively designing the sensory layout and fusing them to complement each other for autonomous driving.

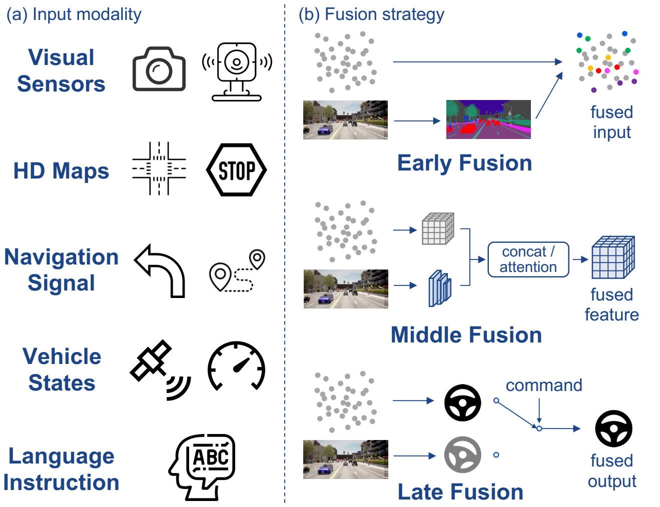

Fig. 4: **Examples of input modality and fusion strategy.** Different modalities have distinct characteristics, leading to the challenge of effective sensor fusion. We take point clouds and images as examples to depict various fusion strategies.

Multi-sensor fusion has predominantly been discussed in perception-related fields, *e.g.*, object detection [132, 133] and semantic segmentation [134, 135], and is typically categorized into three groups: early, mid, and late fusion. End-to-end autonomous driving algorithms explore similar fusion schemes. **Early fusion** combines sensory inputs before feeding them into shared feature extractors, where concatenation is a common way for fusion [32, 136, 137, 138, 139]. To resolve the view discrepancy, some works project point clouds on images [140] or vice versa (predicting semantic labels for LiDAR points [52, 141]). On the other hand, **late fusion** combines multiple results from multi-modalities. It is less discussed due to its inferior performance [6, 142]. Contrary to these methods, **middle fusion** achieves multi-sensor fusion within the network by separately encoding inputs and then fusing them at the feature level. Naive concatenation is also frequently adopted [15, 22, 30, 143, 144, 145, 146, 147]. Recently, works have employed Transformers [27] to model interactions among features [6, 28, 29, 148, 149]. The attention mechanism in Transformers has demonstrated great effectiveness in aggregating the context of different sensor inputs and achieving safer end-to-end driving.

Inspired by the progress in perception, it is beneficial to model modalities in a unified space such as BEV [132, 133]. End-to-end driving also requires identifying policy-related contexts and discarding irrelevant details. We discuss perception-based representations in Sec. 4.2.1. Besides, the self-attention layer, interconnecting all tokens freely, incurs a significant computational cost and cannot guarantee useful information extraction. Advanced Transformer-based fusion mechanisms in the perception field, such as [150, 151], hold promise for application to the end-to-end driving task.

4.1.2 Language as Input

Humans drive using both visual perception and intrinsic knowledge which together form causal behaviors. In areas related to autonomous driving such as embodied AI, incorporating natural language as fine-grained knowledge and instructions to control the visuomotor agent has achieved notable progress [152, 153, 154, 155]. However, compared to robotic applications, the driving task is more straightforward without the need for task decomposition, and the outdoor environment is much more complex with highly dynamic agents but few distinctive anchors for grounding.

To incorporate linguistic knowledge into driving, a few datasets are proposed to benchmark outdoor grounding and visual language navigation tasks [156, 157, 158, 159]. HAD [160] takes human-to-vehicle advice and adds a visual grounding task. Sriram *et al.* [161] translate natural language instructions into high-level behaviors, while [162, 163] directly ground the texts. CLIP-MC [164] and LM-Nav [165] utilize CLIP [166] to extract both linguistic knowledge from instructions and visual features from images.

Recently, observing the rapid development of large language models (LLMs) [167, 168], works encode the perceived scene into tokens and prompt them to LLMs for control prediction and text-based explanations [169, 170, 171]. Researchers also formulate the driving task as a question-answering problem and construct corresponding benchmarks [172, 173]. They highlight that LLMs offer opportunities to handle sophisticated instructions and generalize to different data domains, which share similar advantages to applications in robotic areas [174]. However, LLMs for on-road driving could be challenging at present, considering their long inference time, low quantitative accuracy, and instability of outputs. Potential resolutions could be employing LLMs on the cloud specifically for complex scenarios and using them solely for high-level behavior prediction.

4.2 Dependence on Visual Abstraction

End-to-end autonomous driving systems roughly have two stages: encoding the state into a latent feature representation, and then decoding the driving policy with intermediate features. In urban driving, the input state, *i.e.*, the surrounding environment and ego state, is much more diverse and high-dimensional compared to common policy learning benchmarks such as video games [18, 175], which might lead to the misalignment between representations and necessary attention areas for policy making. Hence, it is helpful to design "good" intermediate perception representations, or first pre-train visual encoders using proxy tasks. This enables the network to extract useful information for driving effectively, thus facilitating the subsequent policy stage. Furthermore, this can improve the sample efficiency for RL methods.

4.2.1 Representation Design

Naive representations are extracted with various backbones. Classic convolutional neural networks (CNNs) still dominate, with advantages in translation equivariance and high efficiency [176]. Depth-pre-trained CNNs [177] significantly boost perception and downstream performance. In contrast, Transformer-based feature extractors [178, 179] show great scalability in perception tasks while not being widely adopted for end-to-end driving yet. For driving-specific representations, researchers introduce the concept of bird's-eye-view (BEV), fusing different sensor modalities and temporal information within a unified 3D space [133, 180, 181]. It also facilitates easy adaptions to downstream tasks [2, 30, 182, 183]. In addition, grid-based 3D occupancy is developed to capture irregular objects and used for collision avoidance in planning [184]. Nevertheless, the dense representation brings huge computation costs compared to BEV methods.

Another unsettled problem is representations of the map. Traditional autonomous driving relies on HD Maps. Due to the high cost of availability of HD Maps, online mapping methods have been devised with different formulations, such as BEV segmentation [185], vectorized lanlines [186], centerlines and their topology [187, 188], and lane segments [189]. However, the most suitable formulation for end-to-end systems remains unvalidated.

Though various representation designs offer possibilities of how to design the subsequent decision-making process, they also place challenges as co-designing both parts is necessary for a whole framework. Besides, given the trends observed in several simple yet effective approaches with scaling up training resources [22, 28], the ultimate necessity of explicit representations such as maps is uncertain.

4.2.2 Representation Learning

Representation learning often incorporates certain inductive biases or prior information. There inevitably exist possible information bottlenecks in the learned representation, and redundant context unrelated to decisions may be removed.

Some early methods directly utilize semantic segmentation masks from off-the-shelf networks as the input representation for subsequent policy training [190, 191]. SESR [192] further encodes segmentation masks into class-disentangled representations through a VAE [193]. In [194, 195], predicted affordance indicators, such as traffic light states, offset to the lane center, and distance to the leading vehicle, are used as representations for policy learning.

Observing that results like segmentation as representations can create bottlenecks defined by humans and result in loss of useful information, some have chosen intermediate features from pre-training tasks as effective representations for RL training [18, 19, 196, 197]. In [198], latent features in VAE are augmented by attention maps obtained from the diffused boundary of segmentation and depth maps to highlight important regions. TARP [199] utilizes data from a series of previous tasks to perform different tasks-related prediction tasks to acquire useful representations. In [200], the latent representation is learned by approximating the π-bisimulation metric, which is comprised of differences of rewards and outputs from the dynamics model. ACO [36] learns discriminative features by adding steering angle categorization into the contrastive learning structure. Recently, PPGeo [12] proposes to learn effective representation through motion prediction together with depth estimation in a self-supervised way on uncalibrated driving videos. ViDAR [201] utilizes the raw image-point cloud pairs and pretrains the visual encoder with a point cloud forecasting pre-task. These works demonstrate that self-supervised representation learning from large-scale unlabeled data for policy learning is promising and worthy of future exploration.

4.3 Complexity of World Modeling for Model-based RL

Besides the ability to better abstract perceptual representations, it is essential for end-to-end models to make reasonable predictions about the future to take safe maneuvers. In this section, we mainly discuss the challenges of current model-based policy learning works, where a world model provides explicit future predictions for the policy model.

Deep RL typically suffers from the high sample complexity, which is pronounced in autonomous driving. Model-based reinforcement learning (MBRL) offers a promising direction to improve sample efficiency by allowing agents to interact with the learned world model instead of the actual environment. MBRL methods employ an explicit world (environment) model, which is composed of transition dynamics and reward functions. This is particularly helpful in driving, as simulators like CARLA are relatively slow.

However, modeling the highly dynamic environment is a challenging task. To simplify the problem, Chen *et al.* [20] factor the transition dynamics into a non-reactive world model and a simple kinematic bicycle model. In [138], a probabilistic sequential latent model is used as the world model. To address the potential inaccuracy of the learned world model, Henaff *et al.* [202] train the policy network with dropout regularization to estimate the uncertainty cost. Another approach [203] uses an ensemble of multiple world models to provide uncertainty estimation, based on which imaginary rollouts could be truncated and adjusted accordingly. Motivated by Dreamer [83], ISO-Dream [204] decouples visual dynamics into controllable and uncontrollable states, and trains the policy on the disentangled states.

It is worth noting that learning world models in raw image space is non-trivial for autonomous driving. Important small details, such as traffic lights, would easily be missed in predicted images. To tackle this, a few works [205, 206, 207] employ the prevailing diffusion technique [208]. MILE [209] incorporates the Dreamer-style world model learning in the BEV segmentation space as an auxiliary task besides imitation learning. SEM2 [137] also extends the Dreamer structure but with BEV map inputs, and uses RL for training. Besides directly using the learned world model for MBRL, DeRL [197] combines a model-free actor-critic framework with the world model, by fusing self-assessments of the action or state from both models.

World model learning for end-to-end autonomous driving is an emerging and promising direction as it greatly reduces the sample complexity for RL, and understanding the world is helpful for driving. However, as the driving environment is highly complex and dynamic, further study is still needed to determine what needs to be modeled and how to model the world effectively.

4.4 Reliance on Multi-Task Learning

Multi-task learning (MTL) involves jointly performing several related tasks based on a shared representation through separate heads. MTL provides advantages such as computational cost reduction, the sharing of relevant domain knowledge, and the ability to exploit task relationships to improve model's generalization ability [210]. Consequently, MTL is well-suited for end-to-end driving, where the ultimate policy prediction requires a comprehensive understanding of the environment. However, the optimal combination of auxiliary tasks and appropriate weighting of losses to achieve the best performance presents a significant challenge.

In contrast to common vision tasks where dense predictions are closely correlated, end-to-end driving predicts a sparse signal. The sparse supervision increases the difficulty of extracting useful information for decision-making in the encoder. For image input, auxiliary tasks such as semantic segmentation [28, 31, 140, 211, 212, 213] and depth estimation [28, 31, 211, 212, 213] are commonly adopted in end-to-end autonomous driving models. Semantic segmentation helps the model gain a high-level understanding of the scene; depth estimation enables the model to capture the 3D geometry of the environment and better estimate distances to critical objects. Besides auxiliary tasks on perspective images, 3D object detection [28, 31, 52] is also useful for LiDAR encoders. As BEV becomes a natural and popular representation for autonomous driving, tasks such as BEV segmentation are included in models [11, 23, 28, 29, 30, 31, 52, 149] that aggregate features in BEV space. Moreover, in addition to these vision tasks, [29, 211, 214] also predict visual affordances including traffic light states, distances to opposite lanes, *etc*. Nonetheless, constructing large-scale datasets with multiple types of aligned and high-quality annotations is non-trivaial for real-world applications, which remain as a great concern due to current models' reliance on MTL.

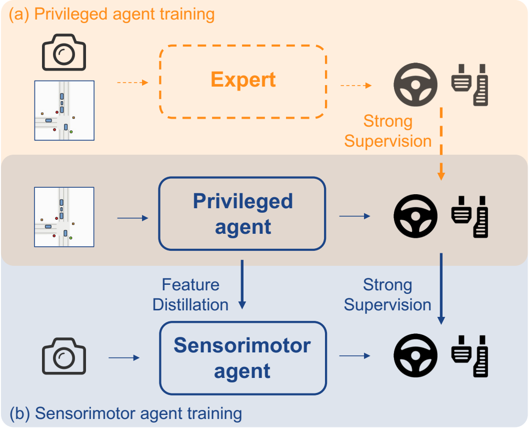

Fig. 5: **Policy distillation.** (a) The **privileged agent** learns a robust policy with access to privileged ground-truth information. The expert is labeled with dashed lines to indicate that it is not mandatory if the privileged agent is trained via RL. (b) The **sensorimotor agent** imitates the privileged agent through both feature distillation and output imitation.

4.5 Inefficient Experts and Policy Distillation

As imitation learning, or its predominant sub-category, behavior cloning, is simply supervised learning that mimics expert behaviors, corresponding methods usually follow the "Teacher-Student" paradigm. There lie two main challenges:

(1) Teachers, such as the handcrafted expert autopilot provided by CARLA, are not perfect drivers, though having access to ground-truth states of surrounding agents and maps. (2) Students are supervised by the recorded output with sensor input only, requiring them to extract perceptual features and learn policy from scratch simultaneously.

A few studies propose to divide the learning process into two stages, *i.e.*, training a stronger teacher network and then distilling the policy to the student. In particular, Chen *et al.* [5, 52] first employ a privileged agent to learn how to act with access to the state of the environment, then let the sensorimotor agent (student) closely imitate the privileged agent with distillation at the output stage. More compact BEV representations as input for the privileged agent provide stronger generalization abilities and supervision than the original expert. The process is depicted in Fig. 5.

Apart from solely supervising planning results, several works also distill knowledge at the feature level. For example, FM-Net [215] employs segmentation and optical flow models as auxiliary teachers to guide feature training. SAM [216] adds L2 feature loss between teacher and student networks, while CaT [23] aligns features in BEV. WoR [20] learns a model-based action-value function and then uses it to supervise the visuomotor policy. Roach [21] trains a stronger privileged expert with RL, eliminating the upper bound of BC. It incorporates multiple distillation targets, *i.e.*, action distribution, values/rewards, and latent features. By leveraging the powerful RL expert, TCP [22] achieves a new state-of-the-art on the CARLA leaderboard with a single camera as visual input. DriveAdpater [182] learns a perception-only student and adapters with the feature alignment objective. The decoupled paradigm fully enjoys the teacher's knowledge and student's training efficiency.

Though huge efforts have been devoted to designing a robust expert and transferring knowledge at various levels, the teacher-student paradigm still suffers from inefficient distillation. For instance, the privileged agent has access to ground-truth states of traffic lights, which are small objects in images and thus hard to distill corresponding features. As a result, the visuomotor agents exhibit large performance gaps compared to their privileged agents. It may also lead to causal confusion for students (see Sec. 4.8). It is worth exploring how to draw more inspiration from general distillation methods in machine learning to minimize the gap.

4.6 Lack of Interpretability

Interpretability plays a critical role in autonomous driving [217]. It enables engineers to better debug the system, provides performance guarantees from a societal perspective, and promotes public acceptance. Achieving interpretability for end-to-end driving models, which are often referred to as "black boxes", is more essential and challenging.

Given trained models, some post-hoc X-AI (explainable AI) techniques could be applied to gain saliency maps [211, 218, 219, 220, 221]. Saliency maps highlight specific regions in the visual input on which the model primarily relies for planning. However, this approach provides limited information, and its effectiveness and validity are difficult to evaluate. Instead, we focus on end-to-end frameworks that directly enhance interpretability in their model design. We introduce each category of interpretability in Fig. 6 below.

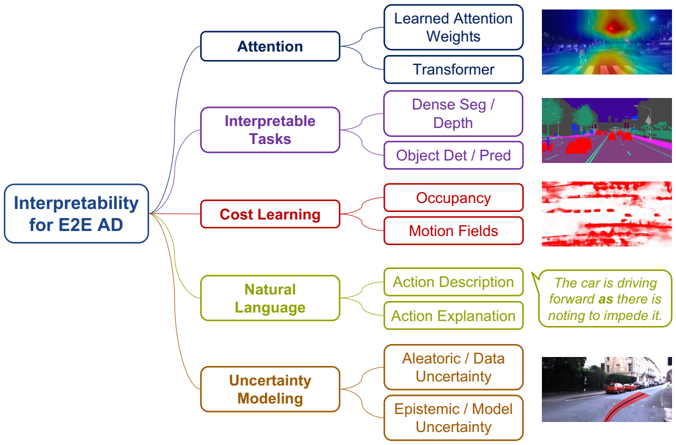

Fig. 6: **Summary of the different forms of interpretability.** They aid in human comprehension of decision-making processes of end-to-end models and the reliability of outputs.

Attention Visualization: The attention mechanism provides a certain degree of interpretability. In [33, 211, 214, 221, 222], a learned attention weight is applied to aggregate important features from intermediate feature maps. Attention weights can also adaptively combine ROI pooled features from different object regions [223] or a fixed grid [224]. NEAT [11] iteratively aggregates features to predict attention weights and refine the aggregated feature. Recently, Transformer attention blocks are employed to better fuse different sensor inputs, and attention maps display important regions in the input for driving decisions [28, 29, 31, 148, 225]. In PlanT [226], attention layers process features from different vehicles, providing interpretable insights into the corresponding action. Similar to post-hoc saliency methods, although attention maps offer straightforward clues about models' focus, their faithfulness and utility remain limited.

Interpretable Tasks: Many IL-based works introduce interpretability by decoding the latent feature representations into other meaningful information besides policy prediction, such as semantic segmentation [2, 11, 15, 28, 29, 31, 52, 140, 164, 211, 212, 213, 227], depth estimation [15, 28, 31, 211, 212], object detection [2, 28, 31, 52], affordance predictions [29, 211, 214], motion prediction [2, 52], and gaze map estimation [228]. Although these methods provide interpretable information, most of them only treat these predictions as auxiliary tasks [11, 15, 28, 31, 140, 211, 212, 214], with no explicit impact on final driving decisions. Some [29, 52] do use these outputs for final actions, but they are incorporated solely for performing an additional safety check.

Rules Integration and Cost Learning: As discussed in Sec. 2.1.2, cost learning-based methods share similarities with traditional modular systems and thus exhibit a certain level of interpretability. NMP [32] and DSDNet [229] construct the cost volume in conjunction with detection and motion prediction results. P3 [39] combines predicted semantic occupancy maps with comfort and traffic rules constraints to construct the cost function. Various representations, such as probabilistic occupancy and temporal motion fields [1], emergent occupancy [71], and freespace [70], are employed to score sampled trajectories. In [38, 126, 183, 230], human expertise and pre-defined rules including safety, comfort, traffic rules, and routes based on perception and prediction outputs are explicitly included to form the cost for trajectory scoring, demonstrating improved robustness and safety.

Linguistic Explainability: As one aspect of interpretability is to help humans understand the system, natural language is a suitable choice for this purpose. Kim *et al.* [33] and Xu *et al.* [231] develop datasets pairing driving videos or images with descriptions and explanations, and propose end-to-end models with both control and explanation outputs. BEEF [232] fuses the predicted trajectory and the intermediate perception features to predict justifications for the decision. ADAPT [233] proposes a Transformer-based network to jointly estimate action, narration, and reasoning. Recently, [170, 172, 173] resort to the progress of multi-modality and foundation models, using LLMs/VLMs to provide decision-related explanations, as discussed in Sec. 4.1.2.

Uncertainty Modeling: Uncertainty is a quantitative approach for interpreting the dependability of deep learning model outputs [234, 235], which can be helpful for designers and users to identify uncertain cases for improvement or necessary intervention. For deep learning, there are two types of uncertainty: aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty is inherent to the task, while epistemic uncertainty is due to limited data or modeling capacity. In [236], authors leverage certain stochastic regularizations in the model to perform multiple forward passes as samples to measure the uncertainty. However, the requirement of multiple forward passes is not feasible in real-time scenarios. Loquercio *et al.* [235] and Filos *et al.* [237] propose capturing epistemic uncertainty with an ensemble of expert likelihood models and aggregating the results to perform safe planning. Regarding methods modeling aleatoric uncertainty, driving actions/planning and uncertainty (usually represented by variance) are explicitly predicted in [147, 238, 239]. Such methods directly model and quantify the uncertainty at the action level as a variable for the network to predict. The planner would generate the final action based on the predicted uncertainty, either choosing the action with the lowest uncertainty from multiple actions [238] or generating a weighted combination of proposed actions based on the uncertainties [147]. Currently, predicted uncertainty is mainly utilized in combination with hard-coded rules. Exploring better ways to model and utilize uncertainty for autonomous driving is necessary.

4.7 Lack of Safety Guarantees

Ensuring safety is of utmost importance when deploying autonomous driving systems in real-world scenarios. However, the learning-based nature of end-to-end frameworks inherently lacks precise mathematical guarantees regarding safety, unlike traditional rule-based approaches [240].

Nevertheless, it should be noted that modular driving stacks have already incorporated specific safety-related constraints or optimizations within their motion planning or speed prediction modules to enforce safety [241, 242, 243]. These mechanisms can potentially be adapted for integration into end-to-end models as post-process steps or safety checks, thereby providing additional safety guarantees. Furthermore, the intermediate interpretability predictions, as discussed in Sec. 4.6, such as detection and motion prediction results, can be utilized in post-processing procedures.

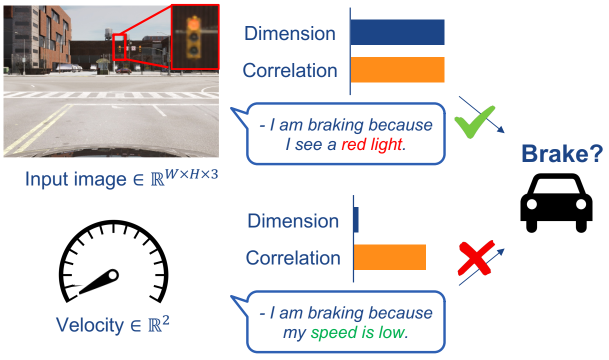
Dimension
Correlation
- I am braking because I see a red light.
Input image ∈ $\mathbb{R}^{W\times H\times 3}$
Brake?
Dimension
Correlation
Velocity ∈ $\mathbb{R}^2$
- I am braking because my speed is low.

Fig. 7: **Causal Confusion.** The current action of a car is strongly correlated with low-dimensional spurious features such as the velocity or the car's past trajectory. End-to-End models may latch on to them leading to causal confusion.

4.8 Causal Confusion

Driving is a task that exhibits temporal smoothness, which makes past motion a reliable predictor of the next action. However, methods trained with multiple frames can become overly reliant on this shortcut [244] and suffer from catastrophic failure during deployment. This problem is referred to as the copycat problem [57] in some works and is a manifestation of causal confusion [245], where access to more information leads to worse performance.

Causal confusion in imitation learning has been a persistent challenge for nearly two decades. One of the earliest reports of this effect was made by LeCun *et al.* [246]. They used a single input frame for steering prediction to avoid such extrapolation. Though simplistic, this is still a preferred solution in current state-of-the-art IL methods [22, 28]. Unfortunately, using a single frame makes it hard to extract the motion of surrounding actors. Another source of causal confusion is speed measurement [16]. Fig. 7 showcases an example of a car waiting at a red light. The action of the car could highly correlate with its speed because it has waited for many frames where the speed is zero and the action is the brake. Only when the traffic light changes from red to green does this correlation break down.

There are several approaches to combat the causal confusion problem when using multiple frames. In [57], the authors attempt to remove spurious temporal correlations from the bottleneck representation by training an adversarial model that predicts the ego agent's past action. Intuitively, the resulting min-max optimization trains the network to eliminate its past from intermediate layers. It works well in MuJoCo but does not scale to complex vision-based driving. OREO [59] maps images to discrete codes representing semantic objects and applies random dropout masks to units that share the same discrete code, which helps in confounded Atari. In end-to-end driving, ChauffeurNet [247] addresses the causal confusion issue by using the past ego-motion as intermediate BEV abstractions and dropping out it with a 50% probability during training. Wen *et al.* [58] propose upweighting keyframes in the training loss, where a decision change occurs (and hence are not predictable by extrapolating the past). PrimeNet [60] improves performance compared to keyframes by using an ensemble,

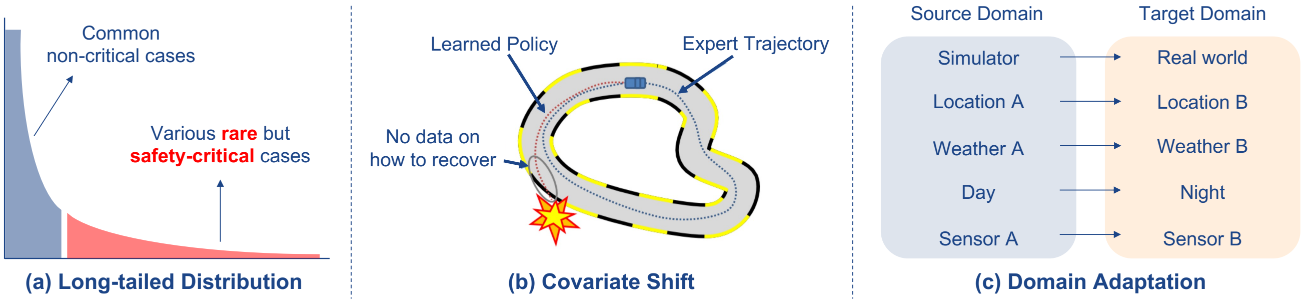

Fig. 8: **Challenges in robustness.** Three primary generalization issues arise in relation to dataset distribution discrepancies, namely long-tailed and normal cases, expert demonstration and test scenarios, and domain shift in locations, weather, *etc*.

where the prediction of a single-frame model is given as additional input to a multi-frame model. Chuang *et al.* [248] do the same but supervise the multi-frame network with action residuals instead of actions. In addition, the problem of causal confusion can be circumvented by using only LiDAR histories (with a single frame image) and realigning point clouds into one coordinate system. This removes ego-motion while retaining information about other vehicles' past states. This technique has been used in multiple works [1, 32, 52], though it was not motivated this way.

However, these studies have used environments that are modified to simplify studying the causal confusion problem. Showing performance improvements in state-of-the-art settings as mentioned in Sec. 3.2.5 remains an open problem.

4.9 Lack of Robustness

4.9.1 Long-tailed Distribution

One important aspect of the long-tailed distribution problem is dataset imbalance, where a few classes make up the majority, as shown in Fig. 8 (a). This poses a big challenge for models to generalize to diverse environments. Various methods mitigate this issue with data processing, including over-sampling [249, 250], under-sampling [251, 252], and data augmentation [253, 254]. Besides, weighting-based approaches [255, 256] are also commonly used.

In the context of end-to-end autonomous driving, the long-tailed distribution issue is particularly severe. Most drives are repetitive and uninteresting *e.g.*, following a lane for many frames. Conversely, interesting safety-critical scenarios occur rarely but are diverse in nature, and hard to replicate in the real world for safety reasons. To tackle this, some works rely on handcrafted scenarios [13, 101, 257, 258, 259] to generate more diverse data in simulation. LBC [5] leverages the privileged agent to create imaginary supervisions conditioned on different navigational commands. LAV [52] includes trajectories of non-ego agents for training to promote data diversity. In [260], a simulation framework is proposed to apply importance-sampling strategies to accelerate the evaluation of rare-event probabilities.

Another line of research [7, 34, 35, 261, 262, 263] generates safety-critical scenarios in a data-driven manner through adversarial attacks. In [261], Bayesian Optimization is employed to generate adversarial scenarios. Learning to collide [35] represents driving scenarios as the joint distribution over building blocks and applies policy gradient RL methods to generate risky scenarios. AdvSim [34] modifies agents' trajectories to cause failures, while still adhering to physical plausibility. KING [7] proposes an optimization algorithm for safety-critical perturbations using gradients through differentiable kinematics models.

In general, efficiently generating realistic safety-critical scenarios that cover the long-tailed distribution remains a significant challenge. While many works focus on adversarial scenarios in simulators, it is also essential to better utilize real-world data for critical scenario mining and potential adaptation to simulation. Besides, a systematic, rigorous, comprehensive, and realistic testing framework is crucial for evaluating end-to-end autonomous driving methods under these long-tailed distributed safety-critical scenarios.

4.9.2 Covariate Shift

As discussed in Sec. 2.1, one important challenge for behavior cloning is covariate shift. The state distributions from the expert's policy and those from the trained agent's policy differ, leading to compounding errors when the trained agent is deployed in unseen testing environments or when the reactions from other agents differ from training time. This could result in the trained agent being in a state that is outside the expert's distribution for training, leading to severe failures. An illustration is presented in Fig. 8 (b).

DAgger (Dataset Aggregation) [26] is a common solution for this issue. DAgger is an iterative training process. The current trained policy is rolled out in each iteration to collect new data, and the expert is used to label the visited states. This enriches the dataset by adding examples of how to recover from suboptimal states that an imperfect policy might visit. The policy is then trained on the augmented dataset, and the process repeats. However, one downside of DAgger is the need for an available expert to query online.

For end-to-end autonomous driving, DAgger is adopted in [24] with an MPC-based expert. To reduce the cost of constantly querying the expert, SafeDAgger [25] extends the original DAgger algorithm by learning a safety policy that estimates the deviation between the current policy and the expert policy. The expert is only queried when the deviation is large. MetaDAgger [56] uses meta-learning with DAgger to aggregate data from multiple environments. LBC [5] adopts DAgger and resamples the data with higher loss more frequently. In DARB [10], to better utilize failure or safety-related samples, it proposes several mechanisms, including task-based, policy-based, and policy & expert-based mechanisms, to sample such critical states.

4.9.3 Domain Adaptation

Domain adaptation (DA) is a type of transfer learning in which the target task is the same as the source task, but the domains differ. Here we discuss scenarios where labels are available for the source domain while there are no labels or a limited amount of labels available for the target domain.

As shown in Fig. 8 (c), domain adaptation for autonomous driving tasks encompasses several cases [264]:

- Sim-to-real: the large gap between simulators used for training and the real world used for deployment.
- Geography-to-geography: different geographic locations with varying environmental appearances.
- Weather-to-weather: changes in sensor inputs caused by weather conditions such as rain, fog, and snow.
- Day-to-night: illumination variations in visual inputs.
- Sensor-to-sensor: possible differences in sensor characteristics, *e.g.*, resolution and relative position.

Note that the aforementioned cases often overlap.

Typically, domain-invariant feature learning is achieved with image translators and discriminators to map images from two domains into a common latent space or representations like segmentation maps [265, 266]. LUSR [267] and UAIL [238] adopt a Cycle-Consistent VAE and GAN, respectively, to project images into a latent representation comprised of a domain-specific part and a domain-general part. In SESR [192], class disentangled encodings are extracted from a semantic segmentation mask to reduce the sim-to-real gap. Domain randomization [268, 269, 270] is also a simple and effective sim-to-real technique for RL policy learning, which is further adapted for end-to-end autonomous driving [190, 271]. It is realized by randomizing the rendering and physical settings of the simulators to cover the variability of the real world during training.

Currently, sim-to-real adaptation through source target image mapping or domain-invariant feature learning is the focus. Other DA cases are handled by constructing a diverse and large-scale dataset. Given that current methods mainly concentrate on the visual gap in images, and LiDAR has become a popular input modality for driving, specific adaptation techniques tailored for LiDARs must also be designed. Besides, traffic agents' behavior gaps between the simulator and the real world should be noticed as well. Incorporating real-world data into simulation through techniques such as NeRF [114] is another promising direction.

5 Future Trends

Considering the challenges and opportunities discussed, we list some crucial directions for future research that may have a broader impact in this field.

5.1 Zero-shot and Few-shot Learning

It is inevitable for autonomous driving models to eventually encounter real-world scenarios that lie beyond the training data distribution. This raises the question of whether we can successfully adapt the model to an unseen target domain where limited or no labeled data is available. Formalizing this task for the end-to-end driving domain and incorporating techniques from the zero-shot/few-shot learning literature are the key steps toward achieving this [272, 273].

5.2 Modular End-to-end Planning

The modular end-to-end planning framework optimizes multiple modules while prioritizing the ultimate planning task, which enjoys the advantages of interpretability as indicated in Sec. 4.6. This is advocated in recent literature [2, 274] and certain industry solutions (Tesla, Wayve, *etc.*) have involved similar ideas. When designing these differentiable perception modules, several questions arise regarding the choice of loss functions, such as the necessity of 3D bounding boxes for object detection, whether opting for BEV segmentation over lane topology for static scene perception, or the training strategies with limited modules' data.

5.3 Data Engine

The importance of large-scale and high-quality data for autonomous driving can never be emphasized enough [275]. Establishing a data engine with an automatic labeling pipeline [276] could greatly facilitate the iterative development of both data and models. The data engine for autonomous driving, especially modular end-to-end planning systems, needs to streamline the process of annotating high-quality perception labels with the aid of large perception models in an automatic way. It should also support mining hard/corner cases, scene generation, and editing to facilitate the data-driven evaluations discussed in Sec. 3.2 and promote diversity of data and the generalization ability of models (Sec. 4.9). A data engine would enable autonomous driving models to make consistent improvements.

5.4 Foundation Model

Recent advancements in foundation models in both language [167, 168] and vision [276, 277] have proved that large-scale data and model capacity can unleash the immense potential of AI in high-level reasoning tasks. The paradigm of finetuning [278] or prompt learning [279], optimization in the form of self-supervised reconstruction [280] or contrastive pairs [166], *etc.*, are all applicable to the end-to-end driving domain. However, we contend that the direct adoption of LLMs for driving might be tricky. The output of an autonomous agent requires steady and accurate measurements, whereas the generative output in language models aims to behave like humans, irrespective of its accuracy. A feasible solution to develop a "foundation" driving model is to train a world model that can forecast the reasonable future of the environment, either in 2D, 3D, or latent space. To perform well on downstream tasks like planning, the objective to be optimized for the model needs to be sophisticated enough, beyond frame-level perception.

6 Conclusion and Outlook

In this survey, we provide an overview of fundamental methodologies and summarize various aspects of simulation and benchmarking. We thoroughly analyze the extensive literature to date, and highlight a wide range of critical challenges and promising resolutions.

Outlook: The industry has dedicated considerable effort over the years to develop advanced modular-based systems capable of achieving self-driving on highways. However, these systems face significant challenges when confronted with complex scenarios, *e.g.*, inner-city streets and intersections. Therefore, an increasing number of companies have started exploring end-to-end autonomous driving techniques specifically tailored for these environments. It is envisioned that with extensive high-quality data collection, large-scale model training, and the establishment of reliable benchmarks, the end-to-end approach will have enormous potential over modular stacks in terms of performance and effectiveness. In summary, end-to-end autonomous driving faces great opportunities and challenges simultaneously, with the ultimate goal of building generalist agents. In this era of emerging technologies, we hope this survey could serve as a starting point to shed new light on this domain.

Acknowledgments

This project is partially supported by National Key R&D Program of China (2022ZD0160104), NSFC (62206172), and Shanghai Committee of Science and Technology (23YF1462000). Andreas Geiger and Bernhard Jaeger are supported by the ERC Starting Grant LEGO-3D (850533), the BMWi in the project KI Delta Learning (project number 19A19013O), and the DFG EXC number 2064/1 - project number 390727645. Kashyap Chitta is supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A. We thank the International Max Planck Research School for Intelligent Systems for supporting Bernhard Jaeger and Kashyap Chitta.

References

[1] S. Casas, A. Sadat, and R. Urtasun, "Mp3: A unified model to map, perceive, predict and plan," in *CVPR*, 2021.

[2] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *CVPR*, 2023.

[3] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *NeurIPS*, 1988.

[4] 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 *ICRA*, 2019.

[5] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *CoRL*, 2020.

[6] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *CVPR*, 2021.

[7] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger, "King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients," in *ECCV*, 2022.

[8] 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.org*, vol. 1604.07316, 2016.

[9] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *ICRA*, 2018.

[10] A. Prakash, A. Behl, E. Ohn-Bar, K. Chitta, and A. Geiger, "Exploring data aggregation in policy learning for vision-based urban autonomous driving," in *CVPR*, 2020.

[11] K. Chitta, A. Prakash, and A. Geiger, "Neat: Neural attention fields for end-to-end autonomous driving," in *ICCV*, 2021.

[12] P. Wu, L. Chen, H. Li, X. Jia, J. Yan, and Y. Qiao, "Policy pre-training for autonomous driving via self-supervised geometric modeling," in *ICLR*, 2023.

[13] CARLA, "CARLA autonomous driving leaderboard." https://leaderboard.carla.org/, 2022.

[14] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," in *CVPR Workshops*, 2021.

[15] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah, *et al.*, "Urban driving with conditional imitation learning," in *ICRA*, 2020.

[16] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *ICCV*, 2019.

[17] X. Liang, T. Wang, L. Yang, and E. Xing, "Cirl: Controllable imitative reinforcement learning for vision-based self-driving," in *ECCV*, 2018.

[18] M. Toromanoff, E. Wirbel, and F. Moutarde, "End-to-end model-free reinforcement learning for urban driving using implicit affordances," in *CVPR*, 2020.

[19] R. Chekroun, M. Toromanoff, S. Hornauer, and F. Moutarde, "Gri: General reinforced imitation and its application to vision-based autonomous driving," *Robotics*, 2023.

[20] D. Chen, V. Koltun, and P. Krähenbühl, "Learning to drive from a world on rails," in *ICCV*, 2021.

[21] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *ICCV*, 2021.

[22] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," in *NeurIPS*, 2022.

[23] J. Zhang, Z. Huang, and E. Ohn-Bar, "Coaching a teachable student," in *CVPR*, 2023.

[24] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. A. Theodorou, and B. Boots, "Agile autonomous driving using end-to-end deep imitation learning," in *RSS*, 2017.

[25] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end simulated driving," in *AAAI*, 2017.

[26] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *AISTATS*, 2011.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[28] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *PAMI*, 2022.

[29] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *CoRL*, 2022.

[30] X. Jia, P. Wu, L. Chen, J. Xie, C. He, J. Yan, and H. Li, "Think twice before driving: Towards scalable decoders for end-to-end autonomous driving," in *CVPR*, 2023.

[31] B. Jaeger, K. Chitta, and A. Geiger, "Hidden biases of end-to-end driving models," in *ICCV*, 2023.

[32] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *CVPR*, 2019.

[33] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, "Textual explanations for self-driving vehicles," in *ECCV*, 2018.

[34] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun, "Advsim: Generating safety-critical scenarios for self-driving vehicles," in *CVPR*, 2021.

[35] W. Ding, B. Chen, M. Xu, and D. Zhao, "Learning to collide: An adaptive safety-critical scenarios generating method," in *IROS*, 2020.

[36] Q. Zhang, Z. Peng, and B. Zhou, "Learning to drive by

watching youtube videos: Action-conditioned contrastive policy pretraining," in *ECCV*, 2022.

[37] J. Zhang, R. Zhu, and E. Ohn-Bar, "Selfd: Self-learning large-scale driving policies from the web," in *CVPR*, 2022.

[38] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *ECCV*, 2022.

[39] 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 *ECCV*, 2020.

[40] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art," *arXiv.org*, vol. 1704.05519, 2017.

[41] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *TNNLS*, 2020.

[42] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, D. Yang, Y. Ai, L. Li, L. Chen, Z. Xuanyuan, *et al.*, "Motion planning for autonomous driving: The state of the art and future perspectives," *TIV*, 2023.

[43] D. Coelho and M. Oliveira, "A review of end-to-end autonomous driving in urban environments," *IEEE Access*, 2022.

[44] A. O. Ly and M. Akhloufi, "Learning to drive by imitation: An overview of deep behavior cloning methods," *TIV*, 2020.

[45] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *TITS*, 2022.

[46] B. Zheng, S. Verma, J. Zhou, I. W. Tsang, and F. Chen, "Imitation learning: Progress, taxonomies and challenges," *TNNLS*, 2022.

[47] Z. Zhu and H. Zhao, "A survey of deep RL and IL for autonomous driving policy learning," *TITS*, 2021.

[48] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. K. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *TITS*, 2021.

[49] M. Bain and C. Sammut, "A framework for behavioural cloning," in *Machine Intelligence 15*, 1995.

[50] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning," in *AAAI*, 2008.

[51] Y. Lecun, E. Cosatto, J. Ben, U. Muller, and B. Flepp, "Dave: Autonomous off-road vehicle control using end-to-end learning," Tech. Rep. DARPA-IPTO Final Report, Courant Institute/CBLL, 2004.

[52] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *CVPR*, 2022.

[53] K. Judah, A. P. Fern, T. G. Dietterich, and P. Tadepalli, "Active imitation learning: Formal and practical reductions to iid learning," *JMLR*, 2014.

[54] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *AISTATS*, 2010.

[55] S. Ross and J. A. Bagnell, "Reinforcement and imitation learning via interactive no-regret learning," *arXiv.org*, vol. 1406.5979, 2014.

[56] A. E. Sallab, M. Saeed, O. A. Tawab, and M. Abdou, "Meta learning framework for automated driving," *arXiv.org*, vol. 1706.04038, 2017.

[57] C. Wen, J. Lin, T. Darrell, D. Jayaraman, and Y. Gao, "Fighting copycat agents in behavioral cloning from observation histories," in *NeurIPS*, 2020.

[58] C. Wen, J. Lin, J. Qian, Y. Gao, and D. Jayaraman, "Keyframe-focused visual imitation learning," in *ICML*, 2021.

[59] J. Park, Y. Seo, C. Liu, L. Zhao, T. Qin, J. Shin, and T.-Y. Liu, "Object-aware regularization for addressing causal confusion in imitation learning," in *NeurIPS*, 2021.

[60] C. Wen, J. Qian, J. Lin, J. Teng, D. Jayaraman, and Y. Gao, "Fighting fire with fire: avoiding dnn shortcuts through priming," in *ICML*, 2022.

[61] D. Brown, W. Goo, P. Nagarajan, and S. Niekum, "Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations," in *ICML*, 2019.

[62] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *ICML*, 2016.

[63] S. Reddy, A. D. Dragan, and S. Levine, "Sqil: Imitation learning via reinforcement learning with sparse rewards," *arXiv.org*, vol. 1905.11108, 2019.

[64] S. Luo, H. Kasaei, and L. Schomaker, "Self-imitation learning by planning," in *ICRA*, 2021.

[65] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *NeurIPS*, 2016.

[66] Y. Li, J. Song, and S. Ermon, "Infogail: Interpretable imitation learning from visual demonstrations," in *NeurIPS*, 2017.

[67] G. Lee, D. Kim, W. Oh, K. Lee, and S. Oh, "Mixgail: Autonomous driving using demonstrations with mixed qualities," in *IROS*, 2020.

[68] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *ACM*, 2020.

[69] H. Wang, P. Cai, R. Fan, Y. Sun, and M. Liu, "End-to-end interactive prediction and planning with optical flow distillation for autonomous driving," in *CVPR Workshops*, 2021.

[70] P. Hu, A. Huang, J. Dolan, D. Held, and D. Ramanan, "Safe local motion planning with self-supervised freespace forecasting," in *CVPR*, 2021.

[71] T. Khurana, P. Hu, A. Dave, J. Ziglar, D. Held, and D. Ramanan, "Differentiable raycasting for self-supervised occupancy forecasting," in *ECCV*, 2022.

[72] S. Chen, B. Jiang, H. Gao, B. Liao, Q. Xu, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Vadv2: End-to-end vectorized autonomous driving via probabilistic planning," *arXiv.org*, vol. 2402.13243, 2024.

[73] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *TNNLS*, 1998.

[74] B. Jaeger and A. Geiger, "An invitation to deep reinforcement learning," *arXiv.org*, vol. 2312.08365, 2023.

[75] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, 2015.

[76] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *JAIR*, 2013.

[77] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. Van Hasselt, and D. Silver, "Distributed prioritized experience replay," *arXiv.org*, vol. 1803.00933, 2018.

[78] J. Bjorck, C. P. Gomes, and K. Q. Weinberger, "Towards deeper deep reinforcement learning with spectral normalization," in *NeurIPS*, 2021.

[79] M. Toromanoff, E. Wirbel, and F. Moutarde, "Is deep reinforcement learning really superhuman on atari? leveling the playing field," *arXiv.org*, vol. 1908.04683, 2019.

[80] E. Ohn-Bar, A. Prakash, A. Behl, K. Chitta, and A. Geiger, "Learning situational driving," in *CVPR*, 2020.

[81] W. B. Knox, A. Allievi, H. Banzhaf, F. Schmitt, and P. Stone, "Reward (mis)design for autonomous driving," *AI*, 2023.

[82] C. Zhang, R. Guo, W. Zeng, Y. Xiong, B. Dai, R. Hu, M. Ren, and R. Urtasun, "Rethinking closed-loop training for autonomous driving," in *ECCV*, 2022.

[83] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *ICLR*, 2020.

[84] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," in *ICLR*, 2021.

[85] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *NeurIPS*, 2018.

[86] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA grand challenge: the great robot race*, vol. 36. Springer, 2007.

[87] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*, vol. 56. Springer Science & Business Media, 2009.

[88] U. of Michigan, "Mcity." https://mcity.umich.edu/, 2015.

[89] T. Team, "Torcs, the open racing car simulator." https://sourceforge.net/projects/torcs/, 2000.

[90] M. Martinez, C. Sitawarin, K. Finch, L. Meincke, A. Yablonski, and A. Kornhauser, "Beyond grand theft auto v for training, testing and enhancing deep learning in self driving cars," *arXiv.org*, vol. 1712.01397, 2017.

[91] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *CoRL*, 2017.

[92] D. Team, "Deepdrive: a simulator that allows anyone with a pc to push the state-of-the-art in self-driving." https://github.com/deepdrive/deepdrive, 2020.

[93] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *PAMI*, 2022.

[94] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural content generation for games: A survey," *TOMM*, 2013.

[95] D. J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: a language for scenario specification and scene generation," in *PLDI*, 2019.

[96] F. Hauer, T. Schmidt, B. Holzmüller, and A. Pretschner, "Did we test all scenarios for automated and autonomous driving systems?," in *ITSC*, 2019.

[97] S. Tan, K. Wong, S. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "Scenegen: Learning to generate realistic traffic scenes," in *CVPR*, 2021.

[98] L. Bergamini, Y. Ye, O. Scheel, L. Chen, C. Hu, L. D. Pero, B. Osinski, H. Grimmett, and P. Ondruska, "Simnet: Learning reactive self-driving simulations from real-world observations," in *ICRA*, 2021.

[99] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou, "Trafficgen: Learning to generate diverse and realistic traffic scenarios," in *ICRA*, 2023.

[100] K. Chitta, D. Dauner, and A. Geiger, "Sledge: Synthesizing simulation environments for driving agents with generative models," in *ECCV*, 2024.

[101] S. Suo, S. Regalado, S. Casas, and R. Urtasun, "Trafficsim: Learning to simulate realistic multi-agent behaviors," in *CVPR*, 2021.

[102] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, 2000.

[103] Z. Zhong, D. Rempe, D. Xu, Y. Chen, S. Veer, T. Che, B. Ray, and M. Pavone, "Guided conditional diffusion for controllable traffic simulation," in *ICRA*, 2023.

[104] D. Xu, Y. Chen, B. Ivanovic, and M. Pavone, "Bits: Bi-level imitation for traffic simulation," in *ICRA*, 2023.

[105] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "TrafficBots: Towards world models for autonomous driving simulation and motion prediction," in *ICRA*, 2023.

[106] S. Manivasagam, S. Wang, K. Wong, W. Zeng, M. Sazanovich, S. Tan, B. Yang, W. Ma, and R. Urtasun, "Lidarsim: Realistic lidar simulation by leveraging the real world," in *CVPR*, 2020.

[107] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, "Geosim: Realistic video simulation via geometry-aware composition for self-driving," in *CVPR*, 2021.

[108] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, "Unisim: A neural closed-loop sensor simulator," in *CVPR*, 2023.

[109] A. Petrenko, E. Wijmans, B. Shacklett, and V. Koltun, "Megaverse: Simulating embodied agents at one million experiences per second," in *ICML*, 2021.

[110] Z. Song, Z. He, X. Li, Q. Ma, R. Ming, Z. Mao, H. Pei, L. Peng, J. Hu, D. Yao, *et al.*, "Synthetic datasets for autonomous driving: A survey," *TIV*, 2024.

[111] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus, "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," *RA-L*, 2020.

[112] A. Amini, T.-H. Wang, I. Gilitschenski, W. Schwarting, Z. Liu, S. Han, S. Karaman, and D. Rus, "Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," in *ICRA*, 2022.

[113] T.-H. Wang, A. Amini, W. Schwarting, I. Gilitschenski, S. Karaman, and D. Rus, "Learning interactive driving policies via data-driven simulation," in *ICRA*, 2022.

[114] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[115] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. on Graphics*, 2023.

[116] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *CVPR*, 2022.

[117] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs," in *CVPR*, 2022.

[118] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. Guibas, A. Tagliasacchi, F. Dellaert, and T. Funkhouser, "Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation," in *CVPR*, 2022.

[119] Y. Yang, Y. Yang, H. Guo, R. Xiong, Y. Wang, and Y. Liao, "Urbangiraffe: Representing urban scenes as compositional generative neural feature fields," in *ICCV*, 2023.

[120] S. R. Richter, H. A. Alhaija, and V. Koltun, "Enhancing photorealism enhancement," *PAMI*, 2023.

[121] A. Schoonwinkel, *Design and test of a computer-stabilized unicycle*. Stanford University, 1988.

[122] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?," in *IV*, 2017.

[123] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[124] F. Codevilla, A. M. Lopez, V. Koltun, and A. Dosovitskiy, "On offline evaluation of vision-based driving models," in *ECCV*, 2018.

[125] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone, *et al.*, "Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking," *arXiv.org*, vol. 2406.15349, 2024.

[126] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *CoRL*, 2023.

[127] 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 *CVPR*, 2020.

[128] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, *et al.*, "Argoverse 2: Next generation datasets

for self-driving perception and forecasting," in *NeurIPS Datasets and Benchmarks*, 2021.

[129] 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 *CVPR*, 2020.

[130] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang, "Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes," *arXiv.org*, vol. 2305.10430, 2023.

[131] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez, "Is ego status all you need for open-loop end-to-end autonomous driving?," in *CVPR*, 2024.

[132] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang, "BEVFusion: A simple and robust liDAR-camera fusion framework," in *NeurIPS*, 2022.

[133] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *ICRA*, 2023.

[134] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, "Sensor fusion for semantic segmentation of urban scenes," in *ICRA*, 2015.

[135] G. P. Meyer, J. Charland, D. Hegde, A. Laddha, and C. Vallespi-Gonzalez, "Sensor fusion for joint 3d object detection and semantic segmentation," in *CVPR Workshops*, 2019.

[136] B. Zhou, P. Krähenbühl, and V. Koltun, "Does computer vision matter for action?," *Science Robotics*, 2019.

[137] Z. Gao, Y. Mu, R. Shen, C. Chen, Y. Ren, J. Chen, S. E. Li, P. Luo, and Y. Lu, "Enhance sample efficiency and robustness of end-to-end urban autonomous driving via semantic masked world model," *TITS*, 2024.

[138] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *TITS*, 2022.

[139] P. Cai, S. Wang, H. Wang, and M. Liu, "Carl-lead: Lidar-based end-to-end autonomous driving with contrastive deep reinforcement learning," *arXiv.org*, vol. 2109.08473, 2021.

[140] Z. Huang, C. Lv, Y. Xing, and J. Wu, "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding," *IEEE Sensors Journal*, 2020.

[141] O. Natan and J. Miura, "Fully end-to-end autonomous driving with semantic depth cloud mapping and multi-agent," in *IV*, 2022.

[142] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *TITS*, 2020.

[143] I. Sobh, L. Amin, S. Abdelkarim, K. Elmadawy, M. Saeed, O. Abdeltawab, M. E. Gamal, and A. E. Sallab, "End-to-end multi-modal sensors fusion system for urban automated driving," in *NeurIPS Workshops*, 2018.

[144] Y. Chen, J. Wang, J. Li, C. Lu, Z. Luo, H. Xue, and C. Wang, "Lidar-video driving dataset: Learning driving policies effectively," in *CVPR*, 2018.

[145] H. M. Eraqi, M. N. Moustafa, and J. Honer, "Dynamic conditional imitation learning for autonomous driving," *TITS*, 2022.

[146] S. Chowdhuri, T. Pankaj, and K. Zipser, "Multinet: Multi-modal multi-task learning for autonomous driving," in *WACV*, 2019.

[147] P. Cai, S. Wang, Y. Sun, and M. Liu, "Probabilistic end-to-end vehicle navigation in complex dynamic environments with multimodal sensor fusion," *RA-L*, 2020.

[148] Q. Zhang, M. Tang, R. Geng, F. Chen, R. Xin, and L. Wang, "Mmfn: Multi-modal-fusion-net for end-to-end driving," in *IROS*, 2022.

[149] H. Shao, L. Wang, R. Chen, S. L. Waslander, H. Li, and Y. Liu, "Reasonnet: End-to-end driving with temporal and global reasoning," in *CVPR*, 2023.

[150] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le, *et al.*, "Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection," in *CVPR*, 2022.

[151] S. Borse, M. Klingner, V. R. Kumar, H. Cai, A. Almuzairee, S. Yogamani, and F. Porikli, "X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation," in *WACV*, 2023.

[152] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR*, 2018.

[153] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *CoRL*, 2022.

[154] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *TETCI*, 2022.

[155] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *arXiv.org*, vol. 2306.17582, 2023.

[156] T. Deruyttere, S. Vandenhende, D. Grujicic, L. Van Gool, and M. F. Moens, "Talk2car: Taking control of your self-driving car," in *EMNLP*, 2019.

[157] P. Mirowski, M. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, A. Zisserman, R. Hadsell, *et al.*, "Learning to navigate in cities without a map," in *NeurIPS*, 2018.

[158] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *CVPR*, 2019.

[159] R. Schumann and S. Riezler, "Generating landmark navigation instructions from maps as a graph-to-text problem," in *ACL*, 2021.

[160] J. Kim, T. Misu, Y.-T. Chen, A. Tawari, and J. Canny, "Grounding human-to-vehicle advice for self-driving vehicles," in *CVPR*, 2019.

[161] S. Narayanan, T. Maniar, J. Kalyanasundaram, V. Gandhi, B. Bhowmick, and K. M. Krishna, "Talk to the vehicle: Language conditioned autonomous navigation of self driving cars," in *IROS*, 2019.

[162] J. Kim, S. Moon, A. Rohrbach, T. Darrell, and J. Canny, "Advisable learning for self-driving vehicles by internalizing observation-to-action rules," in *CVPR*, 2020.

[163] J. Roh, C. Paxton, A. Pronobis, A. Farhadi, and D. Fox, "Conditional driving from natural language instructions," in *CoRL*, 2019.

[164] K. Jain, V. Chhangani, A. Tiwari, K. M. Krishna, and V. Gandhi, "Ground then navigate: Language-guided navigation in dynamic scenes," in *ICRA*, 2023.

[165] D. Shah, B. Osiński, b. ichter, and S. Levine, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *CoRL*, 2023.

[166] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[167] OpenAI, "GPT-4 Technical Report," *arXiv.org*, vol. 2303.08774, 2023.

[168] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv.org*, vol. 2302.13971, 2023.

[169] J. Mao, Y. Qian, H. Zhao, and Y. Wang, "Gpt-driver: Learning to drive with gpt," *arXiv.org*, vol. 2310.01415, 2023.

[170] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end au-

tonomous driving via large language model," *arXiv.org*, vol. 2310.01412, 2023.

[171] H. Shao, Y. Hu, L. Wang, S. L. Waslander, Y. Liu, and H. Li, "Lmdrive: Closed-loop end-to-end driving with large language models," in *CVPR*, 2024.

[172] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, J. Beißwenger, P. Luo, A. Geiger, and H. Li, "Drivelm: Driving with graph visual question answering," in *ECCV*, 2024.

[173] T. Qian, J. Chen, L. Zhuo, Y. Jiao, and Y.-G. Jiang, "Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario," in *AAAI*, 2024.

[174] Z. Yang, X. Jia, H. Li, and J. Yan, "A survey of large language models for autonomous driving," *arXiv.org*, vol. 2311.01043, 2023.

[175] B. Hilleli and R. El-Yaniv, "Toward deep reinforcement learning without a simulator: An autonomous steering example," in *AAAI*, 2018.

[176] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[177] Y. Lee, J.-w. Hwang, S. Lee, Y. Bae, and J. Park, "An energy and gpu-computation efficient backbone network for real-time object detection," in *CVPR Workshops*, 2019.

[178] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[179] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. P. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, *et al.*, "Scaling vision transformers to 22 billion parameters," in *ICML*, 2023.

[180] H. Li, C. Sima, J. Dai, W. Wang, L. Lu, H. Wang, J. Zeng, Z. Li, J. Yang, H. Deng, *et al.*, "Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe," *PAMI*, 2023.

[181] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *ECCV*, 2022.

[182] X. Jia, Y. Gao, L. Chen, J. Yan, P. L. Liu, and H. Li, "Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving," in *ICCV*, 2023.

[183] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," in *ICCV*, 2023.

[184] W. Tong, C. Sima, T. Wang, L. Chen, S. Wu, H. Deng, Y. Gu, L. Lu, P. Luo, D. Lin, *et al.*, "Scene as occupancy," in *ICCV*, 2023.

[185] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "Hdmapnet: An online hd map construction and evaluation framework," in *ICRA*, 2022.

[186] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang, "MapTR: Structured modeling and learning for online vectorized HD map construction," in *ICLR*, 2023.

[187] H. Wang, T. Li, Y. Li, L. Chen, C. Sima, Z. Liu, B. Wang, P. Jia, Y. Wang, S. Jiang, *et al.*, "Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping," in *NeurIPS Datasets and Benchmarks*, 2023.

[188] T. Li, L. Chen, H. Wang, Y. Li, J. Yang, X. Geng, S. Jiang, Y. Wang, H. Xu, C. Xu, *et al.*, "Graph-based topology reasoning for driving scenes," *arXiv.org*, vol. 2304.05277, 2023.

[189] T. Li, P. Jia, B. Wang, L. Chen, K. JIANG, J. Yan, and H. Li, "Lanesegnet: Map learning with lane segment perception for autonomous driving," in *ICLR*, 2024.

[190] G. Wang, H. Niu, D. Zhu, J. Hu, X. Zhan, and G. Zhou, "A versatile and efficient reinforcement learning framework for autonomous driving," *arXiv.org*, vol. 2110.11573, 2021.

[191] A. Behl, K. Chitta, A. Prakash, E. Ohn-Bar, and A. Geiger, "Label efficient visual abstractions for autonomous driving," in *IROS*, 2020.

[192] S.-H. Chung, S.-H. Kong, S. Cho, and I. M. A. Nahrendra, "Segmented encoding for sim2real of rl-based end-to-end autonomous driving," in *IV*, 2022.

[193] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv.org*, vol. 1312.6114, 2013.

[194] M. Ahmed, A. Abobakr, C. P. Lim, and S. Nahavandi, "Policy-based reinforcement learning for training autonomous driving agents in urban areas with affordance learning," *TITS*, 2022.

[195] A. Sauer, N. Savinov, and A. Geiger, "Conditional affordance learning for driving in urban environments," in *CoRL*, 2018.

[196] X. Zhang, M. Wu, H. Ma, T. Hu, and J. Yuan, "Multi-task long-range urban driving based on hierarchical planning and reinforcement learning," in *ITSC*, 2021.

[197] C. Huang, R. Zhang, M. Ouyang, P. Wei, J. Lin, J. Su, and L. Lin, "Deductive reinforcement learning for visual autonomous urban driving navigation," *TNNLS*, 2021.

[198] R. Cheng, C. Agia, F. Shkurti, D. Meger, and G. Dudek, "Latent attention augmentation for robust autonomous driving policies," in *IROS*, 2021.

[199] J. Yamada, K. Pertsch, A. Gunjal, and J. J. Lim, "Task-induced representation learning," in *ICLR*, 2022.

[200] J. Chen and S. Pan, "Learning generalizable representations for reinforcement learning via adaptive meta-learner of behavioral similarities," in *ICLR*, 2022.

[201] Z. Yang, L. Chen, Y. Sun, and H. Li, "Visual point cloud forecasting enables scalable autonomous driving," in *CVPR*, 2024.

[202] M. Henaff, A. Canziani, and Y. LeCun, "Model-predictive policy learning with uncertainty regularization for driving in dense traffic," in *ICLR*, 2019.

[203] J. Wu, Z. Huang, and C. Lv, "Uncertainty-aware model-based reinforcement learning: Methodology and application in autonomous driving," *IV*, 2022.

[204] M. Pan, X. Zhu, Y. Wang, and X. Yang, "Iso-dream: Isolating and leveraging noncontrollable visual dynamics in world models," in *NeurIPS*, 2022.

[205] J. Yang, S. Gao, Y. Qiu, L. Chen, T. Li, B. Dai, K. Chitta, P. Wu, J. Zeng, P. Luo, *et al.*, "Generalized predictive model for autonomous driving," in *CVPR*, 2024.

[206] S. Gao, J. Yang, L. Chen, K. Chitta, Y. Qiu, A. Geiger, J. Zhang, and H. Li, "Vista: A generalizable driving world model with high fidelity and versatile controllability," *arXiv.org*, vol. 2405.17398, 2024.

[207] Y. Wang, J. He, L. Fan, H. Li, Y. Chen, and Z. Zhang, "Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving," in *CVPR*, 2024.

[208] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

[209] A. Hu, G. Corrado, N. Griffiths, Z. Murez, C. Gurau, H. Yeo, A. Kendall, R. Cipolla, and J. Shotton, "Model-based imitation learning for urban driving," in *NeurIPS*, 2022.

[210] R. Caruana, "Multitask learning," *Machine Learning*, 1997.

[211] K. Ishihara, A. Kanervisto, J. Miura, and V. Hautamaki, "Multi-task learning with attention for end-to-end autonomous driving," in *CVPR Workshops*, 2021.

[212] Z. Li, T. Motoyoshi, K. Sasaki, T. Ogata, and S. Sugano, "Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability," *arXiv.org*, vol. 1809.11100, 2018.

[213] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *CVPR*, 2017.

[214] A. Mehta, A. Subramanian, and A. Subramanian, "Learning end-to-end autonomous driving using guided auxiliary supervision," in *ICVGIP*, 2018.

[215] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning to steer by mimicking features from heterogeneous auxiliary networks," in *AAAI*, 2019.

[216] A. Zhao, T. He, Y. Liang, H. Huang, G. Van den Broeck, and S. Soatto, "Sam: Squeeze-and-mimic networks for conditional visual driving policy learning," in *CoRL*, 2020.

[217] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of deep vision-based autonomous driving systems: Review and challenges," *IJCV*, 2022.

[218] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv.org*, vol. 1704.07911, 2017.

[219] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. J. Ackel, U. Muller, P. Yeres, and K. Zieba, "Visualbackprop: Efficient visualization of cnns for autonomous driving," in *ICRA*, 2018.

[220] S. Mohseni, A. Jagadeesh, and Z. Wang, "Predicting model failure using saliency maps in autonomous driving systems," *arXiv.org*, vol. 1905.07679, 2019.

[221] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *ICCV*, 2017.

[222] K. Mori, H. Fukui, T. Murase, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Visual explanation by attention branch network for end-to-end learning-based self-driving," in *IV*, 2019.

[223] D. Wang, C. Devin, Q.-Z. Cai, F. Yu, and T. Darrell, "Deep object-centric policies for autonomous driving," in *ICRA*, 2019.

[224] L. Cultrera, L. Seidenari, F. Becattini, P. Pala, and A. Del Bimbo, "Explaining autonomous driving by learning end-to-end visual attention," in *CVPR Workshops*, 2020.

[225] Y. Xiao, F. Codevilla, D. P. Bustamante, and A. M. Lopez, "Scaling self-supervised end-to-end driving with multi-view attention learning," *arXiv.org*, vol. 2302.03198, 2023.

[226] K. Renz, K. Chitta, O.-B. Mercea, A. S. Koepke, Z. Akata, and A. Geiger, "Plant: Explainable planning transformers via object-level representations," in *CoRL*, 2022.

[227] Y. Sun, X. Wang, Y. Zhang, J. Tang, X. Tang, and J. Yao, "Interpretable end-to-end driving model for implicit scene understanding," in *ITSC*, 2023.

[228] C. Liu, Y. Chen, M. Liu, and B. E. Shi, "Using eye gaze to enhance generalization of imitation networks to unseen environments," *TNNLS*, 2020.

[229] W. Zeng, S. Wang, R. Liao, Y. Chen, B. Yang, and R. Urtasun, "Dsdnet: Deep structured self-driving network," in *ECCV*, 2020.

[230] A. Cui, S. Casas, A. Sadat, R. Liao, and R. Urtasun, "Lookout: Diverse multi-future prediction and planning for self-driving," in *ICCV*, 2021.

[231] Y. Xu, X. Yang, L. Gong, H.-C. Lin, T.-Y. Wu, Y. Li, and N. Vasconcelos, "Explainable object-induced action decision for autonomous vehicles," in *CVPR*, 2020.

[232] H. Ben-Younes, É. Zablocki, P. Pérez, and M. Cord, "Driving behavior explanation with multi-level fusion," *Pattern Recognition*, 2022.

[233] B. Jin, X. Liu, Y. Zheng, P. Li, H. Zhao, T. Zhang, Y. Zheng, G. Zhou, and J. Liu, "Adapt: Action-aware driving caption transformer," in *ICRA*, 2023.

[234] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, 2017.

[235] A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *RA-L*, 2020.

[236] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," *arXiv.org*, vol. 1811.06817, 2018.

[237] A. Filos, P. Tigkas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?," in *ICML*, 2020.

[238] L. Tai, P. Yun, Y. Chen, C. Liu, H. Ye, and M. Liu, "Visual-based autonomous driving deployment from a stochastic and uncertainty-aware perspective," in *IROS*, 2019.

[239] P. Cai, Y. Sun, H. Wang, and M. Liu, "Vtgnet: A vision-based trajectory generation network for autonomous vehicles in urban environments," *IV*, 2020.

[240] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv.org*, vol. 1708.06374, 2017.

[241] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold, "Stochastic model predictive control with a safety guarantee for automated driving," *IV*, 2021.

[242] Y. Lyu, W. Luo, and J. M. Dolan, "Probabilistic safety-assured adaptive merging control for autonomous vehicles," in *ICRA*, 2021.

[243] J. P. Allamaa, P. Patrinos, T. Ohtsuka, and T. D. Son, "Real-time mpc with control barrier functions for autonomous driving using safety enhanced collocation," *arXiv.org*, vol. 2401.06648, 2024.

[244] R. Geirhos, J. Jacobsen, C. Michaelis, R. S. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, 2020.

[245] P. de Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," in *NeurIPS*, 2019.

[246] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. LeCun, "Off-road obstacle avoidance through end-to-end learning," in *NeurIPS*, 2005.

[247] M. Bansal, A. Krizhevsky, and A. S. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," in *RSS*, 2019.

[248] C. Chuang, D. Yang, C. Wen, and Y. Gao, "Resolving copycat problems in visual imitation learning via residual action prediction," in *ECCV*, 2022.

[249] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *NN*, 2018.

[250] J. Byrd and Z. Lipton, "What is the effect of importance weighting in deep learning?," in *ICML*, 2019.

[251] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in *ICML Workshops*, 2003.

[252] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *TCYB*, 2008.

[253] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *CVPR*, 2018.

[254] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2017.

[255] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.

[256] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019.

[257] S. Akhauri, L. Y. Zheng, and M. C. Lin, "Enhanced transfer learning for autonomous driving with systematic accident simulation," in *IROS*, 2020.

[258] Q. Li, Z. Peng, Q. Zhang, C. Liu, and B. Zhou, "Improving the generalization of end-to-end driving through procedural generation," *arXiv.org*, vol. 2012.13681, 2020.

[259] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner,

and E. Wießner, “Microscopic traffic simulation using sumo,” in *ITSC*, 2018.

[260] M. O’Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, “Scalable end-to-end autonomous vehicle testing via rare-event simulation,” in *NeurIPS*, 2018.

[261] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, “Generating adversarial driving scenarios in high-fidelity simulators,” in *ICRA*, 2019.

[262] W. Ding, B. Chen, B. Li, K. J. Eun, and D. Zhao, “Multimodal safety-critical scenarios generation for decision-making algorithms evaluation,” *RA-L*, 2021.

[263] L. Zhang, Z. Peng, Q. Li, and B. Zhou, “CAT: Closed-loop adversarial training for safe end-to-end driving,” in *CoRL*, 2023.

[264] L. T. Triess, M. Dreissig, C. B. Rist, and J. M. Zöllner, “A survey on deep domain adaptation for lidar perception,” in *IV Workshops*, 2021.

[265] Y. You, X. Pan, Z. Wang, and C. Lu, “Virtual to real reinforcement learning for autonomous driving,” in *BMVC*, 2017.

[266] A. Bewley, J. Rigley, Y. Liu, J. Hawke, R. Shen, V.-D. Lam, and A. Kendall, “Learning to drive from simulation without real world labels,” in *ICRA*, 2019.

[267] J. Xing, T. Nagata, K. Chen, X. Zou, E. Neftci, and J. L. Krichmar, “Domain adaptation in reinforcement learning via latent unified state representation,” in *AAAI*, 2021.

[268] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *IROS*, 2017.

[269] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” in *ICRA*, 2018.

[270] J. Matas, S. James, and A. J. Davison, “Sim-to-real reinforcement learning for deformable object manipulation,” in *CoRL*, 2018.

[271] B. Osiński, A. Jakubowski, P. Zięcina, P. Miłoś, C. Galias, S. Homoceanu, and H. Michalewski, “Simulation-based reinforcement learning for real-world autonomous driving,” in *ICRA*, 2020.

[272] R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel, “A survey of zero-shot generalisation in deep reinforcement learning,” *JAIR*, 2023.

[273] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *NeurIPS*, 2017.

[274] P. Karkus, B. Ivanovic, S. Mannor, and M. Pavone, “Diffstack: A differentiable and modular control stack for autonomous vehicles,” in *CoRL*, 2022.

[275] H. Li, Y. Li, H. Wang, J. Zeng, P. Cai, H. Xu, D. Lin, J. Yan, F. Xu, L. Xiong, *et al.*, “Open-sourced data ecosystem in autonomous driving: the present and future,” *arXiv.org*, vol. 2312.03408, 2023.

[276] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, “Segment anything,” in *ICCV*, 2023.

[277] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, “Dinov2: Learning robust visual features without supervision,” *TMLR*, 2024.

[278] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, “Flamingo: a visual language model for few-shot learning,” in *NeurIPS*, 2022.

[279] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, “Training language models to follow instructions with human feedback,” in *NeurIPS*, 2022.

[280] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *CVPR*, 2022.