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 250 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, casual 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. To facilitate future research, we maintain an active repository that contains up-to-date links to relevant 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 deployment strategy, wherein each functionality, such as perception, prediction, and planning, is individually developed and integrated into the onboard vehicle. 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 while driving. 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 provide 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, and others. We conclude by discussing future trends that we think should be embraced by the community to incorporate the latest developments from data engines, large foundation models, and vehicle-to-everything, amongst others.

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, verifiability, and ease of debugging. However, since the optimization goal across modules is different, with detection in perception 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 for the driving system. Moreover, the multi-task, multi-model deployment 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, 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 offer emergent abilities that improve the system by simply scaling training resources.

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

Manuscript received June, 2023.

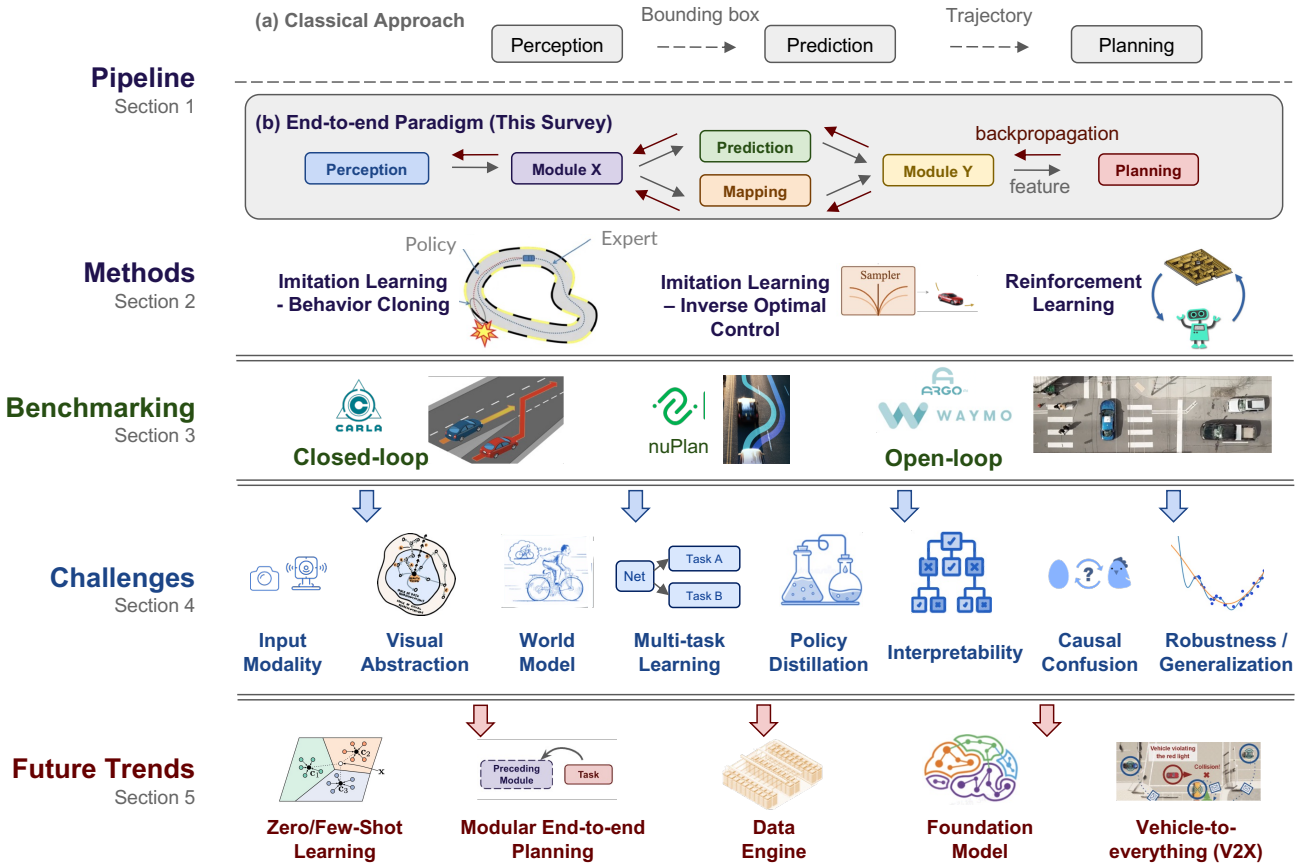

Fig. 1: **Survey at a glance. I. Pipeline and methods.** We define end-to-end autonomous driving as a learning-based algorithm framework with raw sensor input and planning/control output. We classify the 250+ papers surveyed into imitation learning (IL) and reinforcement learning (RL). **II. Benchmarking.** 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. **III. 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. We also comment on promising resolutions to these challenges. **IV. Future trends.** We discuss how end-to-end autonomous driving could benefit from the aid of the rapid development of foundation models, visual pre-training, *etc.* Some graphics are courtesy of online resources.

Note that the end-to-end paradigm does not necessarily indicate one black box with only planning/control outputs. It could be modular with intermediate representations and outputs (Fig. 1 (b)) as in the classical approach. 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 a significant 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. Bojarski *et al.* [8] designed a prototype end-to-end CNN system for both simulation and on-road testing, which reestablished this idea in the new era of GPU computing. 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 the policy of 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 in 2021 for end-to-end autonomous driving. With a diverse range of 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 closed-loop CARLA benchmark [13]. To improve the interpretability and safety of autonomous systems, approaches such as NEAT [11], NMP [32], and BDD-X [33] explicitly incorporate 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 (large) foundation

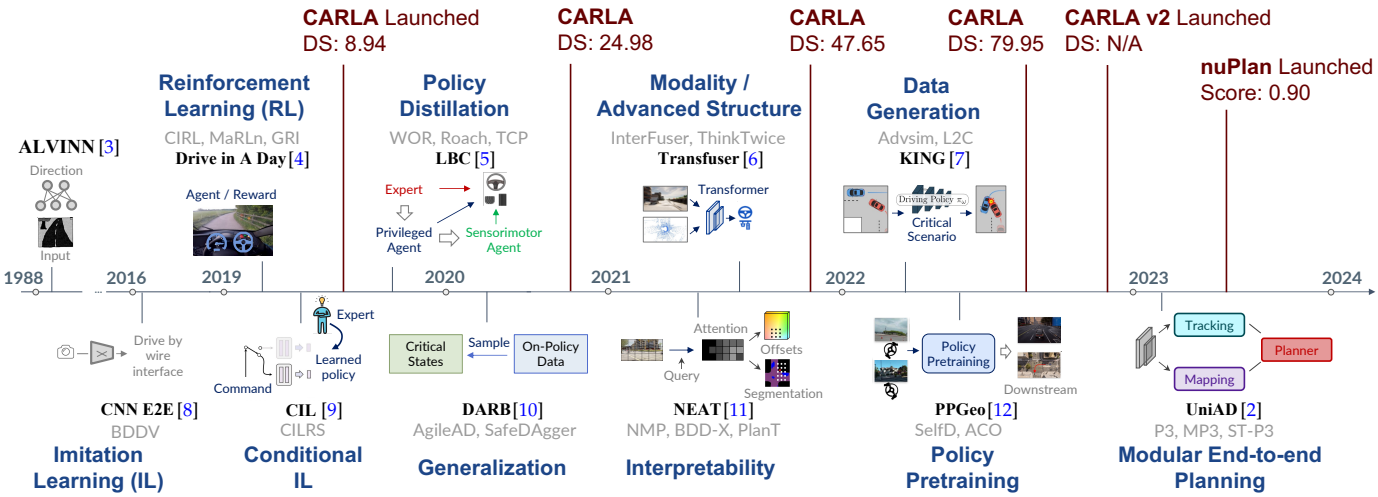

Fig. 2: **Roadmap of End-to-end Autonomous Driving.** We present the key milestones chronologically, grouping similar works under the same theme. The representative or first work is shown in bold with an illustration, while the date of the rest of the literature in the same theme may vary. We also display 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).

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 rest of the prior work focuses 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 and techniques in this field, covering a wide span of topics and providing an in-depth discussion 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 250 papers surveyed, we summarize major aspects and provide in-depth analysis, including topics on generalizability, language-guided learning, casual 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 the fundamental principles behind most existing end-to-end self-driving approaches. Sec. 2.1 discusses methods that use 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 optimal 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 $\{(s_0, a_0, s_1, a_1, \cdots)\}$. The goal of IL is to learn an agent policy π that matches π_β. One important and 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 in the following sections.

2.1.1 *Behavior Cloning*

In behavior cloning, the objective of matching the agent's policy with that of the expert is accomplished by minimizing planning loss as a supervised learning problem 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 tasks [3, 8, 51] utilized an end-to-end neural network to generate control

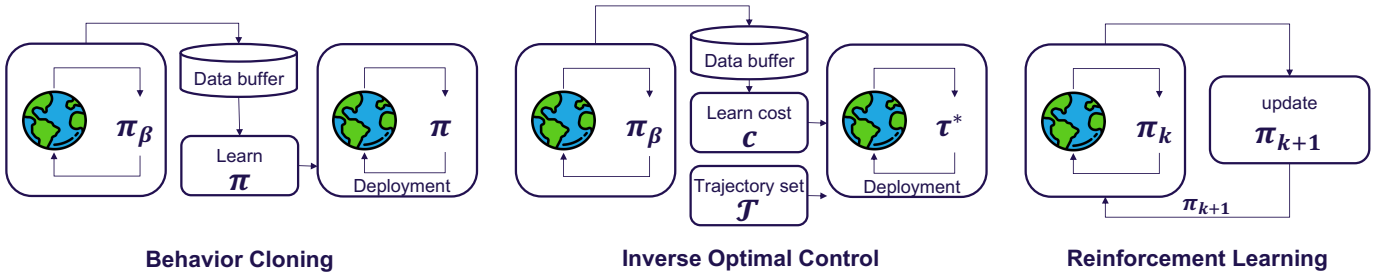

Fig. 3: **Overview of methods in end-to-end autonomous driving.** We illustrate three popular paradigms, including two imitation learning frameworks (behavior cloning and inverse optimal control), as well as online reinforcement learning.

signals from camera inputs. Further enhancements, such as multi-sensor inputs [6, 52], auxiliary tasks [16, 28], and improved expert design [20, 21], have been proposed to enable BC-based end-to-end driving models to handle challenging urban driving scenarios.

Behavior cloning 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 associated with behavior cloning. During training, behavior cloning 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 behavior cloning 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 for imitation learning-based end-to-end autonomous driving are further discussed in Sec. 4.8 and Sec. 4.7, respectively.

2.1.2 *Inverse Optimal Control*

Traditional IOC algorithms learn an unknown reward function $R(s, a)$ in a Markov decision process (MDP) 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 (GAIL) [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 (GANs) [68]. Recently, several works have proposed 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 $c(\cdot)$ in conjunction with other auxiliary tasks, and use simple non-learnable algorithmic trajectory samplers to select the trajectory τ^* with the minimum cost, as illustrated in Fig. 3. Thus, the cost learning paradigm breaks down into two aspects: how to design the cost and how to sample the trajectories to optimize in an end-to-end manner.

Regarding cost design, NMP [32] utilizes a learned cost volume in a bird's-eye-view (BEV). It also conducts object detection in parallel but does not directly link the cost with the detection outputs. Wang *et al.* [69] predict future motion for all agents and use the joint energy as an interactive cost to generate the final plan results. [39, 70, 71] propose estimating a set of probabilistic semantic occupancy or freespace layers as an intermediate representation, which provides explicit clues as to where the vehicle should not maneuver to ensure safety. On the other hand, trajectories are typically sampled from a fixed expert trajectory set [1] 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 still 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. To tackle this, MP3 [1], ST-P3 [38], and IVMP [72] discard the HD Map inputs used in prior work and utilize the predicted BEV map to calculate the cost of traffic rules, such as staying close to centerlines and avoiding collisions with the road boundary. We provide further discussions in Sec. 4.1 and Sec. 4.4. In general, the aforementioned cost learning approaches significantly enhance the safety and interpretability of decisions made by self-driving cars (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] is a field of learning by trial and error. The success of deep Q networks (DQN) [74] in achieving human-level control on the Atari 2600 benchmark [75] has popularized deep reinforcement learning. 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 future reward of that action (when following the same policy afterward). The policy is then implicitly defined by selecting the action with the highest Q value. RL requires an environment that allows potentially unsafe actions to be executed, since it requires exploration

(*e.g.*, sometimes executing random actions during data collection). Additionally, RL requires significantly more data to train than supervised learning. For this reason, modern RL approaches often parallelize data collection across multiple environments [76]. Meeting these requirements in real-world cars presents great challenges. Therefore, almost all papers that use RL in autonomous driving have only investigated the technique in simulation. Most works use different extensions of DQN. As of yet, the community has not converged on a specific RL algorithm.

Reinforcement learning has demonstrated successful learning of lane following [4] on a real car on an empty street. Despite this encouraging early result, it must be noted that a similar task was already accomplished by imitation learning three decades prior [3]. To date, no report has shown results for end-to-end training with RL that are competitive with imitation learning. In a direct comparison conducted in conjunction with the release of the CARLA simulator [77], reinforcement learning fell far behind the modular pipeline and end-to-end imitation learning. It is likely that the reason for this failure is that the gradients obtained via RL are insufficient to train the deep perception architectures (ResNet scale) required for driving. The models used in benchmarks like Atari, where RL succeeds, are relatively shallow, consisting of only a few layers [78].

Reinforcement learning has been successfully applied in autonomous driving when combined with supervised learning. Implicit affordances [18] and GRI [19] both pre-train the CNN encoder part of their architecture using supervised learning with auxiliary tasks like semantic segmentation and classification. In the second stage, the pre-trained encoder is frozen, and a shallow policy head is trained on the implicit affordances from the frozen image encoder with a modern version of Q-learning [79]. Both works report state-of-the-art performance on the CARLA leaderboard [13] at the time of writing. Reinforcement learning has also been successfully used to finetune full architectures on CARLA that were pre-trained using imitation learning [17, 80].

RL has also been effectively applied to planning or control tasks where the network has access to privileged simulator information [48, 81, 82]. In the same spirit, RL has been applied to autonomous driving for dataset curation. Roach [21] trains an RL method on privileged BEV semantic segmentation and uses the policy to automatically collect a dataset with which a downstream imitation learning agent is trained. WoR [20] employs a Q function and tabular dynamic programming to generate additional or improved labels for a static dataset.

A future challenge in the field is to transfer the findings from simulation to the real world. In RL, the objective is expressed as a reward function, and most algorithms require these reward functions to be dense and provide feedback at every environment step. Current works typically use simple objectives, such as progress and collision avoidance, and linearly combine them. These simplistic reward functions have been criticized for encouraging risky behavior [81]. Designing 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], although this presents specific challenges that are discussed in Sec. 4.3. Current RL solutions for autonomous driving rely heavily on low-dimensional representations of the scene, and representation learning is further discussed in Sec. 4.2.

3 Benchmarking

Autonomous driving systems require a comprehensive evaluation of their reliability to ensure safety [86, 87]. To accomplish this, researchers must benchmark these systems using appropriate datasets, simulators, and metrics. This section delineates the two methods of large-scale benchmarking for end-to-end autonomous driving systems: (1) online or closed-loop evaluation in simulation, and (2) offline or open-loop evaluation on human driving datasets. We focus, in particular, on the more principled online setting and provide a brief summary of offline evaluation for completeness.

3.1 Online Evaluation (Closed-loop)

Conducting tests of self-driving systems in the real world is costly and risky. To address this challenge, simulation is a viable alternative [14, 77, 88, 89, 90, 91, 92]. Simulators facilitate rapid prototyping and testing, enable the quick iteration of ideas, and provide low-cost access to a wide range of scenarios. In addition, simulators offer tools for measuring performance reliably and accurately. However, their primary disadvantage is that the results obtained in a simulated environment do not necessarily generalize to the real world (Sec. 4.8.3).

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

3.1.1 Parameter Initialization

Simulation offers the benefit of a high degree of control over the environment, including weather and lighting conditions, maps and 3D assets, and low-level attributes such as the arrangement and pose 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 problem in two ways:

Procedural Generation: Traditionally, initial parameters are hand-tuned by 3D artists and engineers, which we refer to as procedural generation [77, 88, 89, 91]. Each property is typically sampled from a probabilistic distribution with manually set parameters, which is a time-consuming process requiring a lot of expertise. This limits scalability. Nevertheless, this remains one of the most commonly used methods of initialization. Procedural generation algorithms combine rules, heuristics, and randomization to create diverse road networks, traffic patterns, lighting conditions, and object placements [93, 94].

Data-Driven: Data-driven approaches for simulation initialization aim to learn the required parameters. Arguably the simplest data-driven approach to initialization is to sample directly from a log of real-world driving data [14, 92]. In this method, 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 than procedural generation. However, it may not encompass rare or extreme situations that are critical for testing the robustness of the autonomous driving system. 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. These algorithms can then generate novel scenarios that resemble the real world but were not included in the original data [95, 96, 97, 98].

3.1.2 Traffic Simulation

Traffic simulation involves generating and positioning virtual entities within the environment with realistic motion [95, 99]. These entities often include vehicles (such as trucks, 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 the traffic entities [77]. This approach is straightforward to implement, but the resulting motion may not be highly realistic. The most prominent implementation of this concept is the Intelligent Driver Model (IDM) [100]. 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, IDM may be inadequate to capture 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 [95, 97, 99, 101, 102, 103].

3.1.3 Sensor Simulation

Sensor simulation is crucial for evaluating an end-to-end self-driving system. This involves generating simulated raw sensor data, such as camera images or LiDAR scans that the driving system would receive from the environment from different viewpoints in the simulator [104, 105, 106, 107]. This process needs to take into account noise and occlusions in order to realistically assess the autonomous system's performance. There are two branches of ideas in the literature concerning sensor simulation, as described below.

Simulator	Benchmarks
CARLA	CoRL [77], noCrash [118], Town05 [6], LAV [52], Longest6 [28], Leaderboard v1 [13], Leaderboard v2 [13]
nuPlan	Val14 [119], Leaderboard [14]

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

Graphics-Based: Recent computer graphics simulators use 3D models of the environment, along with vehicle and traffic entity models, to generate sensor data via approximations of physical rendering processes in the sensors [77, 89, 91]. For example, this can take into account 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 extremely heavy computation, making parallelization non-trivial [108]. It is closely tied to the quality of the 3D models as well as the approximations used in modeling the sensors. A comprehensive survey of graphics-based rendering for driving data is provided in [109].

Data-Driven: Data-driven sensor simulation leverages and adapts real-world sensor data to create a new simulation where both the ego vehicle and background traffic may move differently from the way they did in the original data [110, 111, 112]. One popular approach is the use of Neural Radiance Fields (NeRF) [113], 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 than graphics-based approaches, but they have limitations such as high rendering times or requiring independent training for each scene being reconstructed [107, 114, 115, 116]. Another approach to data-driven sensor simulation is domain adaptation, which aims to minimize the distribution shift between real and graphics-based simulated sensor data [117]. Machine learning techniques, such as GANs or style transfer, can be employed to improve realism. Further details regarding these ideas can be found in Sec. 4.8.3.

3.1.4 Benchmarks

We provide a succinct overview of driving benchmarks available up to date in Table 1. In 2019, the original benchmark released with CARLA [77] was solved with near-perfect scores [5]. The subsequent NoCrash benchmark [118] involves training on a single CARLA town (Town01) 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 in CARLA while withholding Town05 for testing. Similarly, the LAV benchmark trains on all towns except for Town02 and Town05, which are both reserved for testing to increase the diversity of the test routes. Finally, the Longest6 benchmark [28] expands to a setting with 6 test towns, albeit all seen during training. Two online submission servers for CARLA agents, known as the leaderboard (v1 and v2), are available at [13]. The leaderboard ensures fair comparisons by keeping the evaluation routes confidential. Leaderboard v2 is highly challenging due to the extremely long route length (over 8km on average, as opposed to 1-2km on v1)

and a wide variety of new traffic scenarios. No methods have been benchmarked there yet.

The nuPlan simulator is not currently accessible for end-to-end systems since the sensor data and corresponding sensor simulation aspect (Sec. 3.1.3) have not yet been made public. However, there are two existing benchmarks on which agents directly input maps and object properties available via the data-driven parameter initialization for nuPlan (Sec. 3.1.1). Val14, proposed in [119], uses a publicly accessible validation split of nuPlan [14]. The leaderboard, a submission server for testing on the private test set, was used in the 2023 nuPlan challenge. Unfortunately, this is no longer publicly available for submissions.

3.2 Offline Evaluation (Open-loop)

Open-loop evaluation involves assessing 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 a human driver. Given sensor inputs and goal locations from the dataset as inputs, performance is measured by comparing the system's predicted future trajectory against the trajectory taken by the human 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 [120]. The advantage of open-loop evaluation is that it is easy to implement and 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 by the system 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. Furthermore, the distance between the predicted trajectory and the observed trajectory is not a suitable indicator of performance in a multi-modal scenario. For example, in the case of merging into a turning lane, both the options of merging immediately or later are equally valid, but open-loop evaluation penalizes the option that was not observed in the data. Similarly, the predicted trajectory could depend on observations that will only be available in the future, *e.g.*, stopping at a light that is still green but will turn red soon, a situation that is invalid to evaluate with a single ground truth trajectory.

This approach requires a comprehensive dataset of trajectories to draw from. The most popular datasets for this purpose include nuScenes [121], Argoverse [122, 123], Waymo [124], and nuPlan [14]. All of these datasets comprise a large number of annotated trajectories from real-world driving environments 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 [118, 119, 120]. Overall, a realistic closed-loop benchmarking, if available and applicable, is recommended in future research.

4 Challenges

For each topic/problem illustrated in Fig. 1, we now discuss the related works, current challenges, and promising future trends and opportunities. We start with challenges related to handling different input modalities and formulations 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, causal confusion in Sec. 4.7, and robustness and generalizability in Sec. 4.8.

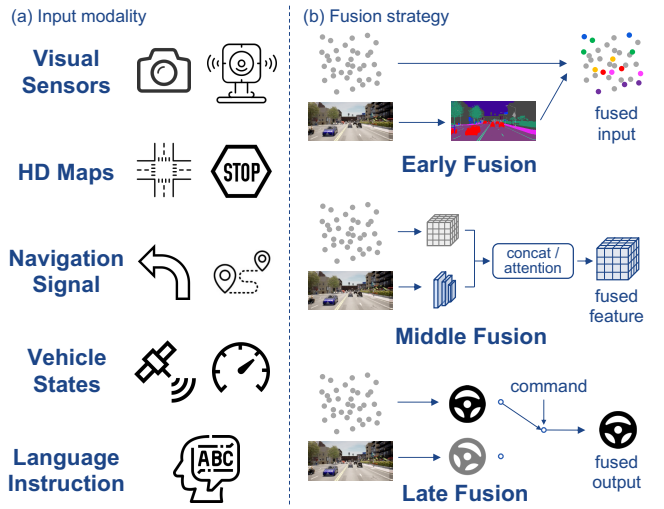

Fig. 4: **Examples of input modality and fusion strategy.** Different modalities have distinct characteristics, leading to the challenge of effectively fusing them and attending to action-critical features. We take point clouds and images as examples to depict various fusion strategies.

4.1 Input Modality

4.1.1 *Multi-sensor Fusion*

Though early works [3, 8] successfully achieved simple self-driving tasks such as lane-following with a monocular camera, this single input modality is insufficient for handling complex scenarios [28]. Therefore, various sensors have been introduced and equipped on recent self-driving vehicles, as shown in Fig. 4. Particularly, RGB images from cameras naturally replicate how humans perceive the world with abundant semantic visual information; LiDARs or stereo cameras provide accurate 3D spatial knowledge. Additionally, vehicle states such as speed and acceleration from speedometers and IMUs, together with high-level navigation commands, are other lines of input that guide the end-to-end system. However, various sensors have distinct perspectives and data distributions, and the large gaps between them present great challenges in effectively fusing them to complement each other for autonomous driving.

Multi-sensor fusion has predominantly been discussed in perception-related fields, *e.g.*, object detection [125, 126], tracking [127, 128], and semantic segmentation [129, 130], and is typically categorized into three groups: early, mid, and late fusion. End-to-end autonomous driving algorithms explore similar fusion schemes [131]. **Early fusion** means combining sensory information before feeding it into the

feature extractor. Concatenation is a common way for fusing various inputs, such as images and depth [132], BEV point clouds and HD maps [32, 133], *etc.*, and then processing them with a shared feature extractor. [134, 135] paint LiDAR points on BEV with the same size as perspective-view images and combine them as inputs. To resolve the view discrepancy, some works attempt to project point clouds on 2D images [136] or attach an additional channel for each LiDAR point by predicting semantic labels from images in advance [52, 137]. On the other hand, the **late fusion** scheme combines multiple results from multi-modalities. It is less discussed due to its inferior performance [6, 131].

Contrary to these approaches, **middle fusion** achieves multi-sensor fusion within the network by separately encoding inputs and then combining them at the feature level. Naive concatenation is also frequently adopted to fuse features from different modalities [15, 22, 30, 138, 139, 140, 141, 142]. Recently, works have employed Transformers [27] to model interactions between pairs of features. Transfuser [6, 28] processes images and LiDAR inputs using two independent convolutional encoders, interconnecting each resolution of features with Transformer encoders, resulting in four-stage feature fusion. Self-attention layers are employed for sensor tokens, attending regions of interest and updating information from other modalities. MMFN [143] further incorporates OpenDrive map and radar inputs on top of Transfurser. [29, 144] take a one-stage Transformer encoder architecture to fuse various features after the last encoder block. The attention mechanism has demonstrated great effectiveness in aggregating the context of different sensor inputs and achieving safer end-to-end driving performance.

Different modalities generally bring increased fields of view and perceptual accuracy, but fusing them to extract critical information for end-to-end autonomous driving requires further exploration. It is essential to model these modalities in a unified space such as BEV [125, 126], identify policy-related context [6, 12], and discard irrelevant sensory information. Besides, fully utilizing the powerful Transformer architecture remains a challenge. The self-attention layer interconnects all tokens to model their areas of interest freely, but it incurs a significant computational cost and cannot guarantee useful information extraction. More advanced Transformer-based multi-sensor fusion mechanisms in the perception field, such as [145, 146], hold promise for application to the end-to-end driving task.

4.1.2 Language as Input

Humans drive cars using both visual perception and intrinsic knowledge, such as traffic rules and desired routes, which together form causal behaviors. In some areas related to autonomous driving, such as robotics and indoor navigation (also called embodied AI), incorporating natural language as fine-grained instructions to control the visuomotor agent has achieved notable progress [147, 148, 149, 150].

However, the outdoor autonomous driving task owns different characteristics compared to indoor robotic applications in the following contexts: (1) The outdoor environment is unknown, and the vehicle cannot explore back and forth. (2) There are few distinctive anchor landmarks, posing great challenges in grounding language instructions. (3) Driving scenarios are much more complex, with a continuous action space and highly dynamic agents. Safety is a top priority during maneuvering. To incorporate linguistic knowledge into driving actions, the Talk2Car [151] dataset provides a benchmark for localizing referred objects in outdoor environments. The Talk2Nav [152], TouchDown [153] and Map2Seq [154] datasets introduce visual language navigation tasks using Google Street View. They model the world as a discrete connectivity graph and require navigating toward the goal in a node selection format. HAD [155] first takes human-to-vehicle advice and adds a visual grounding task with an LSTM-based controller. Sriram *et al.* [156] encode natural language instructions into high-level behaviors including turning left, turning right, not turning left, *etc.*, and verify their language-guided navigation approach in the CARLA simulator. [157, 158] tackle the low-level real-time control problem by attending to the textual action command. Recently, CLIP-MC [159] and LM-Nav [160] utilize CLIP [161] which benefits from large-scale visual language pre-training to extract both linguistic knowledge from instructions and visual features from images. They showcase the advantages of the pre-trained model and provide an attractive prototype for solving complex navigation tasks with multi-modal models.

Despite the successful attempts to use CLIP for landmark feature extraction, the application of large language models (LLMs) such as GPT-3 [162] or instructive language models like ChatGPT [163, 164, 165] in the autonomous driving field remains unclear. Modern LLMs present more opportunities to handle sophisticated linguistic instructions. However, considering their long inference times and instability, it is also crucial to determine the interaction mode for on-road application. Moreover, current language-guided navigation works validate their effectiveness in simulation or specific robot embodiment, and a large-scale benchmark comprising meaningful linguistic prompts is missing.

4.2 Visual Abstraction

An end-to-end autonomous driving system achieves the maneuvering task roughly in two stages: encoding the state space into a latent feature representation, and then decoding the driving policy with the intermediate features. In the case of 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, 19, 166]. Therefore, it is helpful to first pre-train the visual encoder of the network using proxy pre-training tasks. This enables the network to effectively extract useful information for driving, thus facilitating the subsequent policy decoding stage while also meeting memory and model size constraints for all end-to-end algorithms. Furthermore, this can improve the sample efficiency for RL methods.

The process of visual abstraction, or representation learning, often incorporates certain inductive biases or prior information. To achieve a more compact representation than raw images, some methods directly utilize semantic segmentation masks from pre-trained segmentation networks as the input representation for subsequent policy training [167, 168]. SESR [169] takes this a step further by encoding the segmentation masks into class-disentangled representa-

tions through a VAE [170]. In [171, 172], predicted affordance indicators, such as traffic light states, speed, offset to the lane center, hazard indicators, and distance to the leading vehicle, are used as the representation for policy learning.

After observing that segmentation or affordances as a representation can create a bottleneck defined by humans and result in the loss of useful information, some have chosen intermediate latent features from pre-training tasks as effective representations. PIE-G [173] has demonstrated that early layers of the ImageNet [174] pre-trained model can serve as effective representations. [18, 19, 175, 176] employ latent representations pre-trained with tasks including semantic segmentation and/or affordance predictions as input for RL training and achieve excellent performance. In [177], latent features in VAE are augmented by attention maps obtained from the diffused boundary of segmentation and depth maps to highlight important regions. PPGeo [12] proposes to learn effective representation through motion prediction together with depth estimation in a self-supervised way on unlabeled driving videos. TARP [178] utilizes data from a series of previous tasks to perform different tasks-related prediction tasks to acquire useful representations. In [179], the latent representation is learned by approximating the π-bisimulation metric, which is comprised of differences of rewards and outputs from the dynamics model. Besides these pre-training tasks with supervised prediction, unsupervised contrastive learning based on augmented views is adopted in [133]. ACO [36] further adds steering angle discrimination into the contrastive learning structure.

As current methods mainly depend on human-defined pre-training tasks, there inevitably exist possible information bottlenecks in the learned representation, and redundant information unrelated to driving decisions may be included. Thus, how to better extract critical information for driving policy during the representation learning process remains an open question.

4.3 World Model and 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 reinforcement learning typically suffers from the challenge of high sample complexity, which is especially pronounced for tasks like autonomous driving due to the large size of the sample space. 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 explicitly model the world model/environment model, which is composed of transition dynamics and reward function, and the agent can interact with it at a low cost. This is particularly helpful in autonomous driving, as 3D simulators like CARLA are relatively slow.

Modeling the highly complex and dynamic environment in driving is a challenging task. To simplify the problem, Chen *et al.* [20] assume that the world is on rails, factoring the transition dynamics into a non-reactive world model and a simple kinematic bicycle model for the ego vehicle. They enrich the labels of a static dataset by utilizing this factorized world model together with a reward function to optimize for better labels via dynamic programming. In [135], a probabilistic sequential latent model is used as the world model to reduce the sample complexity for RL learning. To address the potential inaccuracy of the learned world model, [180] uses an ensemble of multiple world models to provide uncertainty evaluation. Based on the uncertainty, the imaginary rollout between the world model and the policy agent could be truncated and adjusted accordingly. Motivated by the successful MBRL model Dreamer [83], ISO-Dream [181] considers non-deterministic factors in the environment and decouples visual dynamics into controllable and uncontrollable states. Then, the policy is trained on the disentangled states, considering the uncontrollable factors (such as the motion of other agents) explicitly.

It is worth noting that learning the world model in raw image space is not suitable for autonomous driving. Important small details, such as traffic lights, would easily be missed in the predicted images. To tackle this, MILE [182] incorporates the world model in the BEV semantic segmentation space. It combines world modeling with imitation learning by adopting the Dreamer-style world model learning as an auxiliary task. SEM2 [134] also extends the Dreamer structure but with BEV segmentation maps, and uses RL for training. Besides directly using the learned world model for MBRL, DeRL [176] combines a model-free actor-critic framework with the world model. Specifically, the learned world model provides self-assessments of the current action, which is combined with the state value from the critic to better access the actor's performance.

World model learning or MBRL 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 Multi-task Learning with Policy Prediction

Multi-task learning (MTL) involves jointly performing several related tasks based on a shared representation through separate branches/heads. MTL offers significant computational cost reduction by using a single model for multiple tasks. Additionally, related domain knowledge is shared within the shared model, and the task relationship can be better exploited to improve the generalization ability and robustness of the model [183, 184]. Consequently, MTL is a good fit for end-to-end autonomous driving, where the ultimate policy prediction requires a comprehensive understanding of the current environment.

In contrast to common vision tasks where dense predictions are needed, end-to-end autonomous driving predicts a sparse signal. The sparse supervision here brings a challenge for the input encoder to extract useful information for decision-making. For image input, auxiliary tasks such as semantic segmentation [28, 31, 136, 185, 186, 187] and depth

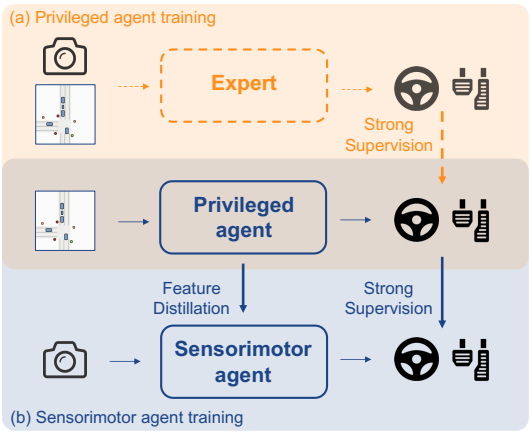

Fig. 5: **Policy distillation.** (a) The **privileged agent** learns a robust policy with access to privileged ground-truth information. The expert is labeled with dashed lines to indicate that it is not mandatory if the privileged agent is trained via RL. (b) The **sensorimotor agent** imitates the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

estimation [28, 31, 185, 186, 187] are commonly adopted in end-to-end autonomous driving models. Semantic segmentation ensures that the model gains a high-level understanding of the scene and recognizes objects of different categories; depth estimation enables the model to understand the 3D geometry of the environment and better estimate distances to critical objects. By performing these tasks, the image encoder can better extract useful and meaningful feature representations for subsequent planning. Besides auxiliary tasks on perspective images, 3D object detection [28, 31, 52] is also useful for the LiDAR encoder. As BEV becomes a natural and popular representation for autonomous driving, tasks such as HD map mapping and BEV segmentation are included in models [11, 23, 28, 29, 30, 31, 52, 144] that aggregate features in BEV space. Moreover, in addition to these vision tasks as multi-tasks, [29, 185, 188] also predict visual affordances including traffic light states, distance to the junction, and distances to opposite lanes, *etc*.

Multi-task learning for end-to-end autonomous driving has demonstrated its effectiveness in improving performance and providing interpretability of the autonomous driving model. However, the optimal combination of auxiliary tasks and the appropriate weighting of their losses to achieve the best performance remain to be explored. Additionally, constructing large-scale datasets with multiple types of aligned and high-quality annotations presents a significant challenge.

4.5 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. Teachers, such as the hand-crafted expert autopilot provided by CARLA, have access to the ground-truth states of surrounding agents and map elements, while students are directly supervised by the collected expert trajectory or control signals with raw sensor input only. This raises great challenges for student models, as they must not only extract perceptual features but also learn a driving policy from scratch.

To address the difficulties above, a few studies propose to divide the learning process into two stages, *i.e.*, training a teacher network and then distilling the policy to the ultimate student network. In particular, Chen *et al.* [5] first employ a privileged agent to learn how to act with direct access to the state of the environment. They then let the sensorimotor agent (student network) closely imitate the privileged agent, with distillation occurring at the output stage. With more compact BEV representations as input for the privileged agent, it provides stronger generalization abilities and supervision than the original expert. The process is depicted in Fig. 5. LAV [52] further empowers the privileged agent to predict the trajectories of all nearby vehicles and distills this ability to the student network which uses vision features.

Apart from directly supervising planning results, several works train their predictive models by distilling knowledge at the feature level. For example, FM-Net [189] employs off-the-shelf networks, including segmentation and optical flow models, as auxiliary teachers to guide feature training. SAM [190] adds L2 feature loss between the teacher and student networks, where the teacher network predicts control signals from ground-truth semantic segmentation maps, and stop intention values. WoR [20] learns a model-based action-value function and then uses it to supervise the visuomotor policy. CaT [23] recently introduces BEV safety hints in the IL-based privileged expert training and distills in BEV space to align features. Roach [21] proposes to train a stronger privileged expert with RL, eliminating the upper bound of imitation learning. It incorporates multiple distillation targets, *i.e.*, action distribution prediction, value estimation, 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.

Though a great amount of effort has been devoted to designing a more robust expert and transmitting knowledge from teachers to students at various levels, the teacher-student paradigm still suffers from inefficient distillation. As shown in all previous works, the visuomotor networks exhibit large performance gaps compared to their privileged agents. For instance, the privileged agent has access to the ground-truth states of traffic lights, which are small objects in images and pose a challenge to distill corresponding features. This may lead to causal confusion for students, as discussed in Sec. 4.7. Thus, it is worth exploring how to draw more inspiration from general distillation methods in machine learning to minimize the gap.

4.6 Interpretability

Interpretability plays a critical role in autonomous driving [191]. It enables engineers and researchers to better test, debug, and improve the system, provides performance guarantees from a societal perspective, increases users' trust, and promotes public acceptance. However, achieving interpretability in end-to-end autonomous driving models, which are often referred to as black boxes, is challenging.

Given a trained autonomous driving model, some post-hoc X-AI (explainable AI) techniques could be applied to the learned model to gain saliency maps [185, 192, 193, 194, 195].

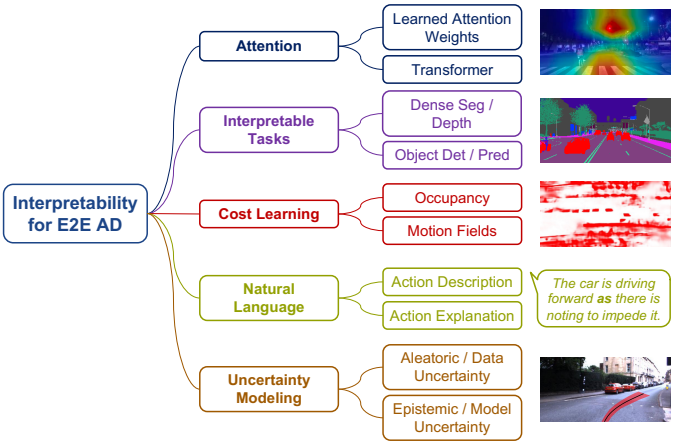

Fig. 6: **Summary of the different forms of interpretability.** They aid in human comprehension of the decision-making processes of end-to-end models, perception failures, and the reliability of the outputs.

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 autonomous driving frameworks that directly enhance interpretability in their model design. We will introduce each category of interpretability in Fig. 6 below.

Attention Visualization: The attention mechanism usually provides a certain degree of interpretability. In [33, 185, 188, 195, 196], a learned attention weight is applied to aggregate important features from the intermediate feature map. Attention weights are learned to adaptively combine ROI pooled features from different object regions [197] or a fixed grid [198]. NEAT [11] iteratively aggregates features to predict attention weights and refine the aggregated feature. Recently, the Transformer attention mechanism has become commonly used in many autonomous driving models. Transformer attention blocks are employed in [6, 11, 28, 29, 31, 143, 199] to better aggregate the information from sensor inputs, and attention maps display important regions in the input for driving decisions. In PlanT [200], the attention layers process features from different vehicles, providing interpretable insights into the corresponding action. Similar to post-hoc saliency methods, although the learned attention maps could offer some straightforward clues about the models' focus, their faithfulness and utility remain limited.

Interpretable Tasks: In a deep driving model, the inputs are initially encoded into intermediate representations for subsequent predictions. Therefore, many IL-based works introduce interpretability by decoding the latent feature representations into other meaningful information, such as semantic segmentation [2, 11, 15, 28, 29, 31, 52, 136, 159, 185, 186, 187], depth estimation [15, 28, 31, 185, 186], object detection [2, 28, 31, 52], affordance predictions [29, 185, 188], motion prediction [2, 52], and gaze map estimation [201]. Although these methods provide interpretable information, most of them only treat these predictions as auxiliary tasks [11, 15, 28, 31, 136, 185, 186, 188], with no explicit impact on the final driving decision. Some [29, 52] do use these outputs for final driving actions, but they are incorporated solely for performing an additional safety check.

Cost Learning: As discussed in Sec. 2.1.2, cost learning-based methods share some similarities with traditional modular autonomous driving systems and thus exhibit a certain level of interpretability. NMP [32] and DSDNet [202] 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. Factors including safety, comfort, traffic rules, and routes based on the perception and prediction outputs are explicitly included [38, 203] to construct the cost volumes.

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] generate the BDD-X dataset, pairing driving videos with descriptions and explanations. They also propose an autonomous driving model with a vehicle controller and an explanation generator, and force spatial attention weights of the two modules to be aligned. BEEF [204] proposes an explanation module that fuses the predicted trajectory and the intermediate perception features to predict justifications for the decision. In [205], a dataset named BDD-OIA is introduced, which includes annotations for both driving decisions and explanations for high-density traffic scenes. Recently, ADAPT [206] proposes a Transformer-based network to jointly estimate action, narration, and reasoning based on driving videos from the BBD-X dataset. Given the recent progress of multi-modality and foundation models, we believe that further combining language with autonomous driving models holds promise for achieving superior interpretability and performance, as discussed in Sec. 4.1.2.

Uncertainty Modeling: Uncertainty is a quantitative approach for interpreting the dependability of model outputs. Since planning results are not always accurate or optimal, it is essential 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 [207], quantitative evaluations of uncertainty for end-to-end autonomous driving systems are conducted, leveraging 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. RIP [208] proposes 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 [142, 209, 210]. With the predicted uncertainty, [209] chooses the output with the lowest uncertainty from multiple outputs, while [142] generates a weighted combination of proposed actions. VTGNet [210] does not directly use the uncertainty for planning but

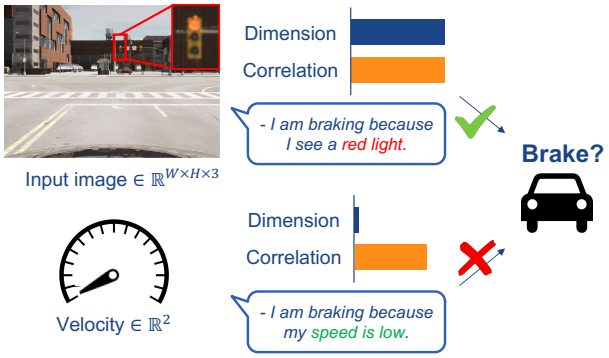

Fig. 7: **Causal Confusion.** The current action of a car is strongly correlated with low-dimensional spurious features such as the velocity or the car's past trajectory. End-to-End models may latch on to them leading to causal confusion.

demonstrates that modeling data uncertainty can improve general performance. 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 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 [211] 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 [212], where access to more information leads to worse performance. One of the earliest reports of this effect was made by LeCun *et al.* [213]. They used a single input frame for steering prediction to avoid such extrapolation. Though simplistic, this is still the preferred solution in current state-of-the-art imitation learning methods [22, 28, 29]. Unfortunately, using a single frame has a downside of not being able to extract the velocity 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 speed of the car is highly correlated with the brake action because the car is waiting for many frames where the speed is zero and the action is the brake. Only at the single frame where the traffic light changes from red to green does this correlation break down.

There are multiple approaches to combat the causal confusion problem when using multiple frames. ChauffeurNet [214] addresses the issue by using intermediate visual abstractions in BEV. One abstraction is the ego agent's past, while other abstractions do not contain this information. During training, the past motion of the ego agent is dropped out with a 50% probability. However, explicit abstractions are required for this approach to work effectively. In [57], the authors attempt to remove spurious temporal correlations from a learned intermediate bottleneck representation by training an adversarial model that predicts the ego agent's past action. This results in a min-max optimization problem where the imitation loss is minimized, and the adversarial loss is maximized. Intuitively, this trains the network to eliminate its own past from its intermediate layers. This approach works well in MuJoCo but does not scale to complex vision-based driving. The first to work on driving complexity is [58]. They propose upweighting keyframes in the training loss. Keyframes are frames where a decision change occurs (and hence are not predictable by extrapolating the past). To find the keyframes, they train a policy that predicts the action with only the ego agent's past as input. PrimeNet [60] improves performance compared to keyframes by using an ensemble, where the prediction of a single-frame model is given as additional input to a multi-frame model. Chuang *et al.* [215] do the same but supervise the multi-frame network with action residuals instead of actions. OREO [59] maps images to discrete codes representing semantic objects and applies random dropout masks to units that share the same discrete code. This helps in Confounded Atari, where the previous action is rendered on the screen. In autonomous driving, the problem of casual confusion can be circumvented by using only LiDAR histories (with a single frame image) and realigning point clouds into the same coordinate system. This removes information about 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.

Causal confusion in imitation learning has been a persistent challenge for nearly two decades. Significant effort has gone into studying this problem in recent years [57, 58, 59, 60, 212, 215]. 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 remains an open problem.

4.8 Robustness

4.8.1 *Long-tailed Distribution*

One important aspect of the long-tailed distribution problem is dataset imbalance, where a few classes make up the majority and many other classes only have a limited number of samples, as shown in Fig. 8 (a). This poses a big challenge for models to generalize to various environments. Various methods tackle this problem with data processing, including over-sampling [216, 217, 218, 219, 220], under-sampling [221, 222, 223], and data augmentation [224, 225, 226]. Besides, weighting-based approaches [227, 228, 229, 230, 231, 232, 233] are also commonly used to mitigate the dataset imbalance problem.

In the context of end-to-end autonomous driving, the issue of long-tailed distributions is particularly severe. Dataset imbalance is especially problematic in driving datasets because most typical 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. To tackle this issue, some works rely on handcrafted scenarios [13, 90, 99, 234, 235] to generate more diverse and interesting data in simulation. LBC [5] leverages the privileged agent to create imaginary supervisions conditioned on different navigational commands. LAV [52] argues that though the ego car for data collection has few accident-prone cases, other agents may have experienced some safety-critical or interesting cases. Therefore, it

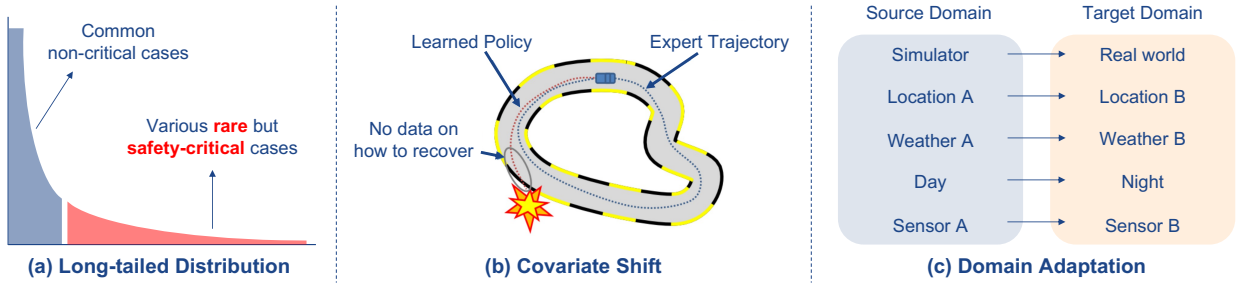

Fig. 8: **Challenges in robustness.** Three primary generalization issues arise in relation to dataset distribution discrepancies, namely long-tailed and normal cases, expert demonstration and test scenarios, and domain shift in locations, weather, *etc*.

includes trajectories of other agents for training to promote data diversity. In [236], a simulation framework is proposed to apply importance-sampling strategies to accelerate of the evaluation of rare-event probabilities.

Another line of research [7, 34, 35, 86, 237] generates safety-critical scenarios in a data-driven manner through adversarial attacks. In [86], 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, while still adhering to physical plausibility, to cause failures and updates the LiDAR accordingly. The recent work 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.

4.8.2 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 the unseen testing environment 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 commonly used method to overcome this issue. DAgger is an iterative training process, in which 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 training dataset by adding examples of how to recover from the 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 and improve safety, 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, and the expert takes over in those dangerous cases. MetaDAgger [56] combines meta-learning with DAgger to aggregate data from multiple environments. LBC [5] adopts DAgger and resamples the data so that samples with higher loss are sampled more frequently. In DARB [10], several modifications are made on DAgger to fit the driving task. 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. It also uses a fixed-sized replay buffer for iterative training to increase diversity and reduce dataset bias.

4.8.3 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 [238]:

- 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 the sensor input.
- Sensor-to-sensor: possible differences in sensor characteristics, *e.g.*, resolution and relative position.

Note that the aforementioned cases often overlap.

VISRI [239] uses a translation network to map simulated images to real images, with segmentation maps serving as the intermediate representation. An RL agent is trained based on the translated simulated images. In [240], domain-invariant feature learning is achieved with image translators and discriminators to map images from two domains into a common latent space. Similarly, LUSR [241] adopts a Cycle-Consistent VAE to project images into a latent representation

comprised of a domain-specific part and a domain-general part, based on which the policy is learned. UAIL [209] targets weather-to-weather adaptation by disentangling the image from different weather conditions into a distinguishable style space and a shared content space with a GAN. In SESR [169], class disentangled encodings are extracted from a semantic segmentation mask to reduce the domain gap between images in simulators and the real world. Domain randomization [242, 243, 244] is also a simple and effective technique for sim-to-real adaptation for RL policy learning in simulators, which is further adapted for end-to-end autonomous driving [167, 245]. It is realized by randomizing the rendering and physical settings of the simulators to cover the variability of the real world during training, and obtain a trained policy with good generalization ability.

Currently, sim-to-real adaptation through source target image mapping or domain-invariant feature learning is the focus for end-to-end autonomous driving. Other domain adaptation cases such as geography-to-geography or weather-to-weather adaptation are handled through the diversity and scale of the training dataset. As LiDAR has become a popular input modality for driving, specific adaptation techniques tailored to the characteristics of LiDAR must also be designed, given that current works mainly concentrate on image-based adaptation. Besides, attention should be drawn to traffic agents' behavior and traffic rule gaps between the simulator and the real world, as current methods only focus on the visual gap in images. Incorporating real-world data into simulation through techniques such as NeRF [113, 246] 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.

5.2 Modular End-to-end Planning

The modular end-to-end planning framework optimizes multiple modules while prioritizing the downstream planning task, which enjoys the advantages of interpretability as indicated in Sec. 4.6. This is advocated in recent literature [2, 247] 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 the occupancy representation is sufficient for detecting general obstacles, or the advantages of opting for BEV segmentation over lane topology for static scene perception.

5.3 Data Engine

The importance of large-scale and high-quality data for autonomous driving can never be emphasized enough. Establishing a data engine with an automatic labeling pipeline [248] 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. 3.1 and promote diversity of data and the generalization ability of models (Sec. 4.8). A data engine would enable autonomous driving models to make consistent improvements.

5.4 Foundation Model

Recent advancements in large foundation models in both language [164, 165, 249, 250] and vision [248, 251, 252, 253] have had a significant impact on various aspects of society. The utilization of large-scale data and model capacity has unleashed the immense potential of AI in high-level reasoning tasks. The paradigm of finetuning [254, 255] or prompt learning [163], optimization in the form of self-supervised reconstruction [256] or contrastive pairs [161], and data pipelining, *etc.*, are all applicable to the end-to-end autonomous driving domain. However, we contend that the direct adoption of LLMs for autonomous driving might seem misaligned between the different goals of the two targets. The output of an autonomous agent often requires steady and accurate measurements, whereas the generative sequence output in language models aims to behave like a human, irrespective of its accuracy. A feasible solution to develop a large autonomous driving model is to train a video predictor that can forecast long-term predictions of the environment, either in 2D or 3D. To perform well on downstream tasks like planning, the objective to be optimized for the large model needs to be sophisticated enough, beyond frame-level perception.

5.5 Vehicle-to-everything (V2X)

Occlusion and obstacles beyond the perceptual range are two fundamental challenges for modern computer vision techniques, which can even pose great difficulties for human drivers when they need to react quickly to crossing agents. Vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X) systems offer promising solutions to address this crux, where information from various viewpoints complements the ego blind spots. Witnessing the progress in information transmission mechanisms for multi-agent scenarios [257, 258, 259, 260, 261], these systems could provide a solution to realize advanced decision intelligence among autonomous vehicles.

6 Conclusion

In this survey, we give 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. Future endeavors to embrace the rapidly developed foundation models and data engines are discussed in the end. 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 that this survey would serve as a starting point to shed new light on this domain.

ACKNOWLEDGMENTS

Li Chen, Penghao Wu, and Hongyang Li were supported by National Key R&D Program of China (2022ZD0160100), Shanghai Committee of Science and Technology (21DZ1100100), and NSFC (62206172). Andreas Geiger and Bernhard Jaeger were 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 was 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 (IMPRS-IS) 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," *arXiv.org*, vol. 2111.08575, 2021.
[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," *arXiv.org*, vol. 2306.07957, 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," *arXiv.org*, vol. 2303.09824, 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 *NIPS*, 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] H. Wang, P. Cai, Y. Sun, L. Wang, and M. Liu, "Learning interpretable end-to-end vision-based motion planning for autonomous driving with optical flow distillation," in *ICRA*, 2021.
[73] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *TNNLS*, 1998.
[74] 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.
[75] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *JAIR*, 2013.
[76] 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.
[77] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *CoRL*, 2017.
[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," *arXiv.org*, vol. 2104.13906, 2021.
[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] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *ICRA*, 2019.
[87] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. A. Funkhouser, "Ibrnet: Learning multi-view image-based rendering," in *CVPR*, 2021.
[88] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software available at http://torcs. sourceforge. net*, vol. 4, no. 6, p. 2, 2000.
[89] 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.
[90] 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.
[91] 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.
[92] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *PAMI*, 2022.
[93] 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.
[94] F. Hauer, T. Schmidt, B. Holzmüller, and A. Pretschner, "Did we test all scenarios for automated and autonomous driving systems?," in *ITSC*, 2019.
[95] 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.
[96] L. Mi, H. Zhao, C. Nash, X. Jin, J. Gao, C. Sun, C. Schmid, N. Shavit, Y. Chai, and D. Anguelov, "Hdmapgen: A hierarchical graph generative model of high definition maps," in *CVPR*, 2021.
[97] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou, "Trafficgen: Learning to generate diverse and realistic traffic scenarios," in *ICRA*, 2023.
[98] S. Tan, K. Wong, S. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "Scenegen: Learning to generate realistic traffic scenes," in *CVPR*, 2021.
[99] S. Suo, S. Regalado, S. Casas, and R. Urtasun, "Trafficsim: Learning to simulate realistic multi-agent behaviors," in *CVPR*, 2021.
[100] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, 2000.
[101] 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.
[102] D. Xu, Y. Chen, B. Ivanovic, and M. Pavone, "Bits: Bi-level imitation for traffic simulation," in *ICRA*, 2023.
[103] 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.
[104] 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.
[105] Z. Yang, Y. Chai, D. Anguelov, Y. Zhou, P. Sun, D. Erhan, S. Rafferty, and H. Kretzschmar, "Surfelgan: Synthesizing realistic sensor data for autonomous driving," in *CVPR*, 2020.
[106] 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.
[107] 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.
[108] A. Petrenko, E. Wijmans, B. Shacklett, and V. Koltun, "Megaverse: Simulating embodied agents at one million experiences per second," in *ICML*, 2021.
[109] 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," *arXiv.org*, vol. 2304.12205, 2023.
[110] 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.
[111] 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.
[112] 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.
[113] 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.
[114] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs," in *CVPR*, 2022.
[115] 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.
[116] Y. Yang, Y. Yang, H. Guo, R. Xiong, Y. Wang, and Y. Liao, "Urbangiraffe: Representing urban scenes as compositional generative neural feature fields," *arXiv.org*, vol. 2303.14167, 2023.
[117] S. R. Richter, H. A. Alhaija, and V. Koltun, "Enhancing photorealism enhancement," *PAMI*, 2023.
[118] F. Codevilla, A. M. Lopez, V. Koltun, and A. Dosovitskiy, "On offline evaluation of vision-based driving models," in *ECCV*, 2018.
[119] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," *arXiv.org*, vol. 2306.07962, 2023.
[120] 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.
[121] 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.
[122] M. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *CVPR*, 2019.
[123] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *NeurIPS*, 2021.
[124] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens,

Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *CVPR*, 2020.
[125] 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.
[126] 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," *arXiv.org*, vol. 2205.13542, 2022.
[127] A. Kim, A. Ošep, and L. Leal-Taixé, "Eagermot: 3d multi-object tracking via sensor fusion," in *ICRA*, 2021.
[128] H.-k. Chiu, J. Li, R. Ambruş, and J. Bohg, "Probabilistic 3d multi-modal, multi-object tracking for autonomous driving," in *ICRA*, 2021.
[129] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, "Sensor fusion for semantic segmentation of urban scenes," in *ICRA*, 2015.
[130] 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.
[131] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *TITS*, 2020.
[132] B. Zhou, P. Krähenbühl, and V. Koltun, "Does computer vision matter for action?," *Science Robotics*, 2019.
[133] 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.
[134] 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," in *NeurIPS Workshops*, 2022.
[135] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *TITS*, 2022.
[136] 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.
[137] O. Natan and J. Miura, "Fully end-to-end autonomous driving with semantic depth cloud mapping and multi-agent," in *IV*, 2022.
[138] 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.
[139] 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.
[140] H. M. Eraqi, M. N. Moustafa, and J. Honer, "Dynamic conditional imitation learning for autonomous driving," *TITS*, 2022.
[141] S. Chowdhuri, T. Pankaj, and K. Zipser, "Multinet: Multi-modal multi-task learning for autonomous driving," in *WACV*, 2019.
[142] 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.
[143] 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.
[144] 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.
[145] 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.
[146] 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.
[147] 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.
[148] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *CoRL*, 2022.
[149] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *TETCI*, 2022.
[150] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *CoRR*, 2023.
[151] T. Deruyttere, S. Vandenhende, D. Grujicic, L. Van Gool, and M. F. Moens, "Talk2car: Taking control of your self-driving car," in *EMNLP*, 2019.
[152] 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.
[153] 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.
[154] R. Schumann and S. Riezler, "Generating landmark navigation instructions from maps as a graph-to-text problem," in *ACL*, 2021.
[155] J. Kim, T. Misu, Y.-T. Chen, A. Tawari, and J. Canny, "Grounding human-to-vehicle advice for self-driving vehicles," in *CVPR*, 2019.
[156] 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.
[157] 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.
[158] J. Roh, C. Paxton, A. Pronobis, A. Farhadi, and D. Fox, "Conditional driving from natural language instructions," in *CoRL*, 2019.
[159] K. Jain, V. Chhangani, A. Tiwari, K. M. Krishna, and V. Gandhi, "Ground then navigate: Language-guided navigation in dynamic scenes," *arXiv.org*, vol. 2209.11972, 2022.
[160] 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.
[161] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.
[162] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *NeurIPS*, 2020.
[163] 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.

[164] OpenAI, "OpenAI: Introducing ChatGPT." https://openai.com/blog/chatgpt, 2022.
[165] OpenAI, "GPT-4 Technical Report," *arXiv.org*, vol. 2303.08774, 2023.
[166] B. Hilleli and R. El-Yaniv, "Toward deep reinforcement learning without a simulator: An autonomous steering example," in *AAAI*, 2018.
[167] 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.
[168] A. Behl, K. Chitta, A. Prakash, E. Ohn-Bar, and A. Geiger, "Label efficient visual abstractions for autonomous driving," in *IROS*, 2020.
[169] 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.
[170] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv.org*, vol. 1312.6114, 2013.
[171] 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.
[172] A. Sauer, N. Savinov, and A. Geiger, "Conditional affordance learning for driving in urban environments," in *CoRL*, 2018.
[173] Z. Yuan, Z. Xue, B. Yuan, X. Wang, Y. Wu, Y. Gao, and H. Xu, "Pre-trained image encoder for generalizable visual reinforcement learning," in *NeurIPS*, 2022.
[174] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.
[175] 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.
[176] 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.
[177] R. Cheng, C. Agia, F. Shkurti, D. Meger, and G. Dudek, "Latent attention augmentation for robust autonomous driving policies," in *IROS*, 2021.
[178] J. Yamada, K. Pertsch, A. Gunjal, and J. J. Lim, "Task-induced representation learning," in *ICLR*, 2022.
[179] J. Chen and S. Pan, "Learning generalizable representations for reinforcement learning via adaptive meta-learner of behavioral similarities," in *ICLR*, 2022.
[180] J. Wu, Z. Huang, and C. Lv, "Uncertainty-aware model-based reinforcement learning: Methodology and application in autonomous driving," *IV*, 2022.
[181] M. Pan, X. Zhu, Y. Wang, and X. Yang, "Iso-dream: Isolating and leveraging noncontrollable visual dynamics in world models," in *NeurIPS*, 2022.
[182] 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.
[183] R. Caruana, "Multitask learning," *Machine Learning*, 1997.
[184] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *NeurIPS*, 2006.
[185] K. Ishihara, A. Kanervisto, J. Miura, and V. Hautamaki, "Multi-task learning with attention for end-to-end autonomous driving," in *CVPR Workshops*, 2021.
[186] 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.
[187] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *CVPR*, 2017.
[188] A. Mehta, A. Subramanian, and A. Subramanian, "Learning end-to-end autonomous driving using guided auxiliary supervision," in *ICVGIP*, 2018.
[189] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning to steer by mimicking features from heterogeneous auxiliary networks," in *AAAI*, 2019.
[190] 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.
[191] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of deep vision-based autonomous driving systems: Review and challenges," *IJCV*, 2022.
[192] 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.
[193] 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.
[194] S. Mohseni, A. Jagadeesh, and Z. Wang, "Predicting model failure using saliency maps in autonomous driving systems," *arXiv.org*, vol. 1905.07679, 2019.
[195] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *ICCV*, 2017.
[196] 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.
[197] D. Wang, C. Devin, Q.-Z. Cai, F. Yu, and T. Darrell, "Deep object-centric policies for autonomous driving," in *ICRA*, 2019.
[198] 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.
[199] 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.
[200] 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.
[201] C. Liu, Y. Chen, M. Liu, and B. E. Shi, "Using eye gaze to enhance generalization of imitation networks to unseen environments," *TNNLS*, 2020.
[202] W. Zeng, S. Wang, R. Liao, Y. Chen, B. Yang, and R. Urtasun, "Dsdnet: Deep structured self-driving network," in *ECCV*, 2020.
[203] A. Cui, S. Casas, A. Sadat, R. Liao, and R. Urtasun, "Lookout: Diverse multi-future prediction and planning for self-driving," in *ICCV*, 2021.
[204] H. Ben-Younes, É. Zablocki, P. Pérez, and M. Cord, "Driving behavior explanation with multi-level fusion," *Pattern Recognition*, 2022.
[205] 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.
[206] 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.
[207] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," *arXiv.org*, vol. 1811.06817, 2018.
[208] 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.
[209] 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.
[210] P. Cai, Y. Sun, H. Wang, and M. Liu, "Vtgnet: A vision-

based trajectory generation network for autonomous vehicles in urban environments," *IV*, 2020.
[211] 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.
[212] P. de Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," in *NeurIPS*, 2019.
[213] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. LeCun, "Off-road obstacle avoidance through end-to-end learning," in *NeurIPS*, 2005.
[214] M. Bansal, A. Krizhevsky, and A. S. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," in *RSS*, 2019.
[215] C. Chuang, D. Yang, C. Wen, and Y. Gao, "Resolving copycat problems in visual imitation learning via residual action prediction," in *ECCV*, 2022.
[216] L. Shen, Z. Lin, and Q. Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *ECCV*, 2016.
[217] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *NN*, 2018.
[218] J. Byrd and Z. Lipton, "What is the effect of importance weighting in deep learning?," in *ICML*, 2019.
[219] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *CVPR*, 2019.
[220] J. Peng, X. Bu, M. Sun, Z. Zhang, T. Tan, and J. Yan, "Large-scale object detection in the wild from imbalanced multi-labels," in *CVPR*, 2020.
[221] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in *ICML Workshops*, 2003.
[222] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *TCYB*, 2008.
[223] D. Devi, B. Purkayastha, *et al.*, "Redundancy-driven modified tomek-link based undersampling: A solution to class imbalance," *Pattern Recognition Letters*, 2017.
[224] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *CVPR*, 2018.
[225] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2017.
[226] H.-P. Chou, S.-C. Chang, J.-Y. Pan, W. Wei, and D.-C. Juan, "Remix: rebalanced mixup," in *ECCV*, 2020.
[227] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *CVPR*, 2016.
[228] Y.-X. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *NeurIPS*, 2017.
[229] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.
[230] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019.
[231] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan, "Equalization loss for long-tailed object recognition," in *CVPR*, 2020.
[232] J. Tan, X. Lu, G. Zhang, C. Yin, and Q. Li, "Equalization loss v2: A new gradient balance approach for long-tailed object detection," in *CVPR*, 2021.
[233] B. Li, Y. Yao, J. Tan, G. Zhang, F. Yu, J. Lu, and Y. Luo, "Equalized focal loss for dense long-tailed object detection," in *CVPR*, 2022.
[234] S. Akhauri, L. Y. Zheng, and M. C. Lin, "Enhanced transfer learning for autonomous driving with systematic accident simulation," in *IROS*, 2020.
[235] 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.
[236] 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.
[237] 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.
[238] 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.
[239] Y. You, X. Pan, Z. Wang, and C. Lu, "Virtual to real reinforcement learning for autonomous driving," in *BMVC*, 2017.
[240] 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.
[241] 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.
[242] 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.
[243] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *ICRA*, 2018.
[244] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *CoRL*, 2018.
[245] 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.
[246] 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.
[247] P. Karkus, B. Ivanovic, S. Mannor, and M. Pavone, "Diffstack: A differentiable and modular control stack for autonomous vehicles," in *CoRL*, 2022.
[248] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv.org*, vol. 2304.02643, 2023.
[249] 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.
[250] S. Narang and A. Chowdhery, "Pathways language model (palm): Scaling to 540 billion parameters for breakthrough performance," 2022.
[251] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," *arXiv.org*, vol. 2211.07636, 2022.
[252] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "Eva-clip: Improved training techniques for clip at scale," *arXiv.org*, vol. 2303.15389, 2023.
[253] 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," *arXiv.org*, vol. 2304.07193, 2023.
[254] 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.
[255] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, *et al.*, "Internimage: Exploring large-scale vision foundation models with deformable convolutions," in *CVPR*, 2023.
[256] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in

CVPR, 2022.
[257] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *ECCV*, 2020.
[258] Y.-C. Liu, J. Tian, C.-Y. Ma, N. Glaser, C.-W. Kuo, and Z. Kira, "Who2com: Collaborative perception via learnable handshake communication," in *ICRA*, 2020.
[259] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *CVPR*, 2020.
[260] J. Cui, H. Qiu, D. Chen, P. Stone, and Y. Zhu, "Coopernaut: End-to-end driving with cooperative perception for networked vehicles," in *CVPR*, 2022.
[261] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," in *ECCV*, 2022.