






# Choose Your Simulator Wisely: A Review on Open-Source Simulators for Autonomous Driving

Yueyuan Li , Wei Yuan , Songan Zhang , Weihao Yan , Qiyuan Shen , Chunxiang Wang , *Member, IEEE*, and Ming Yang , *Member, IEEE*

**Abstract**—Simulators play a crucial role in autonomous driving, offering significant time, cost, and labor savings. Over the past few years, the number of simulators for autonomous driving has grown substantially. However, there is a growing concern about the validity of algorithms developed and evaluated in simulators, indicating a need for a thorough analysis of the development status of the simulators. To address existing gaps in research, this paper undertakes a comprehensive review of the history of simulators, proposes a utility-based taxonomy, and investigates the critical issues within open-source simulators. Analysis of the past thirty years' development trajectory reveals a trend characterized by an increase in open-source simulators and an expansion of their functionality scope. The categorization of simulators based on feature functionalities delineates five primary classes: traffic flow, sensory data, driving policy, vehicle dynamics, and comprehensive simulators. Furthermore, the paper identifies critical unresolved issues in open-source simulators, including concerns regarding the fidelity of sensory data, representation of traffic scenarios, and accuracy in vehicle dynamics simulation, all of which have the potential to undermine experimental confidence. Additionally, challenges in data format inconsistency, labor-intensive map construction processes, sluggish step updating, and insufficient support for Hardware-In-the-Loop testing are discussed as hindrances to experimental efficiency. In light of these findings, the survey furnishes task-oriented recommendations to aid in the selection of simulators, taking into account factors such as accessibility, maintenance status, and quality, while highlighting the inherent limitations of existing open-source simulators in validating algorithms and facilitating real-world experimentation.

**Index Terms**—Autonomous vehicles, simulator, performance evaluation.

## I. INTRODUCTION

IN THE realm of autonomous driving, the importance of simulators has been widely acknowledged within academic and industrial communities. From 2022 to 2023, over 50% of the methods published in this domain were either trained or

tested in simulation environments. This trend is particularly pronounced in driving decision-making tasks, where over 70% of publications underwent simulator-based verification. Compared to real-world scenarios, simulators exhibit the unique capability to swiftly synthesize sensor data on a large scale, replicating extreme weather conditions, unpredictable behaviors of radical drivers, and intricate traffic scenarios. Moreover, simulators offer the unique advantage of replaying traffic accidents without posing safety hazards. Kalra et al. highlighted the impracticality of achieving statistical significance in autonomous vehicle performance validation through real-world testing alone, estimating a staggering 215 billion miles of testing would be required [1]. Addressing this challenge, Shuo et al. presented compelling evidence that introducing proper simulation methodologies can accelerate the evaluation process  $10^3$  to  $10^5$  times faster [2]. Leading technology companies in the autonomous driving industry, including Waymo, Pony.ai, and Didi, emphasize the development and application of simulators [3], [4], [5]. Waymo's remarkable performance, reflected in the Disengagement Report from California between 2020 and 2023 [6], serves as compelling evidence of the critical role that realistic simulators play in achieving outstanding results [3].

Open-source simulators are particularly meaningful in fostering collaboration within the researcher community. They significantly reduce entry barriers for individual researchers and small institutions, especially for those where hardware-based development is financially challenging and creating a new simulation platform is time-intensive [7]. By doing so, open-source simulators alleviate researchers from the burden of repeatedly setting up training workflows and evaluation benchmarks, allowing them to concentrate more on algorithm development. Furthermore, open-source simulators facilitate resource-sharing by establishing standardized data formats and encouraging the sharing of sensor data on a common platform. This collaborative approach will result in significant time and cost savings in data collection efforts [8]. Influential institutions such as the Computer Vision Center and Waymo have recognized the importance of open-source simulators, and thus, they are devoted to developing Carla and Wayax [7], [9]. These endeavors underscore the growing acknowledgment of the value of open-source simulators in advancing research within the field of autonomous driving.

While several surveys have covered various aspects of autonomous driving, there is a distinct gap in the literature regarding a comprehensive exploration of simulators. Despite a considerable increase in open-source simulators in recent years,

Manuscript received 26 December 2023; revised 18 February 2024; accepted 2 March 2024. Date of publication 6 March 2024; date of current version 26 July 2024. This work was supported by the National Natural Science Foundation of China under Grant 62173228, Grant 62203301, Grant 62373250, and Grant U22A20100. (Corresponding author: Ming Yang.)

Yueyuan Li, Weihao Yan, Qiyuan Shen, Chunxiang Wang, and Ming Yang are with the Department of Automation, Shanghai Jiao Tong University, Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China (e-mail: mingyang@sjtu.edu.cn).

Wei Yuan and Songan Zhang are with the Global Institute of Future Technology, Shanghai Jiao Tong University, Shanghai 200240, China.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIV.2024.3374044>.

Digital Object Identifier 10.1109/TIV.2024.3374044

existing surveys in this domain primarily focus on available alternatives. Works such as [10], [11], [12] investigate the functionalities of a few well-known simulators, listing the specific traffic scenarios they support. There remains a critical need for an in-depth and systematic analysis of the evolution, taxonomy, and technical challenges autonomous driving simulators face.

This paper is presented to fill this gap. Readers can regard this survey as a guide for selecting an appropriate open-source simulator tailored to their specific needs while being alerted to the deficiencies threatening simulation-based experiments' validity. Concurrently, developers can utilize this survey as a roadmap that highlights the implementation priority of the urgently needed functionalities and derives inspiration for potential solutions.

The subsequent sections are organized as follows:

Section II reviews existing literature in the autonomous driving field, covering surveys that mention simulators, reviews specific to simulators for autonomous driving, and works discussing issues in simulators for autonomous driving.

Section III overviews the evolution of simulators over the past three decades, proposing a split into the incipient, dormant, and outbreak periods based on development patterns. This section identifies a tendency to develop open-source simulators and expand the simulation task scope.

Section IV introduces a taxonomy categorizing simulators based on their suitability for specific tasks. The influential simulators are classified into five categories: traffic flow simulator, sensory data simulator, driving policy simulator, vehicle dynamics simulator, and comprehensive simulator. Each category is explained, typical functionalities are detailed, and guidelines for selecting appropriate tools are provided.

Section V identifies critical issues in open-source simulators, justifies the necessity to address them, and investigates potential methodologies for improvement. The identified problems, which hazard the validity of experiments, include the degradation of fidelity in sensory data, behavior of traffic participants, and vehicle dynamics. Alongside these are obstacles hindering the effective setup and execution of experiments, including data format inconsistency, labor-intensive map construction, slow step updating, and the absence of an in-built interface for hardware-in-the-loop (HIL) testing.

## II. RELATED WORKS

Most surveys and reviews related to autonomous driving focused on specific tasks [13], [14], [15], algorithm categories [16], [17], [18], and overall system architecture [19], [20]. These reviews admitted that simulators have a wide application in various scenarios. Yurtsever et al. demonstrated the capability of simulators to generate diverse data that captures different weather conditions [19]. The simulator is regarded as a data augmentation approach to enhance the robustness of perception models, enabling them to perform effectively in various scenarios [21]. Similarly, Grigorescu et al. emphasized the importance of simulation for deep learning-based driving decision-making models, as these models require diverse interaction behavior

to improve their decision-making capabilities [16]. This viewpoint is echoed in other reviews focusing on decision-making and motion-planning algorithms [22], [23], [24], [25]. What's more, Tampuu et al. expressed concerns about the fidelity of simulators, highlighting the challenges of transferring decision models trained in virtual environments to the real world [20]. As simulators are not the primary focus of these surveys and reviews, most works merely provide a brief list of available simulation platforms related to their respective themes [26], [27].

It was not until 2020, with the increasing number of open-source simulators, that reviews and discussions focusing on simulators began to emerge. Zhou et al. conducted a detailed survey of seven popular simulators, including CARLA, LGSVL, CarSim, AirSim, Prescan, Matlab/Simulink, and CarMaker. Their work explored the functionalities of these simulators and summarized their respective application scopes [10]. Holen et al. conducted a similar study but expanded the range of simulators, offering a broader perspective on the available options [11]. The simulators mentioned in Kaur et al.'s research were mostly invariant with Zhou's work, with the main difference being the replacement of AirSim with Gazebo [12]. This survey qualitatively describes the performance of the simulators, along with a summary of the desired functionalities and characteristics of simulators. The existing reviews intend to assist researchers in choosing a suitable simulator for their task, so their main content is to investigate the alternatives and check the available functionalities.

Recently, the discussion about the limitations of simulators is gaining attention. Hu et al. stressed the need to narrow the gap between simulation and reality by automating the reconstruction of simulation scenarios and improving the sim2real model transfer efficiency [28]. [29] holds the same opinion and thoroughly reviews the potential solutions for generating driving scenarios. Meanwhile, Zhang et al. identified that enhancing the intelligence of traffic participants' behaviors is a promising avenue to boost simulator performance [30]. Similarly, Siebke et al. emphasized the importance of including human errors in the traffic simulation process to enhance realism [31]. While these existing works successfully pinpoint critical issues within simulators, they tend to focus on isolated specific challenges, creating a void in systematically cataloging the urgent problems that hinder the ongoing development of simulators.

This paper differs from previous works in that it clarifies the relationship between the research tasks and the simulator categories in the ADS, offering task-specific recommendations for open-source simulators. Moreover, the paper conducts a comprehensive analysis of critical issues in existing simulators, accompanied by a compelling rationale emphasizing the urgency for their resolution. In addition to identifying these issues, the paper summarizes the potential solutions, providing valuable insights for addressing the identified challenges.

## III. HISTORY OF SIMULATORS

The history of simulators for autonomous driving provides valuable insights into the technological evolution within this domain. By observing the number of simulators released every

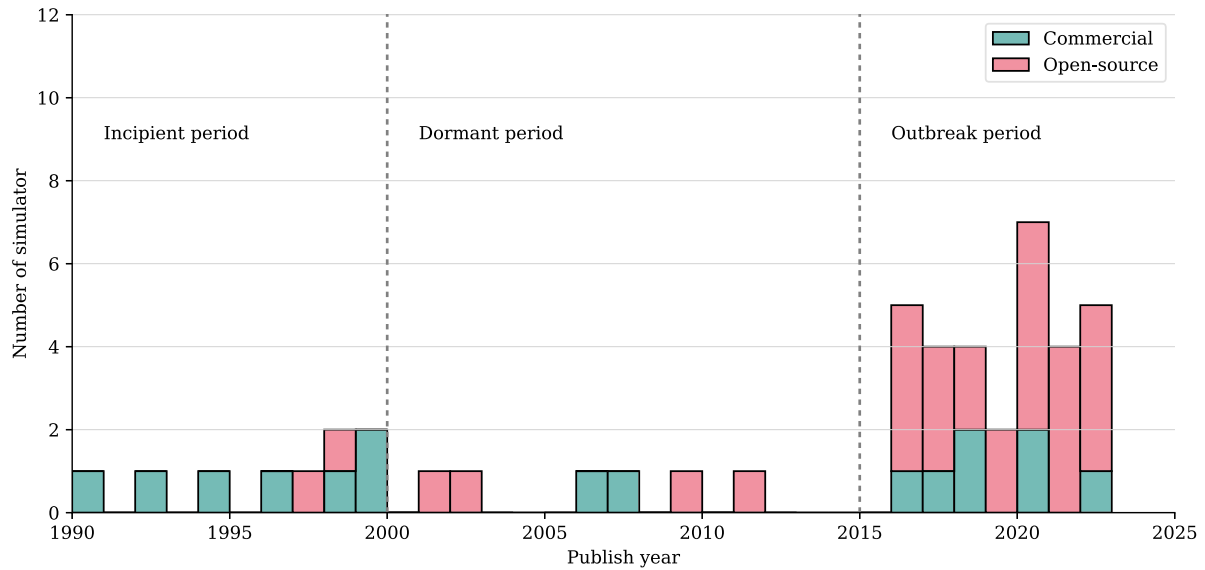


Fig. 1. Number of simulators released by year.

year (Fig. 1), this paper suggests categorizing the development of simulators into three distinct periods: the incipient period, the dormant period, and the outbreak period.

The era before 2000 is characterized as the incipient period, during which simulators began to be deployed to improve the intelligence of vehicles and the transportation system. Notable commercial entries during this time included PTV Vissim by PTV Group in 1992, concentrating on simulations related to traffic flow and logistics system [32], and Paramics in 1994, offering comparable functionalities [33]. Complementing these commercial options, the open-source simulator SUMO emerged as a pivotal force in traffic flow simulation [34]. Simultaneously, in the realm of vehicle dynamics, CarSim and IPG CarMaker, established in the 1990 s, forged enduring partnerships with various vehicle manufacturers [35], [36]. Their reliance on extensive real-world data contributed to their dominance in this domain. RFpro, entering the scene in 2007, stood out as one of the few simulators matching the vehicle dynamics fidelity of CarSim and IPG CarMaker, thanks to real-world data collection from racing cars [37]. During the incipient period, the primary focus of simulators was on two topics: traffic flow and vehicle dynamics. These simulators were developed under well-developed theories, operated on explicit rules and physics models, and demanded relatively modest computational resources.

Between 2000 and 2015, the period witnessed a dormant phase in simulator development, primarily attributed to technological bottlenecks. Shortages in computational resources hindered support for high-performance and large-scale computations. Essential software techniques, including rendering, 3D modeling, and physics engines, were still under development. These factors collectively made the creation of high-fidelity simulators a costly endeavor. Furthermore, the potential applications of simulators remained underexplored during that time. Learning-based methods had yet to demonstrate their potential,

resulting in a lack of urgent demand for generating extensive high-fidelity data. These technological barriers and a dearth of compelling motivations led to a decade-long gap. Only a select few simulators survived and left a lasting impact. Among them were PreScan, supported by TASS International with funding and hardware assistance [38]; rFpro, which benefited from financial backing and data support from Formula 1 (F1) racing groups [37]; and VI-grade, a comprehensive simulation project facilitated by HBK Investment [39]. The survival of these simulators relied on substantial investments from world-leading profit organizations.

A significant growth in the number of simulators began in 2016. Thus, this paper named this the outbreak period. The advent of Neural Networks (NN) boosted the performance of models in many ADS tasks. Learning-based algorithms depend heavily on large-scale data. An image dataset sourced from Grand Theft Auto V (GTA V) underscored the simulators' advantage in synthesizing diverse sensory data [40]. Moreover, simulators provided an ideal environment for training and testing end-to-end ADS, where the costs for collisions and failures are negligible compared to real-world scenarios [41]. Despite the termination of the interface for ADS development by GTA V's manufacturer, the research community embarked on developing independent simulators striving to approximate the real world as much as possible. Microsoft's Airsim, LG's SVL, and CVC's CARLA emerged, designed to generate diverse sensory data and simulate the movement of self-driving vehicles [7], [42], [43]. With an awareness of the computation and time cost of high-fidelity simulators, a trend emerged favoring only providing 2D BEV semantic information to train and verify driving policies [8], [44], [45].

The evolution history of simulators for autonomous driving reveals some trends:

- *Rise of open-source simulators:* The increasing prevalence of open-source simulators might be explained by two

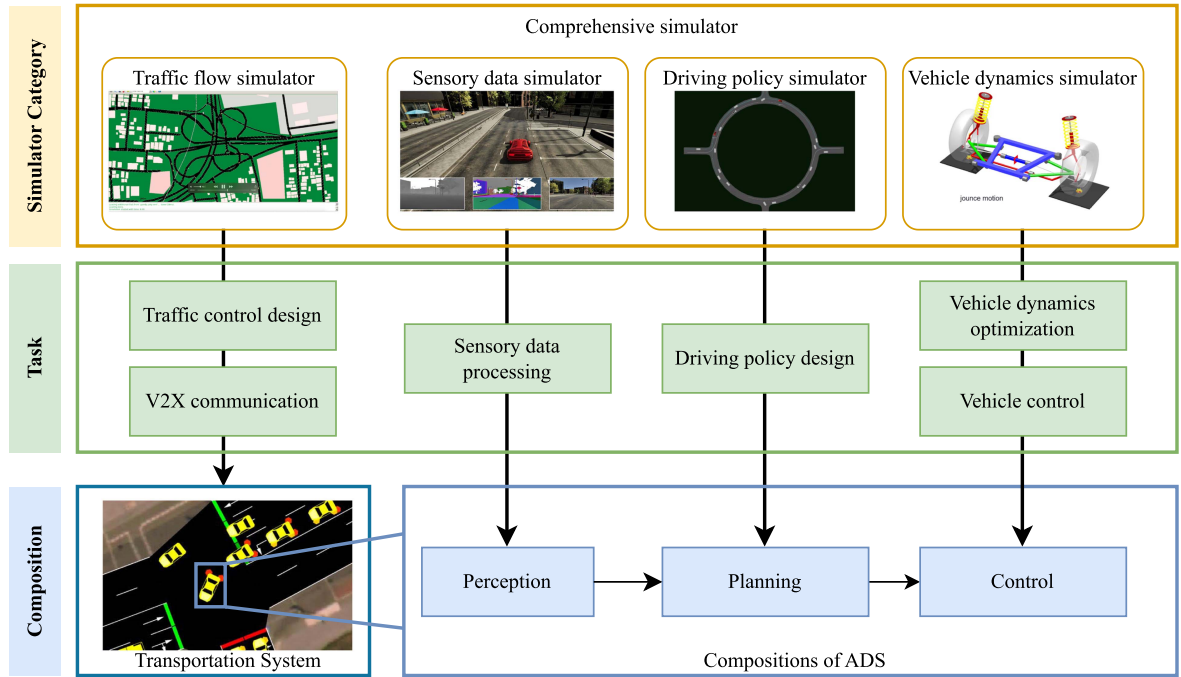


Fig. 2. Relationship between the simulator categories, tasks in the ADS, and the compositions of the ADS [35], [42], [45], [46].

factors. Firstly, technological advancements have made the development of simulators more accessible. The decreasing costs and enhanced availability of computing resources and software tools have made it easier for developers to create and distribute open-source simulators. Secondly, open-source simulators would benefit the entire autonomous driving industry. These platforms might contribute to establishing standardized data formats and evaluation benchmarks. This, in turn, fosters a more cohesive and efficient research environment, facilitating the sharing and validation of findings across the industry.

- *Diversification of task scope:* In the incipient and dormant periods, simulators primarily focused on examining traffic control designs and vehicle structures. However, as the focus in autonomous driving research shifted towards vehicle-centric exploration, simulators assumed a crucial role in the training and testing of algorithms. The expansion in application range has engendered a higher demand for simulation quality, especially in terms of fidelity and diversity.

#### IV. CATEGORIZATION OF SIMULATORS

This section classifies simulators based on their compatibility with distinct ADS tasks. Most simulators can be categorized into five groups: traffic flow simulator, vehicle dynamics simulator, sensory data simulator, driving policy simulator, and comprehensive simulator. It is important to note that specific simulators may exhibit some overlap in functionalities. Each subsection in this section will delineate a specific category and provide practical recommendations for selecting actively maintained open-source simulators for associated tasks.

Table I presents a comprehensive overview of simulators, including their release year, maintenance status, open status, support for HIL testing, and related ADS tasks. This paper identifies *open-source* simulators as those with source codes openly available online. Simulators that distribute their core for free but may charge for technical support or customized functionalities are also considered open-source [7], [69]. Commercial simulators are covered when discussing the history and taxonomy of the simulators, given their dominant position in certain categories. In the *Active Maintenance* column, those simulators without an update in the past year are annotated with a question mark. A simulator is considered no longer actively maintained if it has been out of maintenance for over one year. Considering the proliferation of simulators in recent years, this paper strives to cover relatively influential simulators, defined as those with over 100 citations on Google Scholar or over 100 stars on code-sharing platforms like Github. The table is organized according to the proposed taxonomy. *Support for HIL testing* is contingent upon the explicit statement in the official documentation and the implication by the ecosystem, in which case there is an integration with HIL-supported autonomous driving platforms such as Autoware and Baidu Apollo [77], [78]. The subsequent subsections will present a detailed exploration of the functionalities of actively maintained open-source simulators.

##### A. Traffic Flow Simulator

Traffic flow simulators are software that can simulate the movement of vehicles and other dynamic traffic participants within a transportation system. Typically, a traffic flow simulator may include the following features:



TABLE I  
BASIC INFORMATION OF CATEGORIZED SIMULATORS. FUNCTIONS: ① TRAFFIC CONTROL DESIGN; ② V2X COMMUNICATION; ③ SENSOR DATA PROCESSING; ④ DRIVING POLICY DESIGN; ⑤ END-TO-END DRIVING POLICY DESIGN; ⑥ VEHICLE DYNAMICS OPTIMIZATION; ⑦ VEHICLE CONTROL

Category	Simulator	Released Year	Active Maintenance	Open-source	Support for HIL Testing	Related Tasks
Traffic flow simulator	PTV Vissim [32]	1992	✓	-	-	①
	Paramics [33]	1994	✓	-	-	①
	Aimsun [47]	1999	✓	-	-	① ②
	SUMO [38]	2001	✓	✓	-	① ②
	POLARIS [48]	2016	✓	-	-	① ②
	Flow [49]	2017	-	✓	-	①
	CityFlow [50]	2019	-	✓	-	①
Sensory data simulator	Sim4CV [51]	2016	-	-	✓	③ ⑦
	AirSim [39]	2017	-	✓	✓	③ ④
	Parallel Domain [52]	2017	?	-	-	③
	SVL [45]	2018	-	✓	✓	③ ④
	UniSim [53]	2023	✓	-	-	③
Driving policy simulator	TORCS [54]	1997	-	✓	-	⑤
	VDrift [55]	2011	✓	✓	-	⑤
	CarRacing [56]	2016	✓	✓	-	④
	Udacity [57]	2016	-	✓	-	⑤
	CommonRoad [40]	2017	✓	✓	-	④
	highway-env [46]	2018	✓	✓	-	④
	MACAD [58]	2019	?	✓	-	④
	BARK [59]	2020	?	✓	-	④
	DriverGym [60]	2020	-	✓	-	④
	SMARTS [61]	2020	✓	✓	-	④ ⑤
	SUMMIT [62]	2020	?	✓	✓	④ ⑤
	DI-Drive [63]	2021	?	✓	-	⑤
	L2R [64]	2021	✓	✓	-	⑤
	MetaDrive [65]	2021	✓	✓	-	④ ⑤
	NuPlan [8]	2021	✓	✓	-	④
	InterSim [66]	2022	✓	✓	-	④
	Nocturne [67]	2022	?	✓	-	④
	TBSim [68]	2023	✓	✓	-	④
	Waymax [9]	2023	✓	✓	-	④
Vehicle dynamics simulator	CarSim [35]	1996	✓	-	✓	⑥ ⑦
	Webots [69]	1998	✓	✓	-	⑥ ⑦
	CarMaker [36]	1999	✓	-	✓	③ ⑥ ⑦
	Gazebo [70]	2002	?	✓	✓	⑥ ⑦
	VI-CarRealTime [42]	2009	✓	-	✓	⑥ ⑦
	Matlab [71]	2018	✓	-	✓	⑥ ⑦
Comprehensive simulator	SCANeR Studio [72]	1990	✓	-	✓	③ ④ ⑤ ⑥ ⑦
	Virtual Test Drive [73]	1998	✓	-	✓	③ ④ ⑤ ⑥ ⑦
	PreScan [41]	2006	✓	-	✓	③ ④ ⑤ ⑥ ⑦
	rFpro [37]	2007	✓	-	✓	③ ⑤ ⑥ ⑦
	CARLA [7]	2016	✓	✓	-	③ ④ ⑤ ⑦
	DeepDrive [74]	2018	?	✓	-	③ ④ ⑤ ⑥
	Nvidia Drive Sim [75]	2020	✓	-	✓	③ ⑤ ⑥ ⑦
	Vista [76]	2020	✓	✓	-	③ ⑤
	VI-WorldSim [42]	2020	✓	-	✓	③ ④ ⑤ ⑥ ⑦

- *Road network customization:* A traffic flow simulator enables users to customize the road network by providing detailed information on traffic rules, including speed limits, traffic lights, and road markings. Users can edit lanes, junctions, and traffic rules to tailor the simulation to specific scenarios.
- *Microscopic traffic simulation:* This aspect involves modeling individual vehicles and their interactions. Each vehicle is simulated independently, considering factors such as acceleration, lane following, and lane changing. To accommodate city-level traffic flow simulation, these simulators simplify the behavior models of vehicles.
- *Visualization:* Traffic flow simulators tend to provide a simplified bird-eye-view (BEV) graphical visualization,

illustrating the location and status of traffic participants across the map.

Initially, traffic flow simulators were employed for urban mobility, road infrastructure planning, and traffic control strategy design [32], [33], [49], [50]. Recent advancements include expanding their support to the development of Connected Autonomous Vehicles (CAV) and Vehicle-to-Everything (V2X) technologies [46], [47], [48], reflecting a growing attention to roadside cooperation perception for autonomous driving [79].

SUMO stands out as the only actively maintained open-source option among the available traffic flow simulators. It supports a wide range of map formats and keeps improving its algorithms and hardware efficiency. Together, these make it the recommended choice when working on tasks like transportation

TABLE II  
COMPARISON OF THE SENSORS THAT ACTIVELY MAINTAINED OPEN-SOURCE  
SIMULATORS CAN SYNTHESIZE

Simulator	RGB Camera	Depth Camera	Event Camera	LiDAR	Radar
AirSim	✓	✓	-	-	-
SVL	✓	✓	-	✓	✓
Vista	✓	-	✓	✓	-
CARLA	✓	✓	✓	✓	✓

TABLE III  
COMPARISON OF THE GROUND TRUTH ANNOTATIONS THAT ACTIVELY  
MAINTAINED OPEN-SOURCE SIMULATORS CAN SYNTHESIZE

Simulator	Bounding Box	Semantic Segmentation	Depth Map
AirSim	-	✓	✓
SVL	✓	✓	✓
Vista	-	-	-
CARLA	✓	✓	✓

TABLE IV  
COMPARISON OF THE ADVERSE WEATHERS THAT ACTIVELY MAINTAINED  
OPEN-SOURCE SIMULATORS CAN SYNTHESIZE

Simulator	Night	Cloudy	Rainy	Foggy	Snowy
AirSim	-	-	-	-	-
SVL	✓	-	✓	✓	-
Vista	✓	-	✓	-	-
CARLA	✓	✓	✓	✓	-

planning, traffic control design, road infrastructure design, and V2X and CAV communication.

### B. Sensory Data Simulator

Sensory data simulators can generate highly realistic sensory data and ground truth for perception tasks. To support multi-modal perception tasks, a sensory data simulator must be capable of providing synthesis results from various sensors. One distinct advantage of sensory data simulators is their capacity to expedite data collection under challenging conditions, particularly adverse weather scenarios, thereby saving valuable time [1]. Table II–IV outlines the sensors, ground truth annotations, and adverse weathers each sensory data simulator supports. CARLA is included as it is capable of simulating high-fidelity sensory data.

As is shown in Table I, comprehensive simulators usually cover the functionalities of sensory data simulators. The key difference lies in the absence of quality vehicle dynamics simulation in sensory data simulators, making them unsuitable in end-to-end driving tasks [43], [51], [52].

Notably, most sensory data simulators are no longer actively maintained. This trend may be attributed to the inherent technical challenges and costs of developing high-fidelity sensory data synthesis. As indicated in Table IV, the creation of photo-realistic snowy scenarios remains a considerable challenge, and progress in generating high-fidelity adverse weather conditions is relatively slow [80], [81], [82]. Given this situation,

TABLE V  
COMPARISON OF THE ACTIVELY MAINTAINED OPEN-SOURCE DRIVING POLICY  
SIMULATORS

Simulator	Realistic Perception	Custom Map	Trajectory Importing	Multi- agent
VDrift	✓	-	-	-
CarRacing	-	-	-	-
CommonRoad	-	✓	✓	✓
highway-env	-	-	-	✓
SMARTS	-	✓	-	✓
L2R	✓	✓	-	-
MetaDrive	✓	✓	✓	✓
NuPlan	-	✓	✓	-
InterSim	-	✓	✓	✓
TBSim	-	✓	✓	✓
Waymax	-	✓	✓	✓

researchers engaged in ADS tasks like object detection, semantic segmentation, instance segmentation, depth prediction, and trajectory tracking are advised to avoid complete reliance on simulator-generated data for training and testing purposes. Among the available simulators, CARLA stands out as the most suitable option for synthesizing data [7].

### C. Driving Policy Simulator

Driving policy simulators are software that offers executable traffic scenarios for the development and evaluation of driving policy. The typical functionalities of a driving policy simulator may include:

- *Traffic scenario running*: These simulators enable the execution and visualization of driving policy models within specific traffic scenarios.
- *Traffic status checking*: Typically, driving policy simulators feature a traffic status checking capability, facilitating the validation of driving policy models.
- *Real-world trajectory data import*: Some driving policy simulator offers an interface for importing real-world trajectory data to enhance the fidelity of traffic scenarios.
- *Traffic scenario customization*: This function allows users to customize road layouts and vehicle behavior models to improve the diversity of traffic scenarios.

Table V overviews the most concerned features in actively maintained open-source simulators. *Realistic perception* denotes simulators capable of producing photo-realistic RGB image output; otherwise, they provide a BEV semantic image. *Custom map* represents that the simulator supports users to import maps in some popular data format like OpenStreetMap (OSM) and OpenDRIVE [83], [84]. It is essential to note that most driving policy simulators lack a built-in scenario editor. If the users need to customize their traffic scenario, a third-party scenario editor is necessary. *Trajectory importing* indicates whether the simulator is able to import trajectory data from at least one of the popular open trajectory datasets: HighD and its successors, INTERACTION, NuPlan, and Waymo Open Motion Dataset [8], [85], [86], [87], [88], [89], [90], [91]. *Multi-agent* shows whether the simulator supports multi-agent driving policy models.

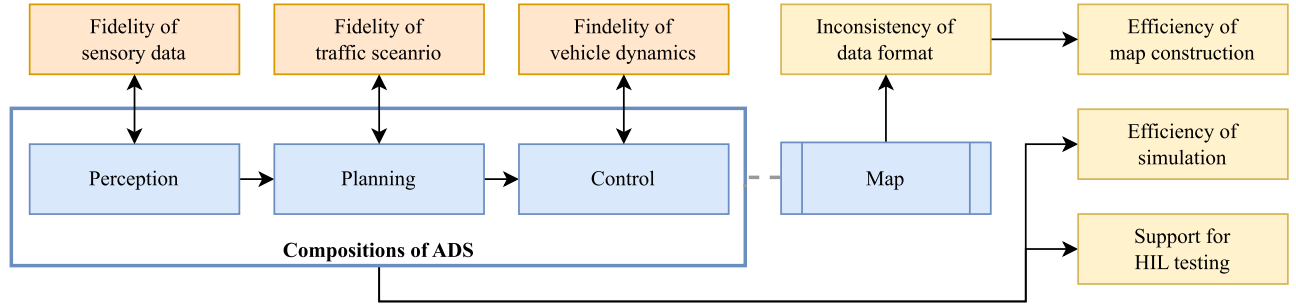


Fig. 3. Critical issues of simulators and their relationship to the compositions of ADS.

There are numerous available driving policy simulators. Readers can refer to Table V and select the most suitable ones based on their requirements. For researchers working on Imitation Learning (IL)-based methodologies, there is no specific requirement to set up the simulator, so they can choose whatever simulator they prefer. Except for VDrift and CarRacing, all the other simulators in Table V have built-in support for RL-based planning algorithms. Simulators offering photo-realistic outputs, such as MetaDrive, are recommended for the end-to-end architecture.

#### D. Vehicle Dynamics Simulator

Vehicle dynamics simulators are software that can replicate the dynamic behavior of vehicles based on physics principles. These simulators consider both internal factors, such as the vehicle's powertrain, suspension, and aerodynamics, and external conditions, including road friction, wind resistance, and slope. By considering these variables, the simulator accurately predicts the dynamic characteristics of vehicles, like acceleration, braking, and steering response to different control commands.

Due to the well-established theories in vehicle dynamics, high-quality vehicle dynamics simulators have existed for a considerable period. However, owing to their reliance on real-world experimental data, the most reliable vehicle dynamics simulators are typically developed for commercial use in close collaboration with car manufacturers [35], [36].

Vehicle dynamics simulators find application in optimizing vehicle design and testing vehicle control algorithms. Among open-source options, Gazebo stands out as a reliable choice due to its rich experience in physics simulation in the robotics domain and its active community support [69]. Additionally, for academic institutions collaborating with MathWorks, Matlab proves to be a considerable choice for its integration capabilities with various simulators [71].

#### E. Comprehensive Simulator

A comprehensive simulator is the integration of the functionalities discussed earlier. A comprehensive simulator can support multiple tasks across the perception, planning, and control parts of the ADS (see Table I). Fig. 2 illustrates how a comprehensive simulator performs Software-In-the-Loop simulation for various ADS tasks.

The development of a reliable, comprehensive simulator necessitates proficiency across multidisciplinary technologies.

The demand for the ability to generate high-fidelity data for every task calls for a substantial commitment of human resources and financial support to the development process. Consequently, most of the qualified comprehensive simulators are for commercial use. The Virtual Test Drive (VTD) is a widely acclaimed comprehensive simulator offering highly integrated functionalities that satisfy diverse needs in developing an autonomous driving system [73]. A similar product, SCAnER, presents comparable attributes, yet its popularity remains relatively limited due to a lack of promotion [72]. The only open-source, comprehensive simulator confirmed to be under maintenance is CARLA [7]. At present, there is no substitute for CARLA in some specific tasks among the open-source simulators.

### V. CRITICAL ISSUES AND IMPROVE SUGGESTIONS

This section will address the unresolved critical issues within simulators. The challenges faced by simulators can be summarized as two general aspects: fidelity and efficiency. Fig. 3 illustrates the relationship between these critical issues and the modules of the ADS, emphasizing their widespread impact on the training and testing processes across various tasks within the ADS framework.

The following subsections will explore the critical issues of the simulators, providing in-depth descriptions and justifications for their significance. Furthermore, the subsections will propose suggestions for improvement by drawing insights from existing literature.

#### A. Fidelity of Sensory Data

1) *Description:* Optical sensors, including cameras, LiDAR, and radar, play a crucial role in interpreting the dynamic traffic environment. Unfortunately, these sensors are vulnerable to adverse light and weather conditions [92]. Factors like illumination, visibility, and humidity fluctuate in different weather, introducing noise into sensor data, so the detection range and precision are compromised [93]. Adverse weather conditions also alter the reflectance of roads, further perturbing the efficacy of perception algorithms [94]. Hence, enhancing the robustness of perception algorithms in diverse light and weather conditions has been a longstanding objective.

The process of collecting sensory data from reality is costly and time-consuming. Given the unpredictable nature of weather,

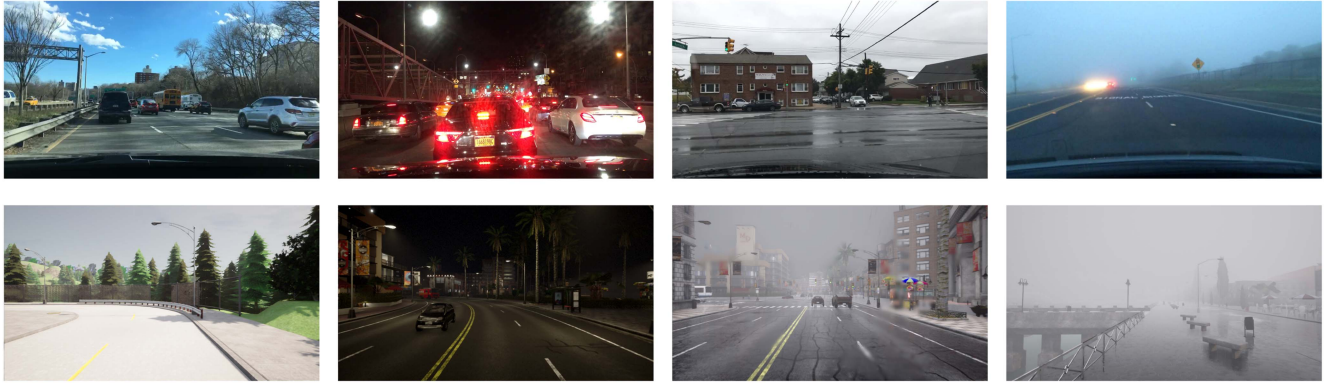


Fig. 4. Comparison of real-world and simulated RGB images in day, night, rainy, and foggy weather. The first row is from BDD100 K dataset, and the second row is from CARLA [7], [96].

it is common for sensory data to be unavailable under demanded conditions. In contrast, simulators offer precise control over weather conditions through adjustable parameters, eliminating the need for extended waits to encounter specific weather scenarios [1]. Moreover, the labor-intensive task of manually annotating ground truth data is obviated. These inherent advantages suggest that if simulators can proficiently generate high-fidelity data in adverse weather conditions, they have the potential to substitute real-world data and promote relevant research [93], [95].

In the context of synthesized sensory data, *fidelity* describes how faithful the synthesized data is to reality. As indicated in Table II, only a few simulators can generate sensory data under complex conditions. Due to technology limitations, the simulators still struggle to present the functionality of adverse weather simulation, with the quality of implementation often overlooked. Fig. 4 shows the fidelity gap between the RGB images obtained from the leading simulator CARLA and the real-world dataset BDD100K [7], [96]. This gap explains why the existing simulators may not be qualified for robust perception model training and testing.

The loss of fidelity of sensory data is closely related to the architecture of simulators. Since these simulators commonly leverage game engines for development, a prevalent practice is to utilize default particle systems within these engines to simulate adverse weather conditions [7], [42], [43]. In a particle system, rain and snow are treated as particles with physics properties. During the simulation process, the emphasis is on the movement trajectory of particles influenced by wind speed and gravity, but less attention is paid to the texture pattern [80], [81], [97]. Furthermore, existing weather simulations often only impact rendering, meaning they can only generate image data for adverse weather without adding corresponding noise interference to LiDAR, radar, etc.

One consequence of the fidelity issue in sensory data is that simulators are limited to providing only ground truth representations of the surrounding and dynamic objects for downstream ADS tasks. Due to insufficient exploration of sensor noise patterns, simulators cannot realistically mimic detection errors. Researchers who evaluate their algorithms using simulated

perception results must be alerted to this simulation-to-reality gap, which poses a hazard to the validity and threatens safety when conducting real-world tests.

2) *Improvement Suggestions*: To improve the fidelity of sensory data simulation, an intuitive idea is to study the realistic noise pattern. In most synthesis processes, the distribution pattern of noise is first extracted, then a mask is generated based on the distribution, and the original data merges with the mask through post-processing. One of the earliest practices involved modeling the noise distribution pattern with particle systems and physical properties [80], [98], [99], which was initially applied to image generation and extended to enhance LiDAR data [82].

In recent years, there has been an increase in learning-based data synthesis methods. This evolution began by employing Deep Neural Networks (DNN) to introduce filter variations, addressing rain, fog, and camera-blurring situations [100], [101], [102]. K-Radar proposed a Neural Network (NN)-based approach to generate adverse weather data specifically for Radar applications [103]. As the quest for improved fidelity and complexity intensified, Adversarial Generative Networks (GAN) were introduced to the synthesis task. GANs enabled the generation of artificial instances of snowing occasions [104], [105]. The study by [106] underscored the potential of GANs in generating RGB images under various lighting conditions, including day, night, and dawn. Tremblay et al. sought to combine physics rules with GAN techniques, allowing for control over precipitation in synthesized images [81]. This synthesis approach successfully maintains visual realism while adhering to fundamental physical laws. ClimateNerf serves as the first attempt to apply NeRF to simulate adverse weather conditions, achieving realistic 3D rendering results, particularly in simulating flood and snow [107].

Sensory data simulation algorithms are often available as open-source projects online [95], [103], [107]. However, these algorithms exhibit variations in environment configurations and hardware requirements, presenting challenges for users seeking quick adoption. If the simulators could establish a standardized interface that defines the input and output of algorithms, it would simplify the process for users and foster a more seamless incorporation of diverse synthesis models into the simulation framework.



## B. Fidelity of Traffic Scenario

1) *Description*: The main components of a traffic scenario are the geometry map, traffic rubrics, and traffic participants. Concerning the static parts, the geometry map, and traffic rubrics, fidelity is not the primary challenge since manual annotation can effectively capture information, including temporary changes. When handling this static data, the efficiency issue is a topic to explore in subsection V-E. The dynamic part of the traffic scenario faces the fidelity problem. Intricate movements and interactions among traffic participants resist explicit depiction. Defined by inherent randomness and individual variations, the dynamic behavior of traffic participants becomes a principal source of complexity of traffic scenarios.

The traffic participants refer to the dynamic objects in a traffic scenario, such as vehicles, cyclists, and pedestrians. Ideally, these participants should be able to perceive their surroundings, comprehend the actions of other participants, and navigate within the scenario [108].

A significant limitation of the ADS simulators lies in the lack of fidelity in traffic participants. Currently, two mainstream methods exist to generate a traffic scenario in simulators. One approach is replicating actual trajectories, resetting the environment when a tested agent violates a rule or experiences a collision [8]. However, this method assumes that the replayed trajectory is an optimal solution, potentially limiting the planners to human-level behavior. Another way is to rely on rule-based behavioral models, which exhibit simplistic behavior patterns [7], [109], [110]. These models are constrained to actions like maintaining a specific distance and stopping, lacking the flexibility to circumvent obstacles or accelerate through scenes [111], [112], [113]. The notable disparity between the behavior of simulated traffic participants and their real-world counterparts undermines the credibility of simulation experiments in planning tasks [31].

2) *Improvement Suggestions*: The key to enhancing the fidelity of dynamic object movements is to refine behavior modeling with realistic random events [23], [114]. Lee et al. proposed to treat the modeling process as a trajectory prediction task, leveraging Graphical Neural Networks (GNN) to learn patterns from real-world datasets [115]. [116] followed this idea and implemented TrafficGraphNet, capable of temporal interactions in a sparse traffic scenario. Similarly, VectorNet and InterSim employ deeper and customized NN to extract motion patterns from datasets [66], [117].

To alleviate a dependency on an initial dataset, researchers have explored the concept of adversarial to interaction generation. RouteGAN applied GAN to produce styled behavior for individual vehicles [118]. Soon, the researchers realized that inverse Reinforcement Learning (RL) structure could adopt a similar idea while better fitting in the time-sequential decision-making [119]. A deep deterministic policy gradient was tried to introduce complexity in lane-changing scenarios [120]. Trajgen pointed out that the combination of trajectory prediction guarantees fidelity, while the RL method adds flexibility to achieve a realistic and diverse trajectory output [30].

Generating interactive traffic participants remains a prevalent research topic [29]. The researchers are developing various

TABLE VI  
COMPARISON OF THE PHYSICS MODEL/DYNAMICS MODEL OF THE ACTIVELY MAINTAINED OPEN-SOURCE SIMULATORS

Simulator	Physics Engine	Bicycle Model
SUMO	-	-
Webots	Open Dynamics Engine [122]	-
VDrift	Custom	-
CarRacing	Box2D [123]	-
CommonRoad	-	✓
highway-env	-	✓
SMARTS	PyBullet [124]	✓
L2R	Unreal Engine 4 [125]	-
MetaDrive	Panda3D [126]	-
NuPlan	-	✓
InterSim	-	✓
TBSim	-	✓
Waymax	-	✓
CARLA	Unreal Engine 4	-

architectures to elevate the sophistication and realism of the generated behaviors. In this context, empowering open-source simulators to provide a user-friendly interface for customizing traffic participants could prove beneficial. By equipping simulators with a reliable traffic event detector and a framework for tailoring behavior models, the research community can facilitate realizing more interactive traffic participant controllers through collaborative contributions.

## C. Fidelity of Vehicle Dynamics

1) *Description*: A reliable vehicle dynamics model is critical to guarantee that the planning result of the ADS is executed as expected. The fidelity of vehicle dynamics is a particularly prevalent problem in open-source simulators, while it has been well resolved in commercial vehicle dynamics simulators. This discrepancy may arise from the demanding nature of constructing an accurate vehicle dynamics model, which necessitates substantial measurement data, particularly under extreme conditions such as sharp turns and sudden stops at high speeds combined with diverse weather and road conditions [121]. Undertaking this endeavor is costly and raises safety concerns, making it challenging to accomplish without collaborative support from vehicle manufacturers.

Table VI investigates the implementation status of vehicle physics models in actively maintained open-source simulators. Half of the simulators integrate a third-party physics engine [122], [123], [124], [125], [126]. The rest choose to implement a dynamics/kinematics bicycle model independently. Most of these physics models are faithful under conditions where velocity, acceleration, and steering angle values are small [127]. In extreme scenarios, including high-speed maneuvers, sharp turns, and emergency braking, the prediction result of bicycle models often diverges significantly from the real world. Therefore, researchers seeking to implement their models in actual vehicles should not rely solely on open-source simulators. Instead, simulations conducted within commercial vehicle dynamics simulators, such as CarMaker, remain indispensable for ensuring robustness and accuracy. The identified limitations in

TABLE VII  
COMPARISON OF THE PUBLIC MAP AND SCENARIO DATA FORMATS THAT THE ACTIVELY MAINTAINED OPEN-SOURCE SIMULATORS SUPPORT

Simulator	Map Data Format				Scenario Data Format	
	SUMO Road Network [128]	OpenStreetMap [83]	OpenDRIVE [84]	Lanelet2 [129]	OpenSCENARIO [130]	Scenic [131]
SUMO	✓	✓	✓	-	-	-
Webots	✓	✓	-	-	-	-
VDrift	-	-	-	-	-	-
CarRacing	-	-	-	-	-	-
CommonRoad	✓	✓	✓	✓	✓	-
highway-env	-	-	-	-	-	-
SMARTS	✓	-	-	-	-	-
L2R	-	-	-	-	-	-
MetaDrive	-	-	-	-	-	-
NuPlan	-	-	-	-	-	-
InterSim	-	-	-	-	-	-
TBSim	-	-	-	-	-	-
Waymax	-	-	-	-	-	-
CARLA	-	-	✓	-	✓	✓

vehicle dynamics within existing open-source simulators underscore the urgent need for continued improvement and increased collaboration in sharing vehicle testing data.

2) *Improvement Suggestions*: Creating a realistic vehicle dynamics model relies on real-world data fraught with cost and safety concerns, so it is hard for open-source simulators to achieve high-fidelity representations. A possible solution lies in establishing a platform where researchers and institutions can contribute locally collected real-vehicle dynamics data. A similar data-sharing endeavor has successfully occurred in structured maps and vehicle 3D models, which suggests the feasibility of such an approach [70], [83].

#### D. Inconsistency of Data Format

1) *Description*: Most open-source simulators employ a self-defined data format to record maps and scenarios. Suppose the publicly available data format is defined as one that has an open document that describes all labels and annotations clearly. In that case, it becomes evident from Table VII that actively maintained open-source simulators provided limited support for such data formats [83], [84], [128], [129], [130], [131]. The inconsistency in map and scenario formats is problematic, as it necessitates redundant efforts in manually constructing traffic scenarios and maps, even when the data is already accessible. The absence of an official data conversion interface also introduces errors during data transition.

2) *Improvement Suggestions*: The underlying problem behind the inconsistency of data format is that the existing data structures fail to fulfill the need to express the scenarios comprehensively and efficiently. The advantages and shortcomings of the map data format are as follows:

- *SUMO Road Network* is one of the earliest publicly available map data formats [128]. It emphasizes the relationship between roads and junctions and provides rich annotation for traffic rule restrictions. However, its utility focuses on traffic flow simulation tasks, leading to a lack of detailed geometry information concerning roads and irregular static obstacles.
- *OSM* originates from a free mapping project [83]. Despite its inclusion of support for static obstacles and regions,

the fundamental structure of OSM draws inspiration from road network representation. Consequently, it continues to employ a centerline to represent lanes, thereby falling short in high-resolution geometry descriptions of lanes.

- *OpenDRIVE* has gained popularity due to its rich parameters and clear documentation [84]. OpenDRIVE excels in recording high-resolution map geometry but overlooks traffic control and road semantic information.
- *Lanelet2* tempts to integrate the strong points of the data formats mentioned above together [129]. It accurately describes geometry structures and versatile, customizable semantic traffic contexts. However, it has not attracted as much support as OpenDRIVE, potentially because Lanelet2 lacks a corresponding scenario data format to articulate the dynamic objects.

For scenario data formats, *OpenSCENARIO* provides a precise though intricate description of dynamic objects [130]. Each traffic participant strictly runs on the assigned trajectory in this format, ensuring a detailed representation of scenarios. In contrast, *Scenic* emphasizes introducing randomness into scenarios [131]. It enables the generation of traffic scenarios by illustrating the probability distribution of behaviors.

Upon examining the current landscape of data formats, it becomes evident that the roadmap for enhancing map data formats involves combining lane-level representation accuracy with comprehensive traffic control and semantic information. Simultaneously, improvements in scenario data formats should accommodate both accurate scenario replay and flexible random movements. A seamless integration between map data and scenario representation demands further attention and dedication. Furthermore, developing an official implementation for a data parser is deemed necessary.

#### E. Efficiency of Map Construction

1) *Description*: The simulators for autonomous driving are expected to have the ability to quickly rebuild or synthesize diverse traffic scenarios for training and testing purposes [29]. While the dynamic objects in traffic scenarios struggle with the fidelity issue (as discussed in Section V-B), the static maps' construction also faces efficiency challenges. Fig. 5 demonstrates

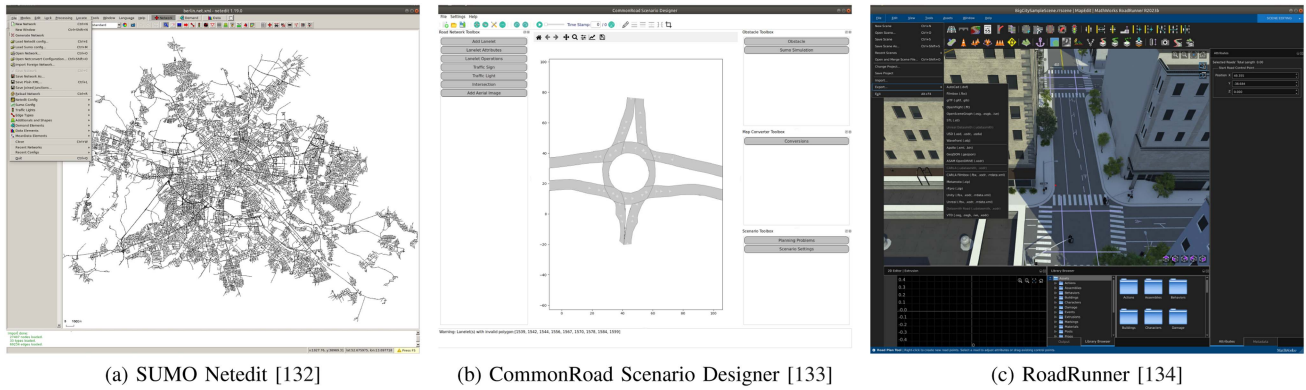


Fig. 5. Graphical User Interface (GUI) of the scenario editors.

the popular scenario editors [132], [133], [134]. Currently, the preferred map reconstruction process requires manually extracting road structures, labeling road network relationships, and annotating traffic signals from LiDAR points and images [135]. This approach heavily relies on handcraft and is inherently time-consuming.

Another challenge in efficiently creating high-quality maps is related to data formats. The common output of 3D map reconstruction, a 3D mesh model, often lacks detailed structural information about roads and buildings [136], [137]. Moreover, semantic annotations for elements like signs and signals are not inherently present in 3D mesh models. Meanwhile, though 2D map reconstruction approaches provide valuable data on road structure, seamlessly integrating the outputs of 2D reconstruction with 3D map reconstruction poses challenges. This is primarily because the 2D map format typically uses polygons to depict geometric structures, while 3D models utilize triangular meshes [83], [129], [138]. Consequently, aligning the 2D road structure with the 3D mesh model becomes complex.

2) *Improvement Suggestions*: Automated map reconstruction utilizes algorithms to process 2D pixels and 3D points and generate structured maps. Achieving partial or complete automated map construction can significantly reduce manual effort and improve the efficiency of building scenarios.

Automated map reconstruction is advancing along two distinct trajectories. One avenue is oriented towards 3D reconstruction, characterized by an emphasis on achieving high-level realism and details. The general idea is to extract features from 2D images and match them with 3D points and pose information. By establishing correlations between object models and textures, this method enables effective automatic map reconstruction [139], [140], [141]. Traditional 3D map reconstruction methods use explicit priors to process large-scale raw data rapidly [141], [142]. The new rising learning-based methods extract data-driven priors by neural networks to handle noise and data sparsity [137], [143]. A recent breakthrough in this area is the NeRF algorithm. It learns the implicit features of space from images captured from different angles, representing 3D space as a continuous radiance field [144], [145]. NVIDIA has implemented 3D surface reconstruction and rendering based

on the NeRF algorithm, demonstrating its great potential for automated 3D map reconstruction [146].

Another avenue focuses on reconstructing the road structure and traffic information, exhibiting a greater tolerance for deviation in the map's geometry details [147]. These techniques prioritize comprehensively depicting the traffic environment, encompassing traffic elements such as the road network, traffic signals, and dynamic objects [108], [148]. With the overarching structure and interplay dynamics of the traffic scenario, the 2D maps generated by these approaches become preferable for driving decision-making algorithms [117], [149].

These two categories of automated map reconstruction methods can reduce the need for manual intervention in constructing simulation scenarios for autonomous driving [65]. However, they are unsuitable for practical implementation at present. The 3D map construction techniques exhibit sensitivity to data distribution and susceptibility to noise interference [148]. The precision and speed of these methods fall short of the standard for industrial application [144].

## F. Efficiency of Simulator

1) *Description*: Training and testing data-driven ADS tasks are time-consuming due to the need for algorithms to cover a variety of scenarios, ensuring that challenging corner cases are addressed [150]. The estimated 215 billion miles for validating an ADS highlights the necessity to apply a simulator to speed up this process [1]. There are several common strategies to improve simulation efficiency:

- *Hardware acceleration* accelerates the simulation process by providing more powerful computational resources. An upgrade to the CPU, GPU, and memory devices could expedite the computational operations and strengthen the capability to handle data batches [9].
- *Computation optimization* is to utilize the computational resource better. It streamlines the simulation workflow by fine-tuning the algorithm, code structure, and cache management [151].
- *Parallelization* maximize the utility of the computational resource at a lower level. It enables simultaneous execution

TABLE VIII  
COMPARISON OF THE HARDWARE REQUIREMENTS AND PERFORMANCE OF THE ACTIVELY MAINTAINED OPEN-SOURCE-SOURCE SIMULATORS

Simulator	Minimal Hardware Requirements			Cross Platform			Computation Accelerating	Distributed Computing	Headless Simulation
	Disk space	Memory	GPU	Windows	Linux	MacOS			
SUMO	512 MB	128 MB	-	✓	✓	✓	-	-	✓
Webots	512 MB	2 GB	512 MB	✓	✓	✓	✓	-	✓
VDrift	1 GB	512 MB	-	✓	✓	✓	-	-	-
CommonRoad	32 MB	2 GB	-	✓	✓	✓	-	-	-
highway-env	1 MB	128 MB	-	✓	✓	✓	-	-	-
SMARTS	8 GB	2 GB	-	✓	✓	✓	-	✓	-
MetaDrive	1.5 GB	3GB	1GB	✓	✓	✓	-	✓	✓
NuPlan	7 GB	128 MB	-	✓	✓	✓	-	-	-
InterSim	512 MB	256 MB	-	✓	✓	✓	-	-	-
TBSim	5 GB	7 GB	-	✓	✓	✓	-	-	-
Waymax	3 GB	128 MB	-	✓	✓	✓	✓	✓	-
CARLA	20 GB	8 GB	8GB	✓	✓	-	-	-	✓

of multiple threads. This approach would significantly reduce the simulation time by distributing the workload across multiple cores or processors [152].

- *Distributed computation* deploys large-scale computation resources over different machines. Running multiple simulators concurrently for training or testing individual models minimizes the overall running time, enhancing the efficiency of the process [152].

Additionally, *headless simulation*, also known as off-screen simulation, may become an unavoidable consequence for achieving efficient simulation, as powerful hardware and distributed computation often depend on servers without display.

A field investigation of the actively maintained open-source simulator shows that most simulators lack the necessary features for high-efficiency simulation. Table VIII provides insights into the minimal hardware requirements for installation disk space, the memory occupancy during tutorial running, and the GPU memory occupancy, which reflects how well the simulator architecture has been optimized. According to the official documentation, most simulators are now cross-platform compatible, but few of them support hardware acceleration, distributed computation, and headless simulation.

2) *Improvement Suggestions*: The efficiency of the simulators is closely related to the software architecture and code quality. It is crucial to make thoughtful choices of the back-end physics engine and programming language at the design stage. Some physics engines have rendering library dependencies incompatible with headless and computational functionalities. Moreover, choosing a high-level programming language could slow down computational speed. Implementing a Continuous Integration (CI) workflow is imperative during the development process, which automatically checks performance and promptly identifies coding issues, ensuring a smoother and more reliable development pipeline.

### G. Support for HIL Testing

1) *Description*: HIL testing is a vital testing methodology that integrates physical hardware with a simulated environment to test and validate systems. In the context of ADS development,

HIL testing enables meticulous checks of the autonomous vehicle stack's readiness for real-world deployment by interfacing sensors and vehicle controllers with the simulator. This approach boosts development efficiency and ensures safety when implementing autonomous driving technologies on actual vehicles.

Regrettably, as is shown in Table I, the investigation of HIL testing support within open-source simulators shows a dismal status. Most simulators lack built-in interfaces to receive input from real-time sensors or send output to vehicle controllers.

2) *Improvement Suggestions*: Developing an interface to facilitate HIL support presents a formidable challenge for developers of open-source simulators. This challenge arises partly due to the proprietary nature of vehicle control protocols, which are closely guarded by manufacturers for commercial and safety reasons. Moreover, the rapid evolution of sensor technologies necessitates a considerable cost to maintain an interface with high compatibility across a diverse range of sensor inputs. Although some attempts have been made to establish third-party HIL co-simulation for existing open-source simulators [153], [154], alongside the ongoing development of simulation platforms equipped with HIL support [155], [156], it is evident that this domain remains in its nascent stages. The progress of this field still awaits enhanced collaboration between open-source simulator developers and hardware providers in the future.

## VI. CONCLUSION

This paper presents a comprehensive overview of the history of simulators for autonomous driving over the past three decades. According to the increment pattern of simulators, the timeline is divided into three distinct phases: the incipient period (1990s-2000 s), the dormant period (2000s-2015), and the outbreak period (2015-present). A tendency toward open-source accessibility and comprehensive functionality is revealed in the evolution process.

After investigating the influential commercial and open-source simulators, they are classified into four task-specific classes: traffic flow simulator, sensory data simulator, driving policy simulator, and vehicle dynamics simulator. Additionally, simulators that enclose multiple functionalities are grouped as the fifth type, comprehensive simulators.



This paper systematically explores simulators and identifies critical issues focusing on open-source variants. Key concerns include simulation fidelity and efficiency. Fidelity issues extend to synthesizing sensory data, traffic scenarios, and vehicle dynamics. In generating traffic scenarios, efficiency challenges arise in constructing detailed maps with rich traffic information with minimal human labor, intricately linked to data format inconsistency. Simulation efficiency is a global concern while supporting HIL testing is a difficult challenge awaiting future opportunities and resources to address effectively. The potential hazards posed by these issues to the validity of ADS models developed in simulators are justified, serving as a reminder to simulator users and developers. The paper also reviews potential mitigation methods for these impacts.

This paper is a comprehensive guide for selecting an appropriate simulator for ADS research tasks. Simultaneously, it calls to address the pressing limitations of actively maintained open-source simulators. Simulation technology advancements are believed to accelerate the implementation and deployment of autonomous driving technology.

## REFERENCES

- [1] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," *Transp. Res. Part A: Policy Pract.*, vol. 94, pp. 182–193, 2016.
- [2] S. Feng et al., "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023.
- [3] The Waymo Team, "Simulation city: Introducing waymo's most advanced simulation system yet for autonomous driving," 2021. Accessed: Mar. 12, 2024. [Online]. Available: <https://waymo.com/blog/2021/06/SimulationCity.html>
- [4] Pony.ai, "Technology," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://pony.ai/tech?lang=en>
- [5] DiDi, "Didi's autonomous driving," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://www.didiglobal.com/science/intelligent-driving>
- [6] A. Sinha, S. Chand, V. Vu, H. Chen, and V. Dixit, "Crash and disengagement data of autonomous vehicles on public roads in California," *Sci. Data*, vol. 8, no. 1, 2021, Art. no. 298.
- [7] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.
- [8] H. Caesar et al., "NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles," 2021, *arXiv:2106.11810*.
- [9] C. Gulino et al., "Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 36.
- [10] J. Zhou, Y. Zhang, S. Guo, and Y. Guo, "A survey on autonomous driving system simulators," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops*, 2022, pp. 301–306.
- [11] M. Holen, K. Knausgård, and M. Goodwin, "An evaluation of autonomous car simulators and their applicability for supervised and reinforcement learning," in *Proc. Int. Conf. Intell. Technol. Appl.*, 2021, pp. 367–379.
- [12] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," in *Proc. IEEE 4th Int. Conf. Connected Auton. Driving*, 2021, pp. 62–70.
- [13] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.
- [14] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
- [15] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 9961–9980, Aug. 2022.
- [16] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robot.*, vol. 37, no. 3, pp. 362–386, 2020.
- [17] A. Haydari and Y. Yılmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 11–32, Jan. 2022.
- [18] Y. Cui et al., "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 722–739, Feb. 2022.
- [19] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.
- [20] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1364–1384, Apr. 2022.
- [21] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?," 2016, *arXiv:1610.01983*.
- [22] C. Katrakazas, M. Qudus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. Part C: Emerg. Technol.*, vol. 60, pp. 416–442, 2015.
- [23] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Ann. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 187–210, 2018.
- [24] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826–1848, May 2020.
- [25] U. M. Gidado, H. Chiroma, N. Aljojo, S. Abubakar, S. I. Popoola, and M. A. Al-Garadi, "A survey on deep learning for steering angle prediction in autonomous vehicles," *IEEE Access*, vol. 8, pp. 163797–163817, 2020.
- [26] Y. Kang, H. Yin, and C. Berger, "Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments," *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 171–185, Jun. 2019.
- [27] L. Wei, Z. Li, J. Gong, C. Gong, and J. Li, "Autonomous driving strategies at intersections: Scenarios, state-of-the-art, and future outlooks," in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 44–51.
- [28] X. Hu, S. Li, T. Huang, B. Tang, and L. Chen, "Sim2real and digital twins in autonomous driving: A survey," 2023, *arXiv:2305.01263*.
- [29] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation—A methodological perspective," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 6971–6988, Jul. 2023.
- [30] Q. Zhang et al., "TrajGEN: Generating realistic and diverse trajectories with reactive and feasible agent behaviors for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24474–24487, Dec. 2022.
- [31] C. Siebke, M. Mai, and G. Prokop, "What do traffic simulations have to provide for virtual road safety assessment? human error modeling in traffic simulations," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1419–1436, Feb. 2023.
- [32] PTV Group, "PTV vissim: Multimodal traffic simulation software," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://www.myptv.com/en/mobility-software/ptv-vissim>
- [33] M. Smith, G. Duncan, and S. Druiett, "PARAMICS: Microscopic traffic simulation for congestion management," in *Proc. IEE Colloquium Dynamic Contr. Strategic Inter-Urban Road Networks*, 1995.
- [34] K. G. Lim, C. H. Lee, R. K. Y. Chin, K. B. Yeo, and K. T. K. Teo, "SUMO enhancement for vehicular ad hoc network (VANET) simulation," in *Proc. IEEE 2nd Int. Conf. Autom. Control Intell. Syst.*, 2017, pp. 86–91.
- [35] Mechanical Simulation, "Carsim," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://www.carsim.com>
- [36] IPG Automotive, "Solutions for virtual test driving," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://ipg-automotive.com/en>
- [37] rFpro, "The world's most accurate simulation environment," 2023. Accessed: Mar. 12, 2024. [Online]. Available: <https://rfpro.com>
- [38] Siemens, "Simcenter prescan software," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://plm.sw.siemens.com/en-US/simcenter/autonomous-vehicle-solutions/prescan>
- [39] VI-grade, "Driving simulators: An invaluable set of tools to successfully accelerate your development process," 2023. Accessed: Mar. 12, 2024. [Online]. Available: <https://www.vi-grade.com/en/solutions/driving-simulators>
- [40] 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," 2017, *arXiv:1712.01397*.
- [41] D. Wang, J. Wen, Y. Wang, X. Huang, and F. Pei, "End-to-end self-driving using deep neural networks with multi-auxiliary tasks," *Automot. Innov.*, vol. 2, pp. 127–136, 2019.

- [42] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Proc. Field Serv. Robot.: Results 11th Int. Conf.*, 2018, pp. 621–635.
- [43] G. Rong et al., "LGSVL simulator: A high fidelity simulator for autonomous driving," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–6.
- [44] E. Leurent, "An environment for autonomous driving decision-making," *GitHub Repository*, 2018. Accessed: Mar. 12, 2024. [Online]. Available: <https://github.com/eleurent/highway-env>
- [45] M. Althoff, M. Koschi, and S. Manzing, "CommonRoad: Composible benchmarks for motion planning on roads," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 719–726.
- [46] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2575–2582. [Online]. Available: <https://elib.dlr.de/124092/>
- [47] J. Barceló and J. Casas, "Dynamic network simulation with AIMSUN," in *Simulation Approaches in Transportation Analysis: Recent Advances and Challenges*. Berlin, Germany: Springer, 2005, pp. 57–98.
- [48] J. Auld, M. Hope, H. Ley, V. Sokolov, B. Xu, and K. Zhang, "Polaris: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations," *Transp. Res. Part C: Emerg. Technol.*, vol. 64, pp. 101–116, 2016.
- [49] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," 2017, *arXiv:1710.05465*.
- [50] H. Zhang et al., "Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario," in *Proc. World Wide Web Conf.*, 2019, pp. 3620–3624.
- [51] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications," *Int. J. Comput. Vis.*, vol. 126, pp. 902–919, 2018.
- [52] Parallel Domain, "Parallel domain," 2023. Accessed: Mar. 12, 2024. [Online]. Available: <https://paralleldomain.com/>
- [53] Z. Yang et al., "UniSim: A neural closed-loop sensor simulator," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 1389–1399.
- [54] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," 2020. Accessed: Mar. 12, 2024. [Online]. Available: <https://sourceforge.net/projects/torcs/>
- [55] F. Kehrle, J. V. Frasc, C. Kirches, and S. Sager, "Optimal control of formula 1 race cars in a vdrift based virtual environment," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 11907–11912, 2011.
- [56] C. Campbell, "Box2D C++ tutorials - top-down car physics," 2014. Accessed: Mar. 12, 2024. [Online]. Available: <http://www.iforce2d.net/b2dtut/top-down-car>
- [57] Udacity, "Udacity self-driving platform," 2017. Accessed: Mar. 12, 2024. [Online]. Available: <https://github.com/udacity/self-driving-car-sim>
- [58] P. Palanisamy, "Multi-agent connected autonomous driving using deep reinforcement learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2020, pp. 1–7.
- [59] J. Bernhard, K. Esterle, P. Hart, and T. Kessler, "BARK: Open behavior benchmarking in multi-agent environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 6201–6208.
- [60] P. Kothari, C. Perone, L. Bergamini, A. Alahi, and P. Ondruska, "DriverGym: Democratising reinforcement learning for autonomous driving," in *Proc. NeurIPS Workshop Mach. Learn. Autonom. Driving*, 2021.
- [61] M. Zhou et al., "Smarts: An open-source scalable multi-agent rl training school for autonomous driving," in *Proc. Conf. Robot Learn.*, 2021, pp. 264–285.
- [62] P. Cai, Y. Lee, Y. Luo, and D. Hsu, "SUMMIT: A simulator for urban driving in massive mixed traffic," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4023–4029.
- [63] DI-drive Contributors, "DI-drive: OpenDILab decision intelligence platform for autonomous driving simulation," 2021. Accessed: Mar. 12, 2024. [Online]. Available: <https://github.com/opendilab/DI-drive>
- [64] J. Herman et al., "Learn-to-race: A multimodal control environment for autonomous racing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9793–9802.
- [65] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3461–3475, Mar. 2023.
- [66] Q. Sun, X. Huang, B. C. Williams, and H. Zhao, "Intersim: Interactive traffic simulation via explicit relation modeling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 11416–11423.
- [67] E. Vinitsky, N. Lichtlé, X. Yang, B. Amos, and J. Foerster, "Nocturne: A scalable driving benchmark for bringing multi-agent learning one step closer to the real world," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 3962–3974.
- [68] D. Xu, Y. Chen, B. Ivanovic, and M. Pavone, "BITS: Bi-level imitation for traffic simulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 2929–2936.
- [69] O. Michel, "Cyberbotics Ltd. webots: Professional mobile robot simulation," *Int. J. Adv. Robot. Syst.*, vol. 1, no. 1, p. 5, 2004.. [Online]. Available: <https://journals.sagepub.com/doi/full/10.5772/5618>
- [70] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, pp. 2149–2154.
- [71] The MathWorks, "Vehicle dynamics blockset: Model and simulate vehicle dynamics in a virtual 3D environment," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://www.mathworks.com/products/vehicle-dynamics.html>
- [72] AVSimulation, "SCANeR 2023.2 release note," 2023. Accessed: Mar. 12, 2024. [Online]. Available: <https://www.avsimulation.com/scaner-2023-2-release-note>
- [73] Hexagon, "Virtual test drive: Complete tool-chain for driving simulation applications," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://hexagon.com/products/virtual-test-drive>
- [74] Deepdrive Team, "Deepdrive: A simulator that allows anyone with a pc to push the state-of-the-art in self-driving," 2019. Accessed: Mar. 12, 2024. [Online]. Available: <https://github.com/deepdrive/deepdrive>
- [75] Nvidia, "Nvidia drive end-to-end platform for software-defined vehicles," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://www.nvidia.com/en-us/self-driving-cars/>
- [76] A. Amini et al., "Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 2419–2426.
- [77] S. Kato et al., "Autoware on board: Enabling autonomous vehicles with embedded systems," in *Proc. ACM/IEEE 9th Int. Conf. Cyber- Phys. Syst.*, 2018, pp. 287–296.
- [78] Baidu, "Baidu apollo," 2023. Accessed: Mar. 12, 2024. [Online]. Available: <https://github.com/ApolloAuto/apollo>
- [79] C. Olaverri-Monreal, J. Errea-Moreno, A. Díaz-Álvarez, C. Biurrun-Quel, L. Serrano-Arriazu, and M. Kuba, "Connection of the SUMO microscopic traffic simulator and the unity 3D game engine to evaluate v2x communication-based systems," *Sensors*, vol. 18, no. 12, 2018, Art. no. 4399.
- [80] A. Von Bernuth, G. Volk, and O. Bringmann, "Simulating photo-realistic snow and fog on existing images for enhanced CNN training and evaluation," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 41–46.
- [81] M. Tremblay, S. S. Halder, R. De Charette, and J.-F. Lalonde, "Rain rendering for evaluating and improving robustness to bad weather," *Int. J. Comput. Vis.*, vol. 129, pp. 341–360, 2021.
- [82] M. Hahner et al., "LiDAR snowfall simulation for robust 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16364–16374.
- [83] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct.–Dec. 2008.
- [84] M. Dupuis, M. Strobl, and H. Grezlikowski, "Opendrive 2010 and beyond—status and future of the de facto standard for the description of road networks," in *Proc. Driving Simul. Conf. Eur.*, 2010, pp. 231–242.
- [85] R. Krajewski, J. Bock, L. Kloecker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2118–2125.
- [86] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1929–1934.
- [87] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The round dataset: A drone dataset of road user trajectories at roundabouts in Germany," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–6.
- [88] J. Bock, L. Vater, R. Krajewski, and T. Moers, "Highly accurate scenario and reference data for automated driving," *ATZ Worldwide*, vol. 123, no. 5, pp. 50–55, 2021.
- [89] Y. Xu et al., "SIND: A drone dataset at signalized intersection in China," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst.*, 2022, pp. 2471–2478.

- [90] W. Zhan et al., "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," 2019, *arXiv:1910.03088*.
- [91] S. Ettinger et al., "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9710–9719.
- [92] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos, "An overview of autonomous vehicles sensors and their vulnerability to weather conditions," *Sensors*, vol. 21, no. 16, 2021, Art. no. 5397.
- [93] C. Goodin, D. Carruth, M. Doude, and C. Hudson, "Predicting the influence of Rain on LiDAR in Adas," *Electronics*, vol. 8, no. 1, p. 89, 2019. [Online]. Available: <https://www.mdpi.com/2079-9292/8/1/89>
- [94] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 103–111, Jun. 2019.
- [95] W. Li, Q. Zhang, Z. Huang, X. Tian, and D. Tao, "Toward real-world single image deraining: A new benchmark and beyond," 2022, *arXiv:2206.05514*.
- [96] F. Yu et al., "Bdd100 k: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2636–2645.
- [97] D. Dunkerley, "Rain event properties in nature and in rainfall simulation experiments: A comparative review with recommendations for increasingly systematic study and reporting," *Hydrological Processes: An Int. J.*, vol. 22, no. 22, pp. 4415–4435, 2008.
- [98] S. Starik and M. Werman, "Simulation of rain in videos," in *Proc. Texture Workshop*, 2003, vol. 2, pp. 406–409.
- [99] C. Wang, Z. Wang, T. Xia, and Q. Peng, "Real-time snowing simulation," *Vis. Comput.*, vol. 22, pp. 315–323, 2006.
- [100] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: Automated testing of deep-neural-network-driven autonomous cars," in *Proc. 40th Int. Conf. Softw. Eng.*, 2018, pp. 303–314.
- [101] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *Int. J. Comput. Vis.*, vol. 126, pp. 973–992, 2018.
- [102] A. Kar et al., "Meta-Sim: Learning to generate synthetic datasets," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4551–4560.
- [103] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, "K-Radar: 4D radar object detection for autonomous driving in various weather conditions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 3819–3829.
- [104] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems," in *Proc. ACM/IEEE 33rd Int. Conf. Automated Softw. Eng.*, 2018, pp. 132–142.
- [105] V. Muşat, I. Fursa, P. Newman, F. Cuzzolin, and A. Bradley, "Multi-weather city: Adverse weather stacking for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2906–2915.
- [106] Z. Yang et al., "SurfelGAN: Synthesizing realistic sensor data for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11118–11127.
- [107] Y. Li, Z.-H. Lin, D. Forsyth, J.-B. Huang, and S. Wang, "Climatenerf: Extreme weather synthesis in neural radiance field," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 3227–3238.
- [108] Z. Ghodsi et al., "Generating and characterizing scenarios for safety testing of autonomous vehicles," in *Proc. IEEE Intell. Veh. Symp.*, 2021, pp. 157–164.
- [109] J. A. Michon, "Explanatory pitfalls and rule-based driver models," *Accident Anal. Prevention*, vol. 21, no. 4, pp. 341–353, 1989.
- [110] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transp. Res. Rec.*, vol. 1999, no. 1, pp. 86–94, 2007.
- [111] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," *Res. Rep. 98-08 German Center Air Space Navigation*, 1998.
- [112] J. Erdmann, "SUMO's lane-changing model," in *Proc. Model. Mobility With Open Data: 2nd SUMO Conf.*, 2015, pp. 105–123.
- [113] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 3, pp. 314–325, Sep. 2005.
- [114] R. Bhattacharyya, S. Jung, L. A. Kruse, R. Senanayake, and M. J. Kochenderfer, "A hybrid rule-based and data-driven approach to driver modeling through particle filtering," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13055–13068, Aug. 2022.
- [115] D. Lee, Y. Gu, J. Hoang, and M. Marchetti-Bowick, "Joint interaction and trajectory prediction for autonomous driving using graph neural networks," 2019, *arXiv:1912.07882*.
- [116] S. Kumar, Y. Gu, J. Hoang, G. C. Haynes, and M. Marchetti-Bowick, "Interaction-based trajectory prediction over a hybrid traffic graph," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 5530–5535.
- [117] J. Gao et al., "VectorNet: Encoding HD maps and agent dynamics from vectorized representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11525–11533.
- [118] Z.-H. Yin, L. Sun, L. Sun, M. Tomizuka, and W. Zhan, "Diverse critical interaction generation for planning and planner evaluation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 7036–7043.
- [119] S. Shiroshita et al., "Behaviorally diverse traffic simulation via reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 2103–2110.
- [120] B. Chen, X. Chen, Q. Wu, and L. Li, "Adversarial evaluation of autonomous vehicles in lane-change scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10333–10342, Aug. 2022.
- [121] N. A. Spielberg, M. Templer, J. Subosits, and J. C. Gerdes, "Learning policies for automated racing using vehicle model gradients," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 130–142, 2023.
- [122] R. Smith et al., "Open dynamics engine," 2004. Accessed: Mar. 12, 2024. [Online]. Available: <http://www.ode.org/>
- [123] I. Parberry, *Introduction to Game Physics With Box2D*. Boca Raton, FL, USA: CRC Press, 2017.
- [124] E. Coumans and Y. Bai, "Pybullet quickstart guide," 2021. Accessed: Mar. 12, 2024. [Online]. Available: [https://raw.githubusercontent.com/bulletphysics/bullet3/master/docs/pybullet\\_quickstartguide.pdf](https://raw.githubusercontent.com/bulletphysics/bullet3/master/docs/pybullet_quickstartguide.pdf)
- [125] A. Sanders, *An Introduction to Unreal Engine 4*. Boca Raton, FL, USA: CRC, 2016.
- [126] M. Goslin and M. R. Mine, "The panda 3D graphics engine," *Computer*, vol. 37, no. 10, pp. 112–114, 2004.
- [127] J. Kong, M. Pfeiffer, G. Schilbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proc. IEEE Intell. Veh. Symp.*, 2015, pp. 1094–1099.
- [128] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—simulation of urban mobility: An overview," in *Proc. SIMUL 3rd Int. Conf. Adv. Syst. Simul.*, 2011, pp. 23–28.
- [129] F. Poggendorf et al., "Lanelet2: A high-definition map framework for the future of automated driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 1672–1679.
- [130] Association for Standardization of Automation and Measuring Systems, "Asam openscenario: User guide," 2020. Accessed: Mar. 12, 2024. [Online]. Available: [https://releases.asam.net/OpenSCENARIO/1.0.0/ASAM\\_OpenSCENARIO\\_BS-1-2\\_User-Guide\\_V1-0-0.html](https://releases.asam.net/OpenSCENARIO/1.0.0/ASAM_OpenSCENARIO_BS-1-2_User-Guide_V1-0-0.html)
- [131] 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 *Proc. 40th ACM SIGPLAN Conf. Program. Lang. Des. Implementation*, 2019, pp. 63–78.
- [132] D. Krajzewicz, G. Hertkorn, J. Ringel, and P. Wagner, "Preparation of digital maps for traffic simulation; Part 1: Approach and algorithms," in *Proc. 3rd Ind. Simul. Conf.*, 2005, pp. 285–290.
- [133] S. Maierhofer, M. Klischat, and M. Althoff, "Commonroad scenario designer: An open-source toolbox for map conversion and scenario creation for autonomous vehicles," in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 3176–3182.
- [134] MathWorks, "Get started with roadrunner," 2024. Accessed: Mar. 12, 2024. [Online]. Available: <https://www.mathworks.com/help/roadrunner/get-started-with-roadrunner.html>
- [135] J. Huang, J. Stoter, R. Peters, and L. Nan, "City3d: Large-scale building reconstruction from airborne LiDAR point clouds," *Remote Sens.*, vol. 14, no. 9, 2022, Art. no. 2254.
- [136] L. Nan and P. Wonka, "PolyFit: Polygonal surface reconstruction from point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2353–2361.
- [137] J. Pan, X. Han, W. Chen, J. Tang, and K. Jia, "Deep mesh reconstruction from single RGB images via topology modification networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9964–9973.
- [138] G.-H. Lee, P.-H. Choi, J.-H. Nam, H.-S. Han, S.-H. Lee, and S.-C. Kwon, "A study on the performance comparison of 3D file formats on the web," *Int. J. Adv. Smart Convergence*, vol. 8, no. 1, pp. 65–74, 2019.
- [139] C. Früh and A. Zakhor, "An automated method for large-scale, ground-based city model acquisition," *Int. J. Comput. Vis.*, vol. 60, pp. 5–24, 2004.
- [140] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," in *Proc. ACM siggraph Papers*, 2006, pp. 835–846.
- [141] S. Agarwal et al., "Building rome in a day," *Commun. ACM*, vol. 54, no. 10, pp. 105–112, 2011.



- [142] F. Williams, M. Trager, J. Bruna, and D. Zorin, "Neural splines: Fitting 3D surfaces with infinitely-wide neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9949–9958.
- [143] A. Boulch and R. Marlet, "Poco: Point convolution for surface reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6302–6314.
- [144] M. Tancik et al., "Block-NeRF: Scalable large scene neural view synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8248–8258.
- [145] J. Huang, Z. Gojcic, M. Atzmon, O. Litany, S. Fidler, and F. Williams, "Neural kernel surface reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4369–4379.
- [146] Z. Li et al., "Neuralangelo: High-fidelity neural surface reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8456–8465.
- [147] A. Mondal, P. Tigas, and Y. Gal, "Real2sim: Automatic generation of open street map towns for autonomous driving benchmarks," in *Proc. Mach. Learn. Auton. Driving Workshop 34th Conf. Neural Inf. Process. Syst.*, 2020.
- [148] S. Tan et al., "SceneGen: Learning to generate realistic traffic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 892–901.
- [149] Y. Tian, A. Carballo, R. Li, and K. Takeda, "Real-to-synthetic: Generating simulator friendly traffic scenes from graph representation," in *Proc. IEEE Intell. Veh. Symp.*, 2022, pp. 1615–1622.
- [150] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2174–2182.
- [151] B. Varga, D. Doba, and T. Tettamanti, "Optimizing vehicle dynamics co-simulation performance by introducing mesoscopic traffic simulation," *Simul. Modelling Pract. Theory*, vol. 125, 2023, Art. no. 102739.
- [152] A. Rousset, B. Herrmann, C. Lang, and L. Philippe, "A survey on parallel and distributed multi-agent systems for high performance computing simulations," *Comput. Sci. Rev.*, vol. 22, pp. 27–46, 2016.
- [153] Y. Chen, S. Chen, T. Zhang, S. Zhang, and N. Zheng, "Autonomous vehicle testing and validation platform: Integrated simulation system with hardware in the loop," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 949–956.
- [154] C. Brogle, C. Zhang, K. L. Lim, and T. Bräunl, "Hardware-in-the-loop autonomous driving simulation without real-time constraints," *IEEE Trans. Intell. Veh.*, vol. 4, no. 3, pp. 375–384, Sep. 2019.
- [155] Ş. Y. Gelbal, S. Tamilarasan, M. R. Cantas, L. Güvenç, and B. Aksun-Güvenç, "A connected and autonomous vehicle hardware-in-the-loop simulator for developing automated driving algorithms," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2017, pp. 3397–3402.
- [156] J. E. Heikkinen, S. Gafurov, S. Kopylov, T. Minav, S. Grebennikov, and A. Kurbanov, "Hardware-in-the-loop platform for testing autonomous vehicle control algorithms," in *Proc. IEEE 12th Int. Conf. Develop. Ecosystems Eng.*, 2019, pp. 906–911.



**Yueyuan Li** received the bachelor's degree in electrical and computer engineering from the University of Michigan-Shanghai Jiao Tong University Joint Institute, Shanghai, China, in 2020. She is currently working toward the Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai. Her research interest include the security of the autonomous driving system and driving decision-making, driving decision-making models, driving simulation, and virtual-to-real model transferring.



**Wei Yuan** received the master's and Ph.D. degrees in automation from Shanghai Jiao Tong University, Shanghai, China, in 2017 and 2021, respectively. He is currently a Postdoctoral Researcher with Shanghai Jiao Tong University. His research interests include autonomous driving systems, computer vision, deep learning, vehicle control, end-to-end learning-based vehicle control, and decision-making.



vehicles, model-based reinforcement learning, and meta-reinforcement learning for autonomous vehicle decision-making.



**Weihao Yan** received the bachelor's degree in automation in 2020 from Shanghai Jiao Tong University, Shanghai, China, where he is currently working toward the Ph.D. degree in control science and engineering. His research interests include autonomous driving systems, computer vision, domain adaptation, virtual-to-real transfer learning, scene segmentation, and foundation models.



**Qiyuan Shen** received the bachelor's degree in automation from Northeastern University, Shenyang, China, in 2022. He is currently working toward the Ph.D. degree in control science and engineering with Shanghai Jiao Tong University, Shanghai, China. His research interests include the localization of the autonomous driving system and SLAM, multimodal mapping, cross-modal localization, geometrical calibration for vision, and LiDAR.



**Chunxiang Wang** (Member, IEEE) received the Ph.D. degree in mechanical engineering from the Harbin Institute of Technology, Harbin, China, in 1999. She is currently an Associate Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. She is working on intelligent vehicles for more than ten years and has participated in several related research projects, such as European CyberC3 Project, and ITER Transfer Cask Project. Her research interests include autonomous driving, assistant driving, and mobile robots.



**Ming Yang** (Member, IEEE) received the master's and Ph.D. degrees from Tsinghua University, Beijing, China, in 1999 and 2003, respectively. He currently holds the position of a Distinguished Professor with Shanghai Jiao Tong University, Shanghai, China, and the Director of the Innovation Center of Intelligent Connected Vehicles. He has engaged in the research of intelligent vehicles for more than 25 years.