The OpenCDA Open-Source Ecosystem for Cooperative Driving Automation Research

Runsheng Xu, Hao Xiang, Xu Han, Xin Xia, Zonglin Meng, Chia-Ju Chen, Camila Correa-Jullian, and Jiaqi Ma, *Member, IEEE*

***Abstract*—Advances in Single-vehicle intelligence of automated driving has encountered great challenges because of limited capabilities in perception and interaction with complex traffic environments. Cooperative Driving Automation (CDA) has been considered a pivotal solution to next-generation automated driving and smart transportation. Though CDA has attracted much attention from both academia and industry, exploration of its potential is still in its infancy. In industry, companies tend to build their in-house data collection pipeline and research tools to tailor their needs and protect intellectual properties. Reinventing the wheels, however, wastes resources and limits the generalizability of the developed approaches since no standardized benchmarks exist. On the other hand, in academia, due to the absence of real-world traffic data and computation resources, researchers often investigate CDA topics in simplified and mostly simulated environments, restricting the possibility of scaling the research outputs to real-world scenarios. Therefore, there is an urgent need to establish an open-source ecosystem (OSE) to address the demands of different communities for CDA research, particularly in the early exploratory research stages, and provide the bridge to ensure an integrated development and testing pipeline that diverse communities can share. In this paper, we introduce the OpenCDA research ecosystem, a unified OSE integrated with a model zoo, a suite of driving simulators at various resolutions, large-scale real-world and simulated datasets, complete development toolkits for benchmark training/testing, and a scenario database/generator. We also demonstrate the effectiveness of OpenCDA OSE through example use cases, including cooperative 3D LiDAR detection, cooperative merge, cooperative camera-based map prediction, and adversarial scenario generation.**

***Index Terms*—Cooperative driving automation, automated driving, intelligent transportation system, digital twin.**

I. Introduction

The advent of automated driving technology has garnered substantial attention and investment in recent years, with many experts positing that it has the potential to revolutionize the transportation industry [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Automated vehicles possess the capability to significantly enhance road safety, alleviate traffic congestion, and increase mobility for individuals who are unable to operate a vehicle, such as the elderly or disabled. Furthermore, the implementation of automated vehicles has the potential to reduce the number of cars on the road, thus leading to decreased pollution and energy consumption.

Despite the significant progress that has been made in the field of automated driving, single-vehicle perception systems are still prone to occlusion and are limited in their ability to detect distant objects [12], [13], [14], [15], [16], [17], [18]. Furthermore, the integration of automated vehicles into a mixed-autonomy transportation system, where they must interact with human-driven vehicles, pedestrians, complicated traffic rules, and various types of infrastructure, poses significant difficulties in making safe decisions [18], [19], [20]. These challenges remain a major obstacle to the large-scale deployment of automated driving technology in real-world scenarios.

Cooperative Driving Automation (CDA) has recently gained significant attention as a solution to the challenges faced by individual automated vehicles, such as occlusion and limited perception range [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. According to SAEJ3216 [31], CDA refers to the use of machine-to-machine communication to enable cooperation among multiple entities, such as vehicles, pedestrians, and infrastructure components, equipped with capable communication technologies. With the support of CDA, automated vehicles can see through occlusion, extend their perception range [32], [33], and make more intelligent decisions [1], [34]. One of the early programs to research CDA is the CDA/CARMA program hosted by the U.S. Department of Transportation, which initiated SAEJ3216 [35].

CDA has the potential to bring significant benefits to the field of automated driving. However, CDA is still in its early stages of development. In academia, researchers in university labs often have limited access to real-world traffic data and computational resources, so they typically study CDA in simplified simulation environments. This makes it difficult to apply their research findings to real-world scenarios. Additionally, most research studies utilize simulation capabilities that are not comparable between studies, hindering the ability to compare and evaluate the performance of different approaches. On the other hand, while companies in the industry typically have access to rich resources, they tend to primarily focus on current single-vehicle

Manuscript received 11 December 2022; revised 17 January 2023 and 30 January 2023; accepted 30 January 2023. Date of publication 15 February 2023; date of current version 19 May 2023. *(Corresponding author: Jiaqi Ma.)*

Runsheng Xu, Hao Xiang, Xu Han, Xin Xia, Zonglin Meng, Chia-Ju Chen, and Jiaqi Ma are with the Department of Civil and Environmental Engineering, University of California, Los Angeles, CA 90024 USA (e-mail: rxx3386@ucla.edu; haxiang@ucla.edu; hanxu417@g.ucla.edu; x35xia@ucla.edu; meng925@g.ucla.edu; ju40268@ucla.edu; majiaqimark@gmail.com).

Camila Correa-Jullian is with the Department of Mechanical Engineering, University of California, Los Angeles, CA 90024 USA (e-mail: ccorreaj@ucla.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2023.3244948.

Digital Object Identifier 10.1109/TIV.2023.3244948

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

technology rather than investing in CDA research. Furthermore, those that do invest in CDA often keep their data and pipelines confidential. To promote the healthy and rapid development of CDA, the creation of an open-source ecosystem (OSE) is essential. Such an OSE should be able to: 1) meet the requirements of various stakeholders; 2) facilitate the integration of development and testing pipelines across different communities; 3) furnish a comprehensive toolset that can be utilized by various communities for the development of functionalities and seamless integration with other pertinent modules throughout the automated vehicle-transportation interaction process.

To this end, we introduce the OpenCDA Ecosystem, which is grown from our previous simulation tool OpenCDA [1]. The OpenCDA ecosystem is a comprehensive OSE that includes a collection of pre-trained models, a range of simulators for driving and traffic at different levels of detail, benchmark datasets for training and testing, and a scenario database and generator for CDA research. The OpenCDA ecosystem comprises four main components: *toolkits*, *scenario databases*, *data*, *and communities:*

- *Toolkits:* The OpenCDA ecosystem offers a comprehensive set of toolchains to assist communities in prototyping and developing their CDA systems and algorithms. It includes a simulation tool, a deep learning framework, and libraries of models that include state-of-the-art algorithms for various automated driving tasks.
- *Data:* The OpenCDA ecosystem contains both simulated and real-world large-scale datasets to support offline training for cooperative perception algorithms.
- *Scenarios:* OpenCDA offers a scenario database, generation, and man- management tools, including various rule-based standard scenarios for testing different cooperative driving applications as benchmarks.
- *Community:* OpenCDA has a comprehensive and well-structured plan to establish and nurture a thriving community of researchers and engineers with diverse backgrounds and skill sets to facilitate the development of CDA.

The rest of the paper is organized as follows: we will first introduce the literature in the CDA field in Section II, then we will describe the details of each component of our OpenCDA ecosystem in Section III. Afterward, we will conduct several case studies to demonstrate how the OpenCDA ecosystem facilitates different research needs. Eventually, we will conclude the work and depict the future orientations.

II. Related Work

A. Simulators

Conducting field experiments in the field of automated driving can present significant challenges, particularly in terms of cost and safety. To mitigate these issues, researchers and engineers often first prototype their systems in simulated environments before implementing them in the real world. Among the available simulators, CARLA [36] has been a popular choice, offering a free, open-source platform that aims to accelerate the development of new automated driving technologies. Furthermore, other simulators like SUMO [37] and Flow [38] are also widely used, but they are typically focused on traffic-level simulation and do not support regular automated driving components and cooperative driving automation. Our previous work, OpenCDA [1], aims to fill this gap by providing an integrated framework that supports full-stack CDA software development in simulation and can be used to simulate various scenarios with the support of various communities.

B. Cooperative Driving Automation

Cooperative driving automation allows vehicles to communicate with each other and with the infrastructure around them to improve safety and increase efficiency on the road. The technology uses a combination of sensors, including radar, lidar, and cameras, as well as wireless communication systems, such as dedicated short-range communication (DSRC) and cellular networks, to enable vehicles to share information about their intents and observation. Several CDA applications have emerged in the past decade and hold great premise in impacting the transportation systems, such as Cooperative Adaptive Cruise Control (CACC), platooning, cooperative merge, cooperative perception, eco-drive at signalized intersections. The applications also evolve quickly as supporting technologies, such as computer vision and vehicular communication, advance.

For example, CACC, i.e., single-lane decentralized ad-hoc operations of multiple vehicles that closely follow each other, has been widely investigated [39], [40], [41]. Although CACC can improve the safety and comfort of driving, it is only applied longitudinally, and each vehicle can only communicate with the front vehicle. In contrast, another important CDA application, i.e., platooning, enabled by higher levels of vehicle automation, can handle multi-lane scenarios and allow the members to cooperate with both the front vehicle and the leader [42]. To implement such a system, each platoon member communicates simultaneously via an established principle that shares intention and operating status that serve as the input of longitudinal (i.e., speed regularization) and lateral control (i.e., lane change). The platooning algorithms can either be distributed [43], [44] or centralized [45], [46].

Recently, another example application that has receive much attention is cooperative perception, which builds upon recent progress in computer vision and ADS perception. Cooperative perception studies how to efficiently fuse visual cues from neighboring vehicles or infrastructure [32], [47], [48], [49], [50], [51], [52], [53]. Based on the vision information sharing strategy, it can be divided into three categories: 1) early fusion, where raw data is shared and gathered to form a holistic view, 2) intermediate fusion, where intermediate neural features are extracted based on each agent's observation and then transmitted; and 3) late fusion, where detection outputs (e.g., 3D bounding box position, confidence score) are circulated. By utilizing OpenCDA [1] simulation tool, OPV2V [13] collects the first large-scale Vehicle-to-Vehicle (V2V) simulated perception dataset and proposes the first cooperative detection framework OpenCOOD [54]. Since then, an increasing number of related works that use OpenCOOD as the base have shown up and brought significant impacts to the CDA field, including

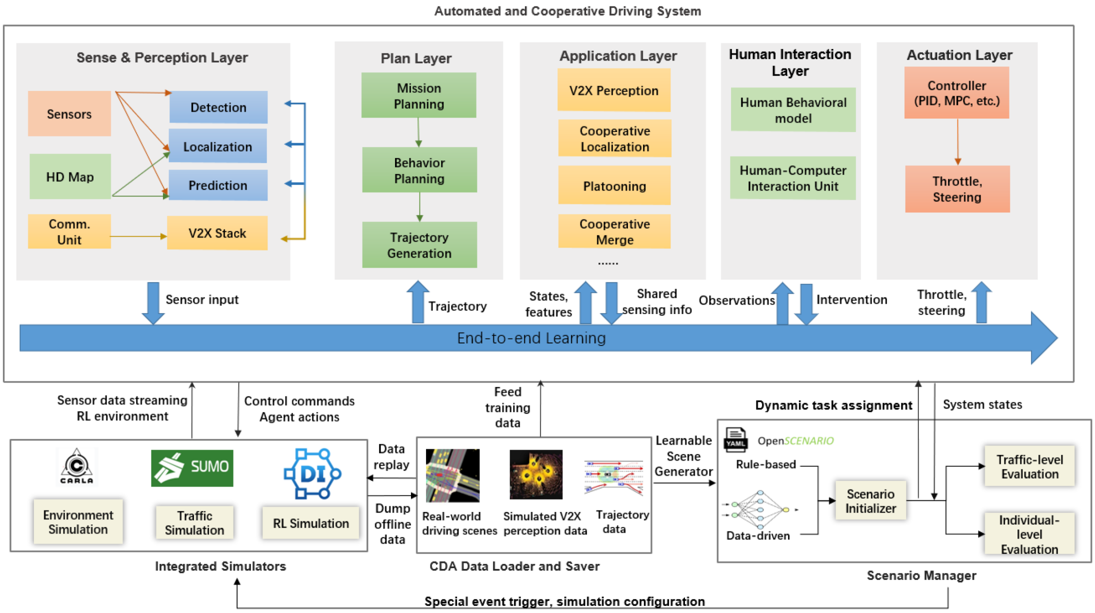


Fig. 1. Overview of the OpenCDA simulation tool architecture.

V2X-Vit [32], CoBEVT [55], Where2Comm [33], Map Container [56], CoAlign [57], and AdaFusion [58].

The Federal Highway Administration's (FHWA) CARMA Program [35] is a leading research initiative that utilizes emerging technologies in automation and cooperation to improve transportation systems management and operations. The program has developed software platforms for both vehicles and infrastructure, as well as tools for full-scale vehicle software simulation and testing. In partnership with the CARMA Program, OpenCDA - an open-source ecosystem - makes a unique contribution by providing early-stage development and testing opportunities for users to evaluate their customized algorithms in terms of specific tasks, such as object detection accuracy and overall pipeline assessment for traffic safety.

III. The OpenCDA Ecosystem

In this section, we will present the details of the major components of the OpenCDA ecosystem (as of the publication date of this paper, i.e., January 2023) and then explain how they interact with each other through the interfaces to form an ecosystem.

A. Development Toolkits

The development toolkits are a set of software routines and utilities that assists the communities in prototyping and developing their CDA systems and algorithms.The toolkits contain a simulation tool, a deep learning framework OpenCOOD [54], and a model zoo that contains state-of-the-art (SOTA) algorithms of different automated driving tasks.

1) OpenCDA Simulation Tool: OpenCDA[1] started as an open co-simulation-based simulation tool integrated with prototype cooperative driving automation pipelines as well as regular automated driving components (e.g., perception, localization, planning, control). This simulation tool is the core element of the whole ecosystem, as it enables CDA evaluation in a CARLA + SUMO co-simulation environment and provides a rich library of source codes of CDA research pipelines. In collaboration with U.S.DOT CDA Research and the FHWA CARMA Program [35], OpenCDA, as an open-source project, is designed and built to support early-stage fundamental research for CDA research and development.

As Fig. 1 depicts, the simulation tool has four major parts: simulator integration, cooperative driving system, scenario manager, and CDA data loader and saver:

Integrated simulation: OpenCDA integrate CARLA [36] for realistic environment rendering, SUMO [37] for realistic background traffic, and DI-Engine [59] to provide reinforcement learning environment. These simulators together will supply the sensor data streaming and necessary environment information to the cooperative driving system.

Automated and Cooperative Driving System: OpenCDA builds on the philosophy that CDA functionalities need to build on and also enhance automated driving technologies of single vehicle intelligence. Therefore, OpenCDA prototypes the entire regular automated driving systems software and adds modular CDA components (e.g., V2X communication, cooperative perception, platooning) to the driving automation system. For example, the percpetion backbones of 3D LiDAR or camera perception algorithms starts within single vehicle automation and cooperative perception builds on top of it by exploring different strategies of information sharing and fusion (e.g., intermediate or late fusion). Platooning modules still relies on signle vehicle plan and control modules but only interfaces with platooning strategic and tactical decision making modules to output desired trajectories and maneuvers that not only guarantee single vehicle safety but also satisfies platooning requirements.

[1] code link: https://github.com/ucla-mobility/OpenCDA

Specifically, OpenCDA purely utilizes Python to prototype the cooperative driving system, which consists of the sensing layer, plan layer, application layer, and actuation layer. After receiving the sensor information from simulators, the sensing layer will perform object detection, road topology prediction, and localization. Then the sensing and localization information will be delivered to the planning layer for behavior plan and trajectory generation. The outputs of the planning module will be fed into the actuation layer, where the control commands are finally generated and sent back to the simulator to update. When multiple AVs meet, the sensing layer's communication unit will start working and activating the application layer, where cooperative functionalities such as cooperative perception and platooning will replace the default perception and planning modules. Besides the classic modular pipeline, OpenCDA also supports end-to-end driving, i.e., the control commands are directly generated from the raw sensor measurements without any intermediate outputs.

The cooperative driving system component also includes a special layer called the human interaction layer, which is made up of two sub-units: a human behavioral model and a human-computer interaction unit. This layer receives inputs from all other layers of the cooperative driving system, as well as human behavior inputs in response to generated driving scenarios. The interaction mechanism unit then outputs commands to the plan and actuation layers. The human operator model is designed to operate in two modes: simulation-based and experimental-based. In simulation-based mode, model- and simulation-based Human Reliability Analysis (HRA) methods, such as the Information-Decision-Action Crew (IDAC) operator model [60], are used to simulate human performance. In the experimental-based mode, humans receive inputs from the automated vehicle system. The experimental setup adheres to human factors and ergonomic principles with a level of fidelity suitable for specific studies. The interaction model allows for human intervention in AV operation at two levels: remote commands (e.g., waypoints, remote stops) and local vehicle control (e.g., take-over requests, disengagement mechanisms). This flexibility allows for the simulation of different operational scenarios and the assessment of task-sharing strategies between the human operator and the AV system under various operating conditions. The setup can also be used to collect data on human factors and behavior performance influencing factors in time-constrained monitoring and intervention scenarios.

Scenario Manager: The scenario manager mainly has three roles: 1) It depicts the scenarios at the beginning of the simulation, including weather, traffic states, testing locations, etc. 2) It assigns the dynamic driving tasks to different AVs, including start positions, destinations, and the category of the cooperating maneuvers need to be operated. 3) It provides both traffic level and individual level evaluation after each run.

CDA Data Loader and Saver: In OpenCDA, we collect the simulated perception and traffic data in a similar approach as the real world. The connected AVs will drive simultaneously in the same scenario, and their multi-modal sensors will keep capturing the surrounding information. The CDA data saver will provide convenient APIs to convert the online sensor data to an appropriate format and save it to the disk, which can be directly used for deep learning training. Please refer to the Section III-B to see more details of the data saved by the CDA Data Saver.Moreover, the CDA data loader will provide interfaces to replay the collected simulated data to evaluate how the trained model performs when integrating with the full-stack system. It is also worth noting that the CDA data loader can also replay real-world automated driving data if it has high-definition maps (HDMap) in osm format with high-precision GPS data.

Component Interactions: The components of the OpenCDA simulation tool interact with each other in an organized manner. The integrated simulators provide sensor data streaming, set up the reinforcement learning environment, and control the background traffic for each CAV. Each CAV is controlled by its own cooperative driving system, which receives sensor data and environment metadata and generates driving decisions and control commands based on this information. These control commands are then sent back to the simulators to update the simulation environment. The scenario manager component defines the dynamic driving task and scenario for each CAV, and upon completion of the simulation, it collects information from each cooperative driving system to provide driving evaluations. If the CDA data loader component is activated, the scenarios can be directly loaded from offline simulated or real-world data, bypassing the scenario manager. During the simulation, each CAV can also use the CDA data saver component to dump collected sensor data and create a new simulation dataset.

Development Status: The OpenCDA simulation tool is continually being updated and expanded with new features. Currently, several components, such as the Human Interaction Layer, the End-to-end Learning pipeline, and the reinforcement learning environment integration in Fig. 1, are still under development. Research teams from various universities across different countries are actively working to improve these components. For example, the team from UC Davis is integrating a multi-agent reinforcement learning framework into OpenCDA, while scholars from Nankai University are using OpenCDA in distributed systems research. Additionally, the team from the Technical University of Munich is working to make OpenCDA compatible with the NS3 network simulator.

2) OpenCOOD: OpenCOOD[2] is the first code framework using PyTorch [61] for cooperative detection and segmentation, foundation for cooperative perception research. Unlike popular 3D detection frameworks such as OpenPCDet [62] and MMDetection3D [63], OpenCOOD deals with multi-agent perception data, where multi-view sensor data streaming will be provided for the same scene simultaneously. As Fig. 2 reveals, there are 6 major components:

- *Config system:* OpenCOOD incorporates modular and inheritance design into the config system to enable users conveniently modify the model/training/inference parameters. Specifically, we use YAML files to configure all the important parameters.
- *Dataset class:* Opencood has 3 different classes that inherit from the base dataset class to convert the multi-agent

[2] [Online]. Available: https://github.com/DerrickXuNu/OpenCOOD

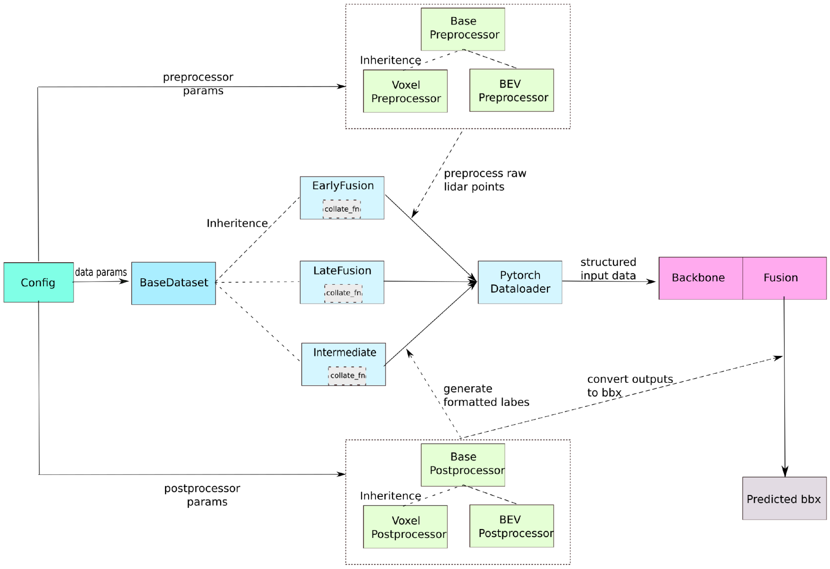


Fig. 2. OpenCOOD pipeline for cooperative perception.

data into formats that common fusion strategies (i.e., early fusion, late fusion, and intermediate fusion) can use.

- *Preprocessor and backbone:* The preprocessor and backbone together will convert the raw point clouds to voxels or 2D bird-eye-view features. Please refer to Section III-A3 to see the backbones this framework supports.
- *Multi-agent fusion:* The fusion module will aggregate the shared raw point cloud, bounding boxes, or neural features from different vehicles to output a holistic fused feature. Refer to Section III-A3 to check the state-of-the-art fusion models we support.
- *Postprocessor:* The postprocessor will convert the model's outputs to the correct bounding box formats that can be utilized for visualization and evaluation.

Since its initial release, OpenCOOD has gained a significant amount of support from various research teams who have contributed to the model zoo and added support for additional datasets and modalities. For instance, the team from Shanghai Jiaotong University has added new models like Where2Comm [33] and When2Comm [64] to OpenCOOD, while teams from Queen's University and Leibniz University Hannover have added support for datasets such as CODD [65] and multi-modality functionality to the platform.

3) Model Libraries: The OpenCDA ecosystem offers a vast collection of models, including both traditional algorithms and deep learning models, which come pre-trained and can be utilized to facilate the development of CDA. This model library can be utilized not only within the OpenCDA ecosystem but also on other platforms. Upon this time, the model library includes the following:

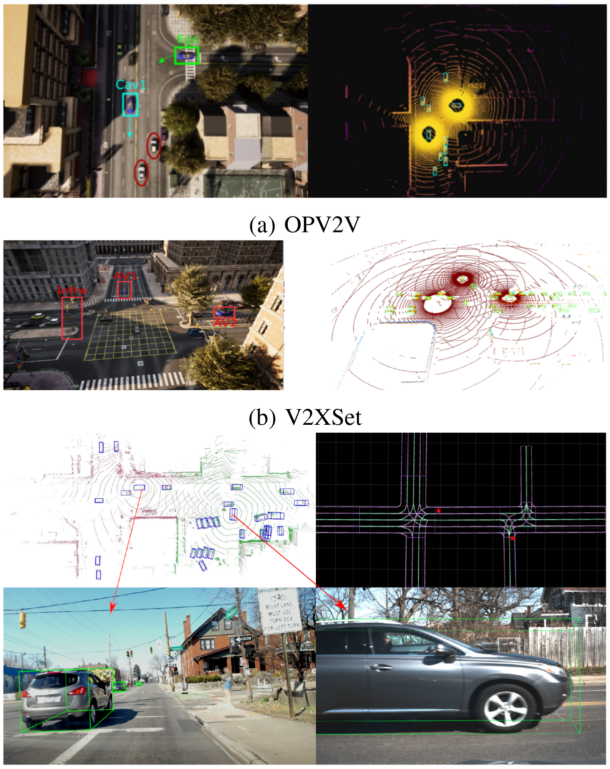
- *3D LiDAR-based Object Detection:* VoxelNet [66], PointPillar [67], Pixor [68], SECOND [67].
- *Cooperative Perception:* V2X-ViT [32], AttFuse [13], CoBEVT [55], Where2Comm [33], FPV-RCNN [51], F-Cooper [70], DiscoNet [48], V2VNet [12].
- *Localization:* Kalman Filter [71], Extended Kalman Filter [72].
- *Planning:* A* Searching, Spline Cubic Polynomial trajectory generation.
- *Platooning:* Finite state machine [42].
- *Control:* PID control [73], MPC [74].

B. Data

Data is an indispensable part of the OpenCDA. Especially for the perception system and learning-based decision-making/control, the models require a large-scale dataset for offline training in OpenCDA. Supported by the CDA data manager of OpenCDA, we have released two simulated large-scale datasets for cooperative perception. We also sent two L4 automated vehicles together to collect real-world data and develop a data acquisition and processing platform [75] to facilitate field experiments for CDA.

1) Simulated Dataset: By utilizing OpenCDA and CARLA, we collect the first large-scale simulated V2V perception dataset OPV2V [13] [see Fig. 3(a) for an example]. OPV2V contains 11 464 frames of 3D LiDAR point clouds and RGB images from multiple vehicles in the same scenario with 230 K annotated 3D boxes. It supports multiple cooperative perception tasks, including cooperative detection, segmentation, and tracking. Since OPV2V was released, it has served as the foundation of cooperative perception [32], [33], [55], [56], [57]. Since OPV2V is a major focus on Vehicle-to-Vehicle cooperation, we proposed another dataset named V2XSet [32], which adds the roadside units on OPV2V to enable Vehicle-to-Infrastructure cooperation. A data sample of V2XSet can be found in Fig. 3(b).

2) Real-World Dataset: To accelerate the process of CDA research moving from simulation to the real world, we proposed an automated driving system (ADS) data acquisition and analytics platform (ADAAP) for vehicle trajectory extraction, reconstruction, and evaluation based on the connected automated vehicle (CAV) cooperative perception [75]. This platform presents a holistic pipeline from the raw advanced sensory data collection to data processing, which can process the sensor data



(a) OPV2V

(b) V2XSet

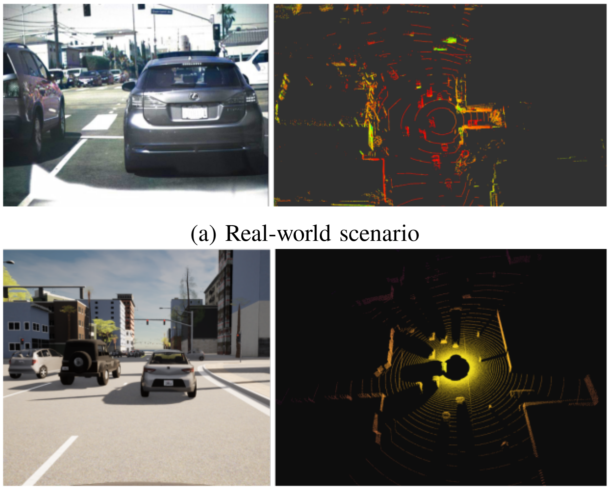
(c) Real-world data. *Left Up*: The aggregated 3D LiDAR. *Right Up*: The annotated HDMap, where the stars indicate the position of the two collection vehicles. *Bottom Row*: The front cameras of the two vehicles.

Fig. 3. Data samples from different datasets in the OpenCDA ecosystem.

from multi-CAVs and extract the objects' Identity (ID) number, position, speed, and orientation information in the map and Frenet coordinates. Besides the dynamic object information, this platform can also help users to generate HDMap directly from raw point clouds. Employing this platform, we sent out two CAVs and collected a real-world cooperative perception dataset that includes LiDAR point clouds, RGB camera images, annotated 3D bounding boxes, and HDMaps. A good sample of the collected dataset can be seen in Fig. 3(c). More importantly, our OpenCDA simulation tool is able to create a digital twin for the real-world dataset given the object annotation and xodr format map file. An example is shown in Fig. 4, where we load a real-world scenario into CARLA. The data acquisition platform, along with the dataset, will be open-sourced as part of the ecosystem soon.

C. Scenario Database

OpenCDA offers a scenario database, generation, and management tools, including various rule-based standard scenarios for testing different cooperative driving applications as benchmarks. These diverse scenarios will be used to evaluate the safety of driving systems in simulation. Since human-defined scenarios can only cover limited cases, we also develop an adversarial scene generator V2XP-ASG [76] to automatically infinitely generate challenging corner cases. Users can easily replace any default modules in the OpenCDA with their designs and test them in the supplied or automatically generated scenarios. If users desire to produce their scenarios, our framework also provides simple APIs to support such customization.



(a) Real-world scenario

(b) Corresponding digital twin in CARLA

Fig. 4. Our OpenCDA ecosystem can create a digital twin in CARLA for the real-world dataset for Culver City in Los Angeles.

1) Rule-Based Scenarios: Our standard scenario database contains a series of YAML files that defines the specific scenario's parameters, including weather, traffic states, testing locations, etc. This database can be loaded by the OpenCDA simulation tool's scenario manager and any other platforms that follow the API format. The current scenario database includes classic scenes for platooning, cooperative merge, intersection control, etc.

2) Adversarial Scenarios: Rule-based scenarios only consider the common scenes and lack the capability of performing stress t ests on corner cases for the target perception systems. Thus, we introduce V2XP-ASG [76] – the first open adversarial scene generator for LiDAR-based V2X perception systems. Unlike rule-based scenarios, V2XP-ASG can automatically generate abundant challenging scenes in an adversarial manner. V2XP-ASG first constructs an adversarial collaboration graph by searching for adversarial collaborators whose viewpoints combination will lead to inferior performance. Afterward, it perturbs multiple agents' poses in an adversarial and plausible way. To reflect the updated LiDAR observation caused by sensor viewpoint change and pose perturbation, the high-fidelity CARLA [20] simulator is used to render the sensor measurements. Extensive experiments show that V2XP-ASG can create challenging scenes to severely deteriorate the performance of modern V2X perception systems. More importantly, training in these adversarial scenes can increase the system's accuracy.

D. Community

1) Ecosystem Growth: OpenCDA aims to establish a community of practice (CoP) made up of various stakeholders in the industry, including developers of automated driving systems

(ADS), OEMs, infrastructure technology developers, and universities and research institutes. The CoP has been increasing in its size since OpenCDA was released. The CoP provides feedback that will be incorporated into the OSE roadmap, and we will host webinars, panel discussions, and workshops to invite industry leaders, researchers, and government technical managers to share their insights. The OSE is initially based on our toolchains, but users can also create subsidiary software tools to perform specific tasks. Within the diverse backgrounds of the stakeholders, the roadmap can make the ecosystem easily support various customizations for different groups of people. To encourage innovation and growth within the ecosystem, we will also hold a series of challenges for users to develop automated systems using the platform and open their code. These challenges, called the OpenCDA Automated Driving Challenge, will be held annually or bi-annually in collaboration with conferences, such as ICRA, IROS, ITSC, and CVPR. Participants will be given reproducible, diverse scenarios to test their systems on and will be evaluated based on metrics, such as pass ratio and collision rate. The challenge will have two tracks: a full-stack system challenge and a specific component challenge. The topics for these tracks will be based on the updated roadmap, which is developed through stakeholder engagement in the CoP.

2) Community Building: The OpenCDA ecosystem also engages with a group of early adopters called the Community of Content Contributors (CCC) to help develop and maintain the platform. The CCC consists of researchers and developers in the AV/intelligent transportation industry and academia who have expressed interest in using and contributing to the platform and have done related work. We aim to hold monthly webinars to discuss the needs and priorities of the CCC and develop strategies to attract additional content contributors. In the beginning, our core development team is dedicated to providing start kits and tutorials for new users and contributors and build introductory materials in Python and Google Colab to lower the entry barrier for participation. These strategies for community building will be detailed in a CCC memorandum.

3) Governance and Organization: OpenCDA will determine an appropriate governance model through discussions with COP and CCC. Many open-source governance models exist, including founder-leader, self-appointing council, electoral, and foundation-backed models. The most suitable one will be selected based on factors, such as flexibility, simplicity, and scalability, as well as considerations like the proprietary nature of AV industry technologies and the availability of resources like vehicle testing and data collection. The chosen model should also support continuous development, integration, and deployment processes and infrastructure for open, asynchronous, and distributed development of the open-source product and ensure sustainability goals for the OSE. Currently, at the beginning stage of the ecosystem, we adopted the founder-led model, meaning that our core development team at UCLA takes the lead. The goal is to move away from a founder-led model to a community-managed approach for sustainability and scalability, although the foundation-backed model will also be explored. The governance model will also address development methodology, quality control, security and privacy, support for users, and onboarding mechanisms for new contributors.

Continuous Development and Quality Control: We aim to adopt a sustainable software development process that supports itself in the long run. Fig. 6 demonstrates the life cycle of the software development for ecosystem. As a standard process, the life cycle consists of two essential stages: the development stage and the review stage. In the development stage, a developer from the CCC will be assigned to a task based on the desired functionality from the roadmap or an issue raised by a user. The developer will design a pipeline, implement the code, write unit tests, and integrate the new functionality with the rest of the system. The developer will then create a pull request to move to the review stage. In the review stage, a qualified reviewer will statically and dynamically analyze the code and request any necessary improvements. If the pull request is approved, it will be merged into the correct branch/version of the codebase after passing an online CI/CD build. Eventually, the reviewer will merge this pull request to the correct branch/version of the codebase.

IV. Case Studies

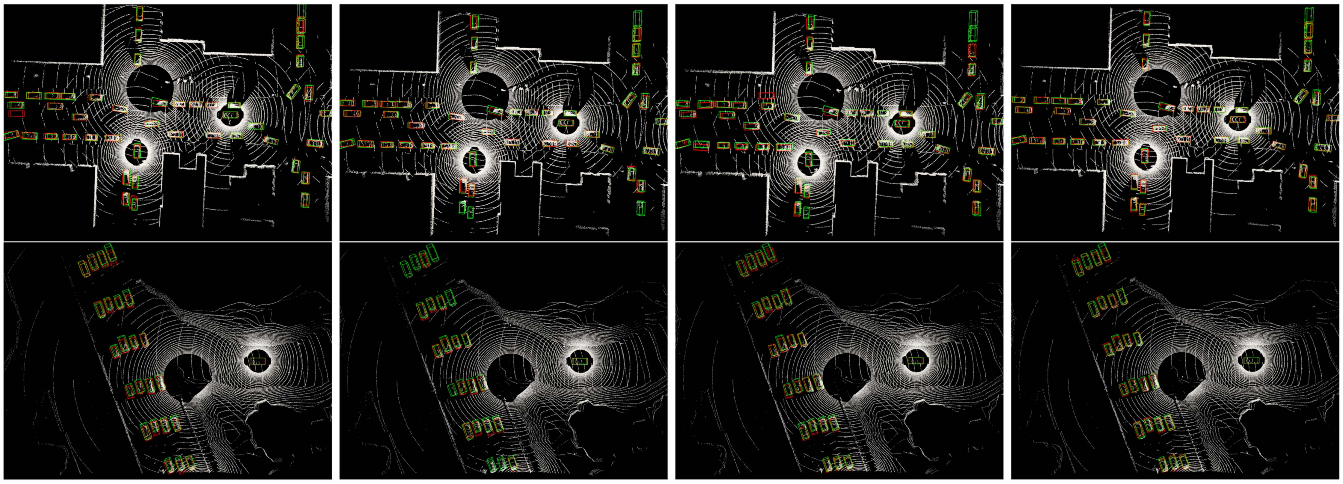
This section will conduct several case studies for different components of our OpenCDA ecosystem to demonstrate examples of advanced CDA research that can be enabled by OpenCDA. They include cooperative 3D LiDAR detection, cooperative merging with the classic modular pipeline, camera-based cooperative perception, and adversarial scene generation for the V2X perception system.

A. Case Study 1: Cooperative 3D LiDAR Detection

Scope: In this case study, we demonstrate how OpenCDA enables cooperative LiDAR detection research for cooperative driving. We employ OpenCOOD to perform the cooperative 3D LiDAR detection task on three datasets from the OpenCDA ecosystem, OPV2V, V2XSet, and paritial of the real-world dataset. The partial of our real-world dataset contains 14 000/2000/4000 frames for train/validation/test set, respectively. We ensure the real-world dataset has the same format as the simulated dataset to be compatible with OpenCOOD. This requires models to be able to leverage multiple LiDAR views from different vehicles/infrastructures to perform 3D object detection on the ego vehicle.

Evaluation: Th evaluation range in x and y direction are $[-140, 140]$ m and $[-40, 40]$ m respectively. The detection performance is measured with Average Precisions (AP) at Intersection-over-Union (IoU) thresholds of 0.5 and 0.7. In this study, we focus on LiDAR-based vehicle detection. Vehicles hit by at least one LiDAR point from any connected agent will be included as evaluation targets.

Implementation details: During training, a random AV is selected as the ego vehicle, while during testing, we evaluate a fixed ego vehicle for all the compared models. The communication range of each agent is set as 70 m based on, whereas all the agents out of this broadcasting radius of ego vehicle are ignored. We



(a) OPV2V [13] (b) V2VNet [12] (c) DiscoNet [48] (d) V2X-ViT [32]

Fig. 5. Qualitative comparison in a congested intersection and a highway entrance ramp in V2XSet. Green and red 3D bounding boxes represent the ground truth and prediction, respectively.

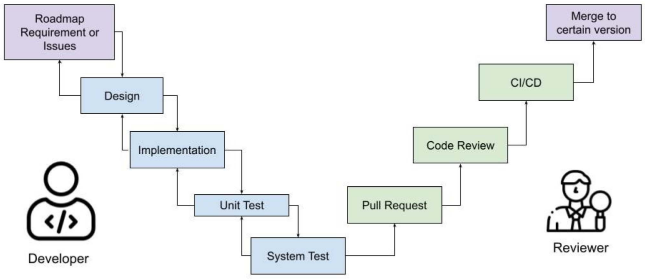


Fig. 6. The healthy and self-sustainable life cycle of software development for the ecosystem.

TABLE I
3D Detection Performance Comparison on OPV2V and V2XSet. We Show Average Precision (AP) At IoU = 0.5, 0.7

	OPV2V		V2XSet		Real-world	
Models	AP0.5	AP0.7	AP0.5	AP0.7	AP0.5	AP0.7
No Fusion	0.679	0.602	0.606	0.402	0.287	0.134
Late Fusion	0.858	0.781	0.549	0.307	0.439	0.219
Early Fusion	0.891	0.800	0.720	0.384	0.482	0.267
F-Cooper [70]	0.887	0.790	0.715	0.469	0.456	0.241
AttFuse [13]	**0.908**	0.815	0.709	0.487	0.479	0.272
V2VNet [12]	0.897	0.822	0.791	0.493	0.470	0.258
DiscoNet [48]	0.899	**0.836**	0.798	0.541	0.468	0.253
V2X-ViT [32]	0.891	0.826	**0.836**	**0.614**	**0.489**	**0.295**

select PointPillar [67] as the backbone to convert raw LiDAR point clouds to 2D features.

Compared models: From our model zoo, we select SOTA multi-agent fusion models *No Fusion* (baseline), *Early Fusion*, *Late Fusion*, AttFuse [13], DiscoNet [48], V2VNet [12], F-Cooper [70], and V2X-ViT [32] to perform the task.

Results: Table I shows the detection performance of different models on the three cooperative perception datasets. It is clear that all the models that leverage the collaborations between agents outperform the single-vehicle perception baseline *No Fusion* by a large margin on both datasets. In the OPV2V dataset, DiscoNet [48] achieves the best performance, slightly higher than the second best method V2x-ViT [32] by 1.0% on AP@0.5. While in the V2XSet and real-world dataset, V2X-ViT is the leading algorithm, outperforming DiscoNet by a large extent (7% and 4.2%). This is reasonable as V2X-ViT has specific modules to handle GPS error and vehicle-infrastructure sensor heterogeneity, which only exist in the V2XSet and real-world dataset. Fig. 10 further demonstrates the qualitative results of selected SOTA models on two different scenarios in V2XSet. These results show how OpenCOOD can enable research studies on cooperative perception.

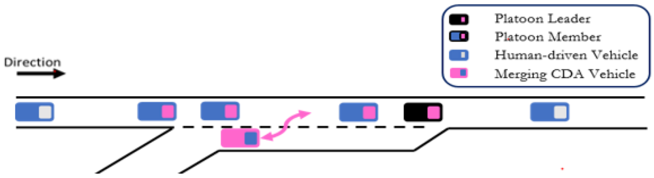


Fig. 7. The cooperative merge testing scenario.

B. Case Study 2: Cooperative Merge Into Platoons

Scope: In this case study, we demonstrate how OpenCDA enables researchers to flexibly use the co-simulation functionality to flexibly develop and test cooperative decision making and control algorithms. We conduct experiments for cooperative merge using the OpenCDA simulation tool (co-simulation functionality). The scenario design is shown in Fig. 7. The mainline has a high-speed traffic flow mixed with human-driven vehicles managed by SUMO and CAVs that are controlled by our own cooperative driving systems. When the single CAV is near the merging area, it will communicate with the mainline platoon and request to join. Once they agree, the single CAV has to finish the merge and join the platoon simultaneously before the acceleration lane ends. The leader will decide the

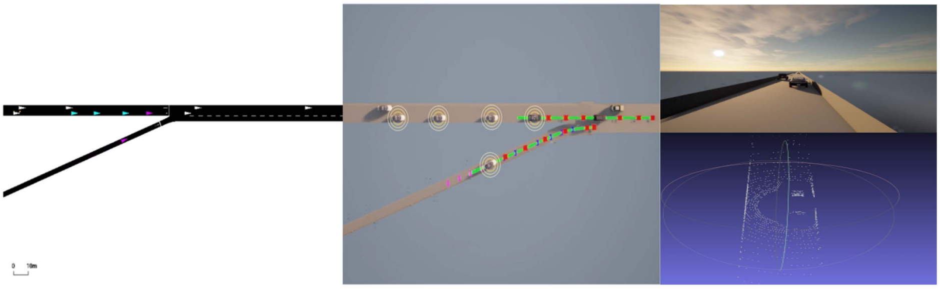
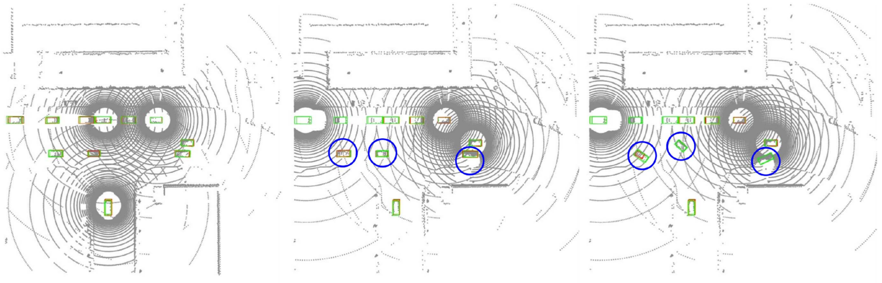


Fig. 8. A snippet of cooperative merge scenario testing under co-simulation setting.



(a) Initial scene (b) After ACS (c) Final scene

Fig. 9. Qualitative visualization of V2XP-ASG generated challenging scenes. (a) Initial scene. (b) Scene after adversarial collaborator search. (c) Scene after adversarial pose perturbation. Here blue circles represent the perturbed vehicles.

best merging position and command certain platoon members to create a gap for the new member. Fig. 8 shows a snippet of the platooning co-simulation testing using a customized benchmark map of a basic freeway merge segment included in the OpenCDA simulation tool.

Evaluation: The whole system is evaluated using safety, stability, and efficiency.

- *Safety:* We evaluate the safety using two indicators, Time-to-Collision (TTC) and hazard frequency. TTC refers to the time required for two vehicles to collide if they continue at their present speed and on the same path. Hazard frequency is defined as the number of events that $TTC < TTC_t$, where TTC_t is the warning threshold of TTC to distinguish between safe and unsafe events. We set the threshold as 2.5 seconds.
- *Stability:* The stability of a platoon refers to whether oscillations are amplified from downstream to upstream vehicles. Two metrics are used to measure the stability: inter-vehicular time gap and accelerations. The inter-vehicular time gap is the time difference between a platoon member and the preceding member. The time-series data of acceleration and statistics of the data, such as mean and standard deviation, are calculated to indicate the driving smoothness of the platoon members.
- *Efficiency:* The efficiency of platoon joining is measured by the time duration required for the maneuver and the smoothness of the process. The time it takes to complete the maneuver, and the acceleration during the joining process are used as indicators of efficiency.

Implementation details: In this testing scenario, there are a total of five connected and automated vehicles (CAVs), including four in the platoon and one on the entrance ramp. All CAVs have the classic modular driving pipeline, including perception, localization, planning, and control. For perception, Yolov5 [77] with simple late LiDAR fusion is used for 3D object detection. Note that Yolov5 can be replaced by any other perception model, such as [78], which adds a bidirectional feature pyramid network, tiny detection layer, and robust attention modules to achieve better performance. For localization, Kalman Filter with GPS and IMU is used, and PID controller is used for control. In terms of planning, the leading vehicle of the platoon uses a normal rule-based trajectory planning, while the others use the Platooning finite state machines algorithm [42].

Compared models: A critical decision for cooperative merging is the merging position. Here we compare two different algorithms: the heuristic-based method and Genetic Fuzzy System (GFS) [79]. The first approach will choose the vehicle in the platoon that has the shortest Euclidean distance as the frontal vehicle for merging. The second one utilizes fuzzy logic to decide the best merging position. Different from a heuristic-based method, it also takes platoon members' speed and surrounding human-driven vehicles' information into consideration.

Results: Table II reveals the cooperative merging results with two different algorithms. Both algorithms achieve good safety and stability with high average TTC, 0 hazard frequency, and slight acceleration fluctuations for platooning. In terms of efficiency, the GFS method performs favorably against the heuristic method.

TABLE II
Quantitative Results of Two Different Scenario Tests. The Desired Platoon Time Gap Is Set to 0.6 Seconds. Attc: average time-to-collision(second), Hf: hazard frequency(number of times), Atg:average Platoon Time gap(second), Tg_std:platoon Time Gap Standard deviation(second), Tcm: Time to Complete maneuver(second)

Vehicle id	Safety		Stability			Efficiency	
	attc	hf	atg	tg_std	acc_std	tcm	acc_std
0	49.8	0	NA	NA	0.92	NA	0.03
1	25.5	0	0.607	0.005	0.63	NA	0.01
2	31.03	0	0.607	0.003	0.56	NA	0.01
3	31.53	0	0.612	0.013	1.33	13.5	2.83
4	32.59	0	0.707	0.23	0.82	NA	1.37

(a) Cooperative merge and platoon join using heuristic method

Vehicle id	Safety		Stability			Efficiency	
	attc	hf	atg	tg_std	acc_std	tcm	acc_std
0	49.8	0	NA	NA	0.95	NA	0.02
1	31.40	0	0.608	0.007	1.27	9.9	2.51
2	31.24	0	0.674	0.16	0.79	NA	1.18
3	30.14	0	0.609	0.008	0.65	NA	0.88
4	29.9	0	0.607	0.006	0.61	NA	0.59

(b) Cooperative merge and platoon join using GFS

C. *Case Study 3: Camera-Based Cooperative Perception*

Scope: In this case study, we demonstrate how OpenCDA enables researchers to innovate in multi-modal sensing and perception research via the use our flexible framework and data processing structure. We aim to investigate the use of camera-based cooperative perception approaches for the task of bird's-eye-view (BEV) semantic segmentation [55], [80] in dynamic environments. To do this, we use both the OpenCDA simulation tool and OpenCOOD, focusing on the OPV2V dataset. Since the OPV2V dataset does not include map information, we use the driving log replay functionality in the OpenCDA CDA Data Loader to generate ground truth for the HDMap. Each connected vehicle in the case study has four cameras facing in different directions. To perform the BEV segmentation, the models must first infer 3D information from the multi-view 2D images and then aggregate the inferred 3D information from all the vehicles to predict the BEV segmentation mask.

Evaluation: During testing, we select a fixed vehicle as the ego vehicle and evaluate the performance of our models within a 100 m × 100 m area around it with a map resolution of 39 cm. We use the intersection over union (IoU) between the predicted map and the ground truth map-view labels as our performance metric. We segment both dynamic objects and the topology of the map.

Compared models: For the multi-agent perception task, we consider a single-agent perception system without fusion as the baseline. We compare the performance of this baseline system with state-of-the-art multi-agent perception algorithms such as F-Cooper [70], AttFuse [13], V2VNet [12], DiscoNet [48], and CoBEVT [55]. Additionally, we also implement a straightforward fusion strategy called Map Fusion, which transmits the segmentation map instead of BEV features and fuses all the maps by selecting the closest agent's prediction for each pixel.

Results: As shown in Tab. III, all cooperative methods perform better than *No Fusion*, which proves the benefits of multi-agent perception system. Among all fusion models, CoBEVT achieves the best IoU for all classes, outperforming the second-best method by 7.5%, 2.3%, and 7.2% on the vehicle, drivable area, and lane, respectively. We also show the qualitative results of CoBEVT on scenes containing 3 AVs in Fig. 10. In each row, we draw the front camera image of each AV, along with the ground truth and prediction pairs. Through collaboration, our framework can overcome most of the occlusions and perceive distant objects accurately, benefiting from our Transformer design that learns from all agents and views.

TABLE III
Map-View Segmentation on OPV2V Camera-Track. We Report IoU for All Classes

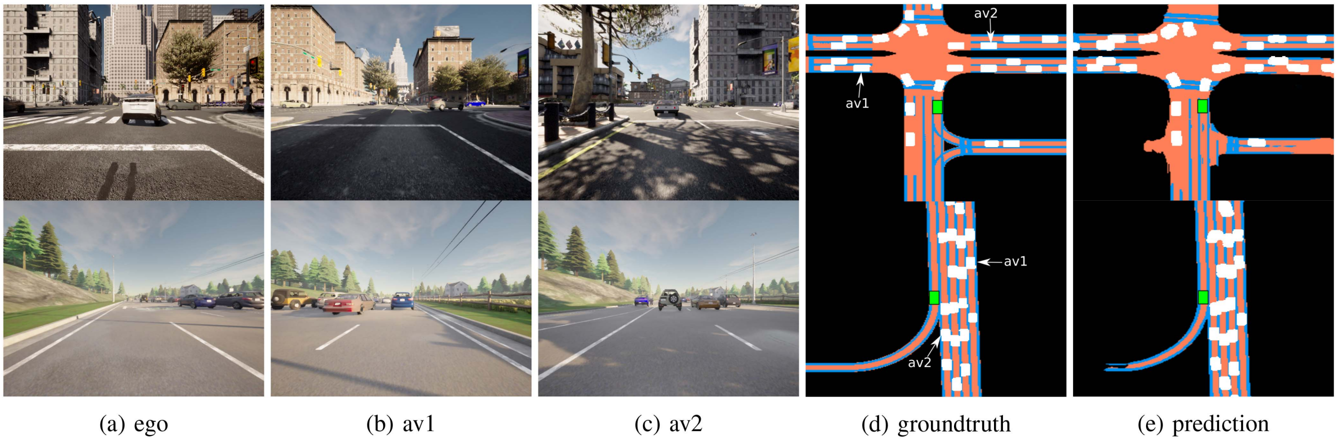
Method	Veh.	Dr.Area	Lane
No Fusion	37.7	57.8	43.7
Map Fusion	45.1	60.0	44.1
F-Cooper [70]	52.5	60.4	46.5
AttFuse [13]	51.9	60.5	46.2
V2VNet [12]	53.5	60.2	47.5
DiscoNet [48]	52.9	60.7	45.8
CoBEVT	**60.4**	**63.0**	**53.0**

D. *Case Study 4: Adversarial Scene Generation*

Scope: In this case study, we show how OpenCDA can be used as a framework for generating challenging scenarios for testing and training algorithms. We use V2XP-ASG [76], one of our core team's algorithms, to create 331 challenging scenes for V2X perception systems from the augmented OPV2V dataset. Scenes are modified in an adversarial and realistic way by manipulating agent positions and collaborator choices. The approach has two stages: 1) adversarial collaborator search based on fusion model attention weights and 2) adversarial perturbation search using black-box optimization with historical observations and loss pairs. The perturbations are physically plausible by eliminating collisions and out-of-range positions.

Evaluation: We benchmark V2XP-ASG's performance for several cooperative perception algorithms, including No Fusion, Late Fusion, Early Fusion, and AttFuse [13]. We conduct the experiments based on the CARLA simulator and evaluate the model for the range: $x, y \in [-48, 48]$ m. For consistency, the ego agent is fixed during the whole optimization process and evaluation. To reduce the influence of noise on the perception performance, we set zero noise for LiDAR simulation so that the performance drop is solely dependent on the intrinsic scene configurations.

Results: In Table IV, we evaluate the performance drop of various V2X fusion methods on challenging scenes generated by V2XP-ASG, compared to normal scenes. Our results indicate that V2XP-ASG effectively creates challenging scenarios for diverse fusion models. Fig. 10 presents a qualitative analysis of the adversarial search impact on detection, where mis-detected vehicles can be seen, presenting a significant challenge for V2X systems.



(a) ego (b) av1 (c) av2 (d) groundtruth (e) prediction

Fig. 10. Qualitative results of case study 3. From left to right: the front camera image of (a) ego, (b) av1, (c) av2, (d) groundtruth and (e) prediction of CoBEVT. The green bounding boxes represent ego vehicle, while the white boxes denote the segmented vehicles. CoBEVT demonstrates robust performance under various traffic situations and road types.

TABLE IV
EVALUATION OF V2X PERCEPTION MODELS ON INITIAL *NORMAL* SCENES AND V2XP-ASG GENERATED *CHALLENGING* SCENES. THE NUMBER IN BOLD IS THE AP DROP

Methods	Scenes Type	AP@0.3	AP@0.5	AP@0.7
No Fusion	Normal Challenging	55.4 31.1(**-24.3**)	54.8 30.3(**-24.5**)	46.3 25.1(**-21.2**)
Late Fusion	Normal Challenging	73.4 42.0(**-31.4**)	72.6 40.0(**-32.6**)	62.6 32.8(**-29.8**)
Early Fusion	Normal Challenging	80.3 49.8(**-30.5**)	79.8 48.3(**-31.5**)	73.5 43.4(**-30.1**)
AttFuse	Normal Challenging	82.4 46.6(**-35.8**)	81.5 44.9(**-36.6**)	74.6 40.5(**-34.1**)

V. CONCLUSIONS

In this paper, we introduce the OpenCDA ecosystem, which integrates a model zoo, a suite of driving simulators at various resolutions, benchmark training/testing datasets, and a scenario database/generator for CDA research. We describe how the ecosystem work and grows and conduct several case studies to demonstrate its capability of facilitating CDA research from different perspectives.

In the future, we plan to continue expanding and improving our OpenCDA simulation tool by adding missing components such as a human interaction layer and end-to-end learning. We will also continually grow our model zoo to support more research studies. Additionally, we plan to interface with real systems in various ways. For example, the perception model trained using OpenCOOD can be transferred to real vehicles, and we will provide a corresponding toolchain to help users easily migrate the models for real-world deployment. We also plan to develop a ROS version of the OpenCDA simulation tool, which can interact with the real system in two ways: 1) it can be used to provide the environment for hardware-in-the-loop testing, and 2) the components of the cooperative driving system (e.g., perception, planning) in OpenCDA can be directly deployed on the real vehicle.

The development of OpenCDA will also heavily rely on joint efforts from the community. As OpenCDA gradually transits from the founder-led model to a community-managed approach, contributions from the community to the ecosystem are vital for the future continuous development of OpenCDA. We strongly encourage existing users of OpenCDA and other CDA developers to contribute their work to the OpenCDA community.

REFERENCES

[1] R. Xu, Y. Guo, X. Han, X. Xia, H. Xiang, and J. Ma, “OpenCDA: An open cooperative driving automation framework integrated with co-simulation,” in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 1155–1162.

[2] R. Valiente, B. Toghi, R. Pedarsani, and Y. P. Fallah, “Robustness and adaptability of reinforcement learning-based cooperative autonomous driving in mixed-autonomy traffic,” *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 397–410, 2022.

[3] M. Hua, G. Chen, B. Zhang, and Y. Huang, “A hierarchical energy efficiency optimization control strategy for distributed drive electric vehicles,” *Proc. Inst. Mech. Engineers, Part D: J. Automobile Eng.*, vol. 233, no. 3, pp. 605–621, 2019.

[4] G. Chen, M. Hua, C. Zong, B. Zhang, and Y. Huang, “Comprehensive chassis control strategy of FWIC-EV based on sliding mode control,” *IET Intell. Transport Syst.*, vol. 13, no. 4, pp. 703–713, 2019.

[5] S. C. Calvert and G. Mecacci, “A conceptual control system description of cooperative and automated driving in mixed urban traffic with meaningful human control for design and evaluation,” *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 147–158, 2020.

[6] R. A. Shet and S. Yao, “Cooperative driving in mixed traffic: An infrastructure-assisted approach,” *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 429–447, 2021.

[7] A. Ferrara, G. P. Incremona, E. Birliba, and P. Goatin, “Multi-scale model based hierarchical control of freeway traffic via platoons of connected and automated vehicles,” *IEEE Open J. Intell. Transp. Syst.*, 2022.

[8] W. Liu, X. Xia, L. Xiong, Y. Lu, L. Gao, and Z. Yu, “Automated vehicle sideslip angle estimation considering signal measurement characteristic,” *IEEE Sensors J.*, vol. 21, no. 19, pp. 21675–21687, Oct. 2021.

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

[10] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, “Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception,” *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.

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

[12] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2VNet: Vehicle-to-vehicle communication for joint perception and prediction," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 605–621.

[13] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 2583–2589.

[14] Z. Bai et al., "A survey and framework of cooperative perception: From heterogeneous singleton to hierarchical cooperation," 2022, *arXiv:2208.10590*.

[15] H. Hu, Z. Liu, S. Chitlangia, A. Agnihotri, and D. Zhao, "Investigating the impact of multi-LiDAR placement on object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2550–2559.

[16] Q. Li, W. Lin, Z. Liu, and A. Prorok, "Message-aware graph attention networks for large-scale multi-robot path planning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5533–5540, Jul. 2021.

[17] J. Blumenkamp, Q. Li, B. Wang, Z. Liu, and A. Prorok, "Learning to navigate using visual sensor networks," 2022, *arXiv:2208.00759*.

[18] K. Ma, H. Wang, and T. Ruan, "Analysis of road capacity and pollutant emissions: Impacts of connected and automated vehicle platoons on traffic flow," *Physica A: Stat. Mechanics Appl.*, vol. 583, 2021, Art. no. 126301.

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

[20] H. Hu, B. Yang, Z. Qiao, S. Liu, D. Zhao, and H. Wang, "Seasondepth: Cross-season monocular depth prediction dataset and benchmark under multiple environments," 2020, *arXiv:2011.04408*.

[21] Z. Wang, K. Han, and P. Tiwari, "Digital twin-assisted cooperative driving at non-signalized intersections," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 198–209, Jun. 2022.

[22] Z. Zuo et al., "MPC-based cooperative control strategy of path planning and trajectory tracking for intelligent vehicles," *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 513–522, Sep. 2021.

[23] R. van Hoek, J. Ploeg, and H. Nijmeijer, "Cooperative driving of automated vehicles using B-splines for trajectory planning," *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 594–604, Sep. 2021.

[24] E. van Nunen, J. Reinders, E. Semsar-Kazerooni, and N. van de Wouw, "String stable model predictive cooperative adaptive cruise control for heterogeneous platoons," *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 186–196, Jun. 2019.

[25] Y. Tian, J. Wang, Y. Wang, C. Zhao, F. Yao, and X. Wang, "Federated vehicular transformers and their federations: Privacy-preserving computing and cooperation for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 456–465, Sep. 2022.

[26] W. Wu, Z. Peng, L. Liu, and D. Wang, "A general safety-certified cooperative control architecture for interconnected intelligent surface vehicles with applications to vessel train," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 627–637, Sep. 2022.

[27] M. Hasan, S. Mohan, T. Shimizu, and H. Lu, "Securing vehicle-to-everything (V2X) communication platforms," *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 693–713, Dec. 2020.

[28] H. Xie, Y. Wang, X. Su, S. Wang, and L. Wang, "Safe driving model based on V2V vehicle communication," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 449–457, 2022.

[29] Q. Han, Y.-S. Zhou, Y.-X. Tang, X.-G. Tuo, and P. He, "Event-triggered finite-time sliding mode control for leader-following second-order nonlinear multi-agent systems," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 570–579, 2022.

[30] S. E. Shladover, "Opportunities and challenges in cooperative road vehicle automation," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 216–224, 2021.

[31] On-Road Automated Driving (ORAD) Committee, "SAE J3216 standard: Taxonomy and definitions for terms related to cooperative driving automation for on-road motor vehicles," SAE International Standard. J3216_202005, 2020.

[32] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2X-ViT: Vehicle-to-everything cooperative perception," in *Proc. Comput. Vis.: 17th Eur. Conf.*, 2022, pp. 107–124, doi: 10.1007/978-3-031-19842-7_7.

[33] Y. Hu, S. Fang, Z. Lei, Y. Zhong, and S. Chen, "Where2comm: Communication-efficient collaborative perception via spatial confidence maps," 2022, *arXiv:2209.12836*.

[34] I. B. Viana, H. Kanchwala, K. Ahiska, and N. Aouf, "A comparison of trajectory planning and control frameworks for cooperative autonomous driving," *J. Dyn. Syst., Meas., Control*, vol. 143, no. 7, 2021.

[35] T. Lochrane, L. Dailey, and C. Tucker, "Carma^{SM}: Driving innovation," *Public Roads*, vol. 83, no. 4, 2020. [Online]. Available: https://its.dot.gov/cda/

[36] 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.

[37] 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/

[38] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," vol. 10, 2017, *arXiv:1710.05465*.

[39] L. Hao et al., "Using cooperative adaptive cruise control (CACC) to form high-performance vehicle streams.final report," 2018.

[40] E. S. Steven, C. Nowakowski, X. Y. Lu, and R. Ferlis, "Cooperative adaptive cruise control (CACC) definitions and operating concepts," in *Proc. TRB Annu. Meeting*, vol. 1, Jan. 2015, pp. 1–16.

[41] J. Thunberg, G. Sidorenko, K. Sjöberg, and A. Vinel, "Efficiently bounding the probabilities of vehicle collision at intelligent intersections," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 47–59, 2021.

[42] X. Han et al., "Strategic and tactical decision-making for cooperative vehicle platooning with organized behavior on multi-lane highways," *Transp. Res. Part C: Emerg. Technol.*, vol. 145, 2022, Art. no. 103952.

[43] H. Zhang, J. Liu, Z. Wang, H. Yan, and C. Zhang, "Distributed adaptive event-triggered control and stability analysis for vehicular platoon," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1627–1638, Mar. 2021.

[44] Z. Zhou, F. Zhu, D. Xu, B. Chen, S. Guo, and Y. Dai, "Event-triggered multi-lane fusion control for 2-D vehicle platoon systems with distance constraints," *IEEE Trans. Intell. Veh.*, early access, Oct. 17, 2022, doi: 10.1109/TIV.2022.3215133.

[45] Y. Guo and J. Ma, "Scopto: Signalized corridor management with vehicle platooning and trajectory control under connected and automated traffic environment," *Transportmetrica B: Transport Dyn.*, vol. 9, no. 1, pp. 673–692, 2021.

[46] V. Sokolov, J. Larson, T. Munson, J. Auld, and D. Karbowski, "Maximization of platoon formation through centralized routing and departure time coordination," *Transp. Res. Rec.*, vol. 2667, no. 1, pp. 10–16, 2017.

[47] X. Cai et al., "Analyzing infrastructure lidar placement with realistic lidar," 2022, *arXiv:2211.15975*.

[48] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang, "Learning distilled collaboration graph for multi-agent perception," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 29541–29552.

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

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

[51] Y. Yuan, H. Cheng, and M. Sester, "Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3054–3061, Apr. 2022.

[52] Z. Bai, G. Wu, M. J. Barth, Y. Liu, E. A. Sisbot, and K. Oguchi, "Vinet: Lightweight, scalable, and heterogeneous cooperative perception for 3 d object detection," 2022, *arXiv:2212.07060*.

[53] Y. Li, J. Zhang, D. Ma, Y. Wang, and C. Feng, "Multi-robot scene completion: Towards task-agnostic collaborative perception," in *Proc. 6th Annu. Conf. Robot Learn.*, 2022.

[54] R. Xu, "Opencood main repository," 2022. [Online]. Available: https://github.com/DerrickXuNu/OpenCOOD

[55] R. Xu, Z. Tu, H. Xiang, W. Shao, B. Zhou, and J. Ma, "Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers," 2022, *arXiv:2207.02202*.

[56] K. Jiang et al., "Map container: A map-based framework for cooperative perception," 2022, *arXiv:2208.13226*.

[57] Y. Lu et al., "Robust collaborative 3 D object detection in presence of pose errors," 2022, *arXiv:2211.07214*.

[58] D. Qiao and F. Zulkernine, "Adaptive feature fusion for cooperative perception using lidar point clouds," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 1186–1195.

[59] D. engine Contributors, "DI-engine: OpenDILab decision intelligence engine," 2021. [Online]. Available: https://github.com/opendilab/DI-engine

[60] Y. Chang and A. Mosleh, "Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents: Part 1: Overview of the IDAC model," *Rel. Eng. Syst. Saf.*, vol. 92, no. 8, pp. 997–1013, 2007.

[61] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[62] O. D. Team, "Openpcdet: An open-source toolbox for 3D object detection from point clouds," 2020. [Online]. Available: https://github.com/open-mmlab/OpenPCDet

[63] M. Contributors, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," 2020. [Online]. Available: https://github.com/open-mmlab/mmdetection3d

[64] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4106–4115.

[65] E. Arnold, S. Mozaffari, and M. Dianati, "Fast and robust registration of partially overlapping point clouds," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1502–1509, Apr. 2022.

[66] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.

[67] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12697–12705.

[68] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3D object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7652–7660.

[69] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018, Art. no. 3337.

[70] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-Cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds," in *Proc. 4th IEEE/ACM Symp. Edge Comput.*, 2019, pp. 88–100.

[71] C. K. Chui et al., *Kalman Filtering*. Berlin, Germany: Springer, 2017.

[72] G. A. Einicke and L. B. White, "Robust extended Kalman filtering," *IEEE Trans. Signal Process.*, vol. 47, no. 9, pp. 2596–2599, Sep. 1999.

[73] M. A. Johnson and M. H. Moradi, *PID Control*. Berlin, Germany: Springer, 2005.

[74] A. Afram and F. Janabi-Sharifi, "Theory and applications of HVAC control systems–a review of model predictive control (MPC)," *Building Environ.*, vol. 72, pp. 343–355, 2014.

[75] X. Xia et al., "Automated driving systems data acquisition and processing platform," 2022, *arXiv:2211.13425*.

[76] H. Xiang, R. Xu, X. Xia, Z. Zheng, B. Zhou, and J. Ma, "V2XP-ASG: Generating adversarial scenes for vehicle-to-everything perception," in *Proc. Int. Conf. Robot. Automat.*, 2023.

[77] G. Jocher et al., "Ultralytics/yolov5: V6.2 - YOLOv5 classification models, apple M1, reproducibility, ClearML and deci.ai integrations," Aug. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7002879

[78] W. Liu, K. Quijano, and M. M. Crawford, "YOLOv5-tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 8085–8094, 2022.

[79] O. Cordon, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* (Advances in Fuzzy Systems-Applications and Theory), vol. 19. Singapore: World Sci., 2001.

[80] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13750–13759.

Runsheng Xu received the B.S. degree in electrical engineering from North China Electrical Power University, Beijing, China, in 2016, and the M.S. degree in electrical engineering from Northwestern University, Evanston, IL, USA, in 2017. He is currently working toward the Ph.D. degree with the UCLA Mobility Lab, University of California, Los Angeles, Los Angeles, CA, USA. His research interests include computer vision, autonomous driving perception system, and cooperative driving automation.

Hao Xiang received the B.S. degree in microelectronics from Fudan University, Shanghai, China, in 2018, and the M.S. degree in machine learning and data science from the University of California, San Diego, La Jolla, CA, USA. He is currently working toward the Ph.D. degree with UCLA Mobility Lab, University of California, Los Angeles, Los Angeles, CA, USA. His research interests include computer vision, multi-agent learning, multi-modal learning, trajectory prediction, robotics, and autonomous driving.

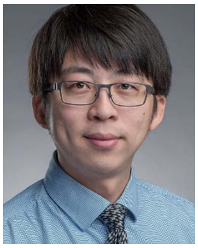
Xu Han received the B.S. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, USA, in 2016, and the M.S. degree in electrical engineering from Washington University in St.Louis, St. Louis, MO, USA. He is currently working toward the Ph.D. degree with UCLA Mobility Lab, University of California, Los Angeles, Los Angeles, CA, USA, advised by Prof. Jiaqi Ma. His research interests include V2X cooperation, V2V cooperation, and high-level behavior planning for autonomous vehicles.

Xin Xia received the B.E. degree in vehicle engineering from the School of Mechanical and Automotive Studies, South China University of Technology, Guangzhou, China, in 2014, and the Ph.D. degree in vehicle engineering from the School of Automotive Studies, Tongji University, Shanghai, China, in 2019. From January 2020 to March 2021, he was a Postdoctoral Fellow Associated with Dr. Amir Khajepour with the Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently an Assistant Project Scientist with the Department of Civil and Environmental Engineering, University of California, Los Angeles, Los Angeles, CA, USA. His research interests include state estimation, cooperative localization, cooperative perception, and dynamics control of the autonomous vehicle.

Zonglin Meng received the B.S. degree in computer science from the University of Wisconsin Madison, Madison, WI, USA, in 2019. He is currently working toward the Ph.D. degree with UCLA Mobility Lab, University of California, Los Angelos, Los Angelos, CA, USA. His research interests include computer vision, detection, tracking, and HD mapping.

Chia-Ju Chen received the B.S. degree in electrical and computer engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2017, and the M.S. degree from The University of Texas at Austin, Austin, TX, USA, in 2019. She is currently working toward the Ph.D. degree with UCLA Mobility Lab. Her research interests include privacy, computer vision, and autonomous driving.

Camila Correa-Jullian received the M.S. Reliability Engineering degree in risk analysis for automated system from the University of Maryland, College Park, MD, USA, in 2021, and the B.S. Mechanical Engineering degree with the University of Chile, Santiago, Chile, in 2019. She is currently working toward the Ph.D. degree in mechanical and aerospace engineering with the University of California, Los Angeles, Los Angeles, CA, USA. Her research with the B. John Garrick Institute for the risk sciences is focuses on studying the operational risks of autonomous vehicles and the role of human-system interactions in automated system safety.



Jiaqi Ma (Member, IEEE) received the Ph.D. degree in transportation engineering from the University of Virginia, Charlottesville, VA, USA, in 2014. He is currently an Associate Professor with the UCLA Samueli School of Engineering and a Faculty Lead with the New Mobility Program, UCLA Institute of Transportation Studies, Los Angeles, CA, USA. His research interests include intelligent transportation systems, autonomous driving, and cooperative driving automation. He is a Member of the TRB Standing Committee on Vehicle-Highway Automation, TRB Standing Committee on Artificial Intelligence and Advanced Computing Applications, and American Society of Civil Engineers Connected and Autonomous Vehicles Impacts Committee, and the Co-Chair of the IEEE ITS Society Technical Committee on Smart Mobility and Transportation 5.0. He is the Editor in Chief of the IEEE Open Journal of Intelligent Transportation Systems.