Cyber Mobility Mirror for Enabling Cooperative Driving Automation in Mixed Traffic

A Co-Simulation Platform

Zhengwei Bai* and Guoyuan Wu
Are with the Department of Electrical and Computer Engineering, University of California, Riverside, Riverside, CA 92507 USA. E-mail: zbai012@ucr.edu; gywu@cert.ucr.edu

Xuewei Qi, Yongkang Liu, and Kentaro Oguchi
Are with the Toyota North America R&D Labs, Mountain View, CA 94043 USA. E-mail: qixuewei@gmail.com; yongkang.liu@toyota.com; kentaro.oguchi@toyota.com

Matthew J. Barth
Is with the Department of Electrical and Computer Engineering, University of California, Riverside, Riverside, CA 92507 USA. E-mail: barth@ucr.edu

*Corresponding author
Abstract—Endowed with automation and connectivity, connected and automated vehicles (CAVs) will be a revolutionary promoter for cooperative driving automation (CDA). Nevertheless, CAVs need high-fidelity perception information on their surroundings, which is available but costly to collect from various onboard sensors, as well as vehicle-to-everything communications. Therefore, authentic perception (AP) information based on high-fidelity sensors via a cost-effective platform is crucial for enabling CDA-related research, e.g., cooperative decision making or control. Most state-of-the-art traffic simulation studies for CAVs rely on the situation-awareness information by directly calling on intrinsic attributes of the objects, which impedes the reliability and fidelity of the assessment of CDA algorithms. In this study, a cyber mobility mirror (CMM) co-simulation platform is designed for enabling CDA by providing AP information. The CMM co-simulation platform can emulate the real world with a high-fidelity sensor perception system and the cyberworld with a real-time rebuilding system acting as a “mirror” of the real-world environment. Concretely, the real-world simulator is mainly in charge of simulating the traffic environment, the sensors, and the AP process. The mirror-world simulator is responsible for representing objects and providing their information as intrinsic attributes of the simulator to support the development and evaluation of CDA algorithms. To illustrate the functionality of the proposed co-simulation platform, a roadside lidar-based vehicle perception system for enabling CDA is prototyped as a study case. Specific traffic environments and CDA tasks are designed for experiments whose results are demonstrated and analyzed to show the performance of the platform.

With the rapid development of the economy and society, the field of transportation is facing several major challenges caused by drastically increased traffic demand, such as improving traffic safety, mitigating traffic congestion, and reducing mobile source emissions. Cooperative driving automation (CDA) enabled by connected and automated vehicles (CAVs) is a promising solution to the aforementioned challenges [1]. In the past few decades, several projects have been conducted to explore the potential of CDA. The California Partners for Advanced Transit and Highways (PATH) program [2] demonstrated the improvement of traffic throughput by an automated platoon utilizing connectivity. The European DRIVE C2X project [5] assessed the cooperative system by large-scale field operational tests of various connected vehicle applications. Fujitsu has launched a digital twin platform for supporting mobility by connectivity and artificial intelligence [4]. These projects have demonstrated that CDA can be a transformative pathway toward the next-generation transportation system, which is enabled by ubiquitous perception, seamless communication, and advanced artificial intelligence technologies.

Introduction
Given the fact that the cost of large-scale real-world deployment is prohibitive, it is imperative to design and assess CDA systems based on simulation. For instance, as one key component of a CDA system, CAV technologies heavily rely on simulation to comprehensively assess their performance in terms of safety, efficiency, and environmental sustainability. Therefore, simulation platforms for CDA are of great significance, and their development receives much attention.

Additionally, to enable CDA, accurate perception information can lay a solid foundation, which requires inputs from different types of high-fidelity sensors, such as radar, camera, and lidar. The direct implementation of these sensors to perceive the real-world environment may be costly or time-consuming and, in some cases, restricted by application scenarios. Thus, simulation platforms with high-fidelity sensor modeling and perception capability would provide a cost-effective alternative solution to CDA-related research.

Many existing traffic simulators have been developed to test various aspects of CDA. For instance, CARLA [5] and SVL [6] are designed for modeling autonomous vehicles, while Simulation of Urban Mobility (SUMO) [7] targets microscopic traffic flow. From the perspective of sensing fidelity, most existing studies directly use intrinsic attributes of target objects without considering the potential imperfection of perception for CDA models. This considerably limits the transferability and reliability of the real-world deployment of these CDA models. To the best of our knowledge, this article is the first attempt to integrate the entire deep learning-based perception pipeline into the simulator to create a cyber mobility mirror (CMM) system in which simulated traffic objects are authentically perceived and reconstructed with a 3D representation. The high-level CDA model can leverage such perceived information from the module output in a CMM system rather than intrinsic attributes of target objects to improve the model fidelity and validate the system with more confidence.
The systematic structure of CMM based on a co-simulation platform is shown in Figure 1, where one simulator is designed to model the real-world traffic environment and the authentic perception (AP) pipeline, while the other simulator is used to work as a mobility mirror, i.e., reconstructing the perceived objects and presenting them. The output interface provides readily retrieved postperception data for CDA applications.

The main contributions of the article can be summarized as follows:

- This article proposes a CMM architecture for enabling CDA research and development.
- A prototype CMM system is designed and developed using a CARLA-based co-simulation platform.
- A CARLA-based 3D object detection training dataset is presented.
- Based on the CMM co-simulation platform, a case study is conducted for roadside lidar-based vehicle detection to demonstrate the functionality and necessity of the proposed co-simulation platform for enabling CDA algorithm development and validation.

The rest of this article is organized as follows. A brief background about traffic simulation and object perception is given in the “Background” section. In the section “Platform Structure and Design,” we first introduce the concept of CMM and then describe the design and development of the prototype system based on co-simulation. In the “Case Study: Roadside Lidar-Based Vehicle Detection for Enabling CDA” section, we present a case study for detecting and reconstructing vehicles based on roadside lidar sensing and deep learning methods, followed by the “Conclusion and Discussion” section.

**Background**

Simulation plays a crucial role in enabling CDA, such as the assessment of CAV cooperative perception algorithms and decision-making/control models [8], [9]. High-fidelity simulated sensor information lays a solid foundation for these high-level CDA algorithms and models. In this section, we briefly review the background information for simulators enabling CDA and object detection.

**Simulators Enabling CDA**

**Microscopic Traffic Simulators**

To model the evolution of traffic states based on traffic dynamics and interactions among traffic objects, microscopic traffic simulators have been developed for decades and have greatly stimulated the development of intelligent transportation systems [10]. These simulators mainly consist of three components: 1) a transportation network defining road topology; 2) a traffic generator creating traffic flows with certain demand distributions; and 3) microscopic traffic flow control strategies, including traffic

---

FIG 1 (a) The systematic architecture for the proposed CMM co-simulation platform and the workflow for supporting CDA. Three sample figures, including (b) the real traffic environment, (c) perception results visualization, and (d) cyberenvironment, are shown for illustrating the main goal of the platform. CCAC: cooperative adaptive cruise control; EAD: eco-approach and departure; CP: cooperative perception.
The development of DNNs has significantly improved the possibility of dealing with large-scale data, such as high-fidelity images or 3D point clouds.

signal management, vehicle driving behaviors, and moving strategies for pedestrians.

Several simulation platforms are of great popularity in CDA research, such as VISSIM [11], Aimsun [12], and SUMO [15]. Specifically, VISSIM and Aimsun are widely used in dealing with multimodal traffic flow simulation due to their capabilities of providing fundamental 3D preview and statistical simulation results. SUMO is an open source, highly portable, microscopic, and continuous traffic simulation package designed to handle large networks [7]. Additionally, high compatibility to connect and interact with different kinds of external simulators, e.g., OMNeT++ [14], CARLA, etc., is one of the key features of SUMO. These microscopic traffic simulators mainly focus on general assessment for traffic dynamic performance at the network level under different traffic scenarios. Nevertheless, the design and assessment of CDA-based cooperative perception, decision-making, or control models highly rely on the fidelity of sensor data, which is a major challenge for these conventional simulators. In recent years, simulators that are capable of modeling high-fidelity sensor gain have gained more and more interest, and these are introduced in the following section.

Autonomous Driving Simulators
With the development of CDA, especially autonomous driving technologies [15], the requirement for high-fidelity sensors in simulators has gained more and more attention. In recent years, several autonomous driving simulators equipped with high-fidelity sensors have been developed based on game engines, such as Unity [16] and Unreal Engine [17]. For instance, AirSim [18], SVL, and CARLA have the capability to offer physically and visually realistic simulations for autonomous vehicle technologies (AVTs) as well as CDA systems. Specifically, AirSim includes a physics engine that can operate at a high frequency for real-time hardware-in-the-loop simulations with support for popular protocols, such as MAVLink [19]. SVL is a high-fidelity simulator for AVTs that provides end-to-end and full-stack simulation that is ready to be hooked up with several open source autonomous driving stacks, such as Autoware [20] and Apollo [21]. CARLA, an open source simulator for autonomous driving, supports flexible specifications of sensor suites and environmental conditions. In addition to open source codes and protocols, CARLA provides open digital assets (e.g., urban layouts, buildings, and vehicles) that can be used in a friendly manner for researchers.

These simulators have been developed from the ground up to support the development, training, and validation of AVTs, enabling the development of CDA. They have the capacity to assess the CDA system in a cost-effective manner as well as to provide high-fidelity sensing information.

Although they have these existing simulators, researchers still struggle with the imperative assumption that perception data (e.g., the location, velocity, rotation, etc.) is collected directly from the intrinsic attributes of simulation engines when they develop and evaluate their high-level CAV functions, such as decision-making or control methods for CDA. Therefore, developing a generic platform that can not only support physically and visually realistic simulation but also provide perception data based on high-fidelity sensor information is still a research gap for enabling CDA system research and development.

Object Detection in Traffic Scenes
Object detection plays a crucial and fundamental role in enabling CDA, and it can be roughly divided into two major types: 1) traditional model-based algorithms and 2) data-driven methods based on deep neural networks (DNNs) to extract hidden features from input signals and then generate detected results [22]. The following section briefly introduces several state-of-the-art methods for each type, and high-resolution sensors, e.g., camera and lidar, are mainly focused on.

Model-Based Methods
At the early stage of traffic detection, sensors requiring low computational power were widely used, such as loop detectors, radar, ultrasonic, etc. Although most of them are still implemented in contemporary transportation systems, they suffer from different kinds of intrinsic problems, such as detecting uncertainties, traffic disruption at installation, and high maintenance costs [23]. With the development of computer vision (CV) technology and the improvement of computational power, camera-based traffic object detection has been widely developed and adapted. Aslani and Mahdavi-Nasab [24] tried to gather useful information from stationary cameras for detecting moving objects in digital videos based on optical flow estimation. In situations where limited memory and computing resources are available, Lee et al. [25] presented a moving object detection method for real-time traffic surveillance applications based on a genetic algorithm.

Recently, lidar sensors have been increasingly implemented for traffic object detection tasks due to their advantages
of having a higher tolerance for lighting conditions and better accuracy of relative distance. Regarding traditional methods to deal with the 5D point cloud, one popular workflow is 1) background filtering, 2) traffic object clustering, and 3) object classification [26]. Additionally, the lidar point cloud can also be used to identify lane markings [27], [28]. Although some traditional methods have been applied to the 5D point cloud, the greater potential of lidar data should be tapped by data-driven methods (e.g., deep learning), which are described next.

Data-Driven Methods
The development of DNNs has significantly improved the possibility of dealing with large-scale data, such as high-fidelity images or 5D point clouds. With the great success of deep learning in the image recognition area [29], [30], many DNN-based models have been implemented in object detection for traffic scenarios using cameras or lidar sensors. Chabot et al. [31] presented an approach called Deep MANTA for multitask vehicle analysis based on the monocular image. In terms of different lighting conditions, Lin et al. [52] developed a nighttime vehicle detection method based on image style transfer. Chen et al. [55] proposed a shallow model named concatenated feature pyramid network to detect smaller objects in traffic flow from fish-eye camera image.

Additionally, 5D lidar is also getting more popular in traffic object detection [54]. For instance, Asvadi et al. [55] presented an algorithm named DepthCN, which used deep convolutional neural networks (CNNs) for vehicle detection. Considering the real-time requirement for autonomous driving applications, Zeng et al. [56] proposed a real-time 5D object detection method by utilizing pre-RoI-pooling convolution and pose-sensitive feature maps. Simony et al. [37] proposed an Euler region proposal for real-time 5D object detection with point clouds, called ComplexYolo, which is capable of generating rotated bounding boxes for 3D objects. For CDA applications, traffic object detection needs meticulous consideration. Thus, in our CMM co-simulation framework, the ComplexYolo model is adapted with customized improvements for real-time vehicle and pedestrian detection.

Platform Structure and Design
This section describes the concept of CMM in detail and the system architecture of the co-simulation platform based on CARLA. Specifically, the design and development of the real-world simulator, the mirror simulator, the data communication module, and the AP module are introduced.

CMM can further tap the potential of traffic object surveillance systems to enable CDA, especially for routing planning, cooperative decision making, and motion control. As described earlier, CMM can further tap the potential of traffic object surveillance systems to enable CDA, especially for routing planning, cooperative decision making, and motion control. From this perspective, this article presents a co-simulation platform based on the CMM concept. Figure 2 demonstrates the concept of CMM with an intersection scenario and the system architecture of the CMM-based co-simulation.

In Figure 2(a), the upper left part represents the real-world traffic scenario at an intersection equipped with several roadside high-fidelity sensors, e.g., a camera and 3D lidar, roadside computing server, and communication system. High-fidelity sensor data are retrieved by the roadside server, in which perception tasks are executed, e.g., traffic object detection, classification, tracking, and motion prediction. Then, perception results are encoded and transmitted to the CMM server via communication networks, e.g., cellular, dedicated short-range communication (DSRC), or wireless local area network. The upper right part of Figure 2(b) represents the mirror environment, which reconstructs 3D objects based on perception information and outputs to CDA applications, e.g., collision avoidance, smart lane selection [58], ecodriving [59], etc.

As aforementioned, building a comprehensive traffic object perception system in the real world requires plenty of hardware and labor resources. In this article, we propose a cost-effective means to emulate the real-world traffic environment via a game engine-based simulator, CARLA, which has the capability of generating traffic environment and high-fidelity sensor information. The structure of the real-world simulator, demonstrated in Figure 2(c), consists of four modules: 1) the initialization of the system settings; 2) the configuration of the CARLA-based traffic environment with traffic objects and equipped sensors; 3) AP using DNNs, such as the ComplexYolo model; and 4) data encoding for communication. As shown in Figure 2(d), the mirror simulator is also developed based on CARLA and consists of three components: 1) decoding of the communicated data; 2) 3D rebuilding for vehicles and pedestrians; and 3) an output interface for CDA applications to readily retrieve the postperception data.
In this article, we present the basic concept of CMM and design the concrete workflow of a roadside sensor-based intersection surveillance scenario. In the following section about co-simulation design and development, we focus on 1) applying roadside 3D lidar for perception, 2) object detection and classification for vehicles and pedestrians, and 3) multi-vehicle 3D rebuilding for enabling CDA applications.

**Real-World Simulator Design**

The main purpose of the real-world simulator is to generate a virtual environment based on CARLA to emulate the real-world traffic environment. The main subtasks of the development effort are described as follows.

**Traffic Scenario Design**

CARLA has provided several well-developed virtual towns with different road maps and textures. In this article, we implement “town03” as our fundamental traffic map and select a target intersection for research. This is shown in Figure 3(a). Additionally, we can generate vehicles and pedestrians via CARLA-based Python scripts. Figure 5(b) shows a specific top-down view of the traffic scenario with vehicles, pedestrians, and a 3D lidar installed at the target intersection. The traffic is generated via the CARLA interface with respect to certain traffic demands, and the traffic signals are controlled by the built-in traffic signal manager in CARLA.

**Infrastructure-Based Sensor Design**

In this article, we implement a roadside 3D lidar as our main sensor. The roadside lidar is installed at the southwest corner of the target intersection, which is demonstrated in Figure 5 (the lidar is installed below the arm of the traffic signal pole). Specifically, detailed settings about the roadside 3D lidar are described in Table 1. To reduce concern about the transferability of the deployed DNN model, our approach resembles the lidar setups used to obtain the KITTI dataset [40]. This means that the pitch, yaw, and roll of the lidar are set as zeros in the CARLA global coordinate, as shown in Figure 3(b). Specifically, the lidar intensity is calculated by

\[ I = I_0 \cdot e^{-ad} \]  

where \( I_0 \) represents the initial intensity value (equal to one in this study); \( a \) represents the attenuation coefficient,
depending on the sensor’s wavelength and atmospheric conditions (which can be modified by the lidar attribute “atmosphere_attenuation_rate”); and $d$ is the distance from the hit point to the sensor. Furthermore, realistic lidar features, such as random no returns and background noise, are also considered [41]. More details about the implemented 3D lidar are described in the “Case Study: Roadside Lidar-Based Vehicle Detection for Enabling CDA” section.

Deep Learning-Based Perception Methods

In this article, we apply the ComplexYolo model [37] as our fundamental 3D object detection method. The basic pipeline of the ComplexYolo model is demonstrated in Figure 4.

In the ComplexYolo model, raw 3D lidar data are first cut into a certain shape with respect to the target region; then, the 3D point cloud is processed into a bird’s-eye view (BEV) red, green, blue (RGB) map based on different features. The CNN-based ComplexYolo network generates detection outputs based on the RGB map; moreover, postprocessing is implemented to filter the detection results with respect to certain thresholds. Finally, 3D bounding boxes are calculated and displayed on the RGB map image. The optimization loss function $L$ for ComplexYolo is defined as

$$ L = L_{\text{Yolo}} + L_{\text{Euler}} $$

where $L_{\text{Yolo}}$ is defined as the sum of squared errors using the multipart loss introduced in YOLO [42] and YOLOv2 [43], while the Euler regression part $L_{\text{Euler}}$ is defined to handle complex numbers and has a closed mathematical space for angle comparisons [37]. The implementation details of the ComplexYolo model in this article are introduced in the “Case Study: Roadside Lidar-Based Vehicle Detection for Enabling CDA” section.

Mirror Simulator Design

The main purposes of the mirror simulator are 1) to perform 3D rebuilding for the perceived objects and 2) to

Table 1. The parameter configuration and description for the 3D lidar sensor deployed in the case study.

| Parameters                      | Default | Description                      |
|---------------------------------|---------|----------------------------------|
| Channels                        | 64      | Number of lasers                 |
| Height                          | 1.73 m  | Height with respect to the road surface |
| Range                           | 100 m   | Maximum distance to measure/ray cast in meters |
| Rotation_frequency              | 10 Hz   | Lidar rotation frequency         |
| Upper_fov                       | 2       | Angle in degrees of the highest laser beam |
| Lower_fov                       | –24.9   | Angle in degrees of the lowest laser beam |
| Atmosphere_attenuation_rate     | 0.004   | Coefficient that measures the lidar intensity loss |
| Noise_stddev                    | 0.01    | Standard deviation of the noise model of points along its ray cast |
| Dropoff_rate                    | 45%     | General proportion of points that are randomly dropped |
| Dropoff_intensity_limit         | 0.8     | Threshold of the intensity value for exempting dropoff |
| Dropoff_zero_intensity          | 40%     | Probability value of dropoff for zero-intensity points |
The main purposes of the mirror simulator are 1) to perform 3D rebuilding for the perceived objects and 2) to provide a readily used output interface for CDA applications.

provide a readily used output interface for CDA applications. In this article, the mirror simulator is also developed based on the CARLA simulator. The same CARLA town, “town03,” is used to build this mobility mirror, as shown in Figure 3(a). The 3D rebuilding procedure consists of two parts: 1) decoding postperception data received from the real-world simulator via communication and 2) generating 3D traffic objects with respect to the decoding data based on CARLA Python application programming interfaces (APIs). Specifically, the message decoding method is further described in the next section.

Co-Simulation Workflow Design
In this article, the TCP protocol [44] is implemented to transmit postperception data from the real-world simulator to the mirror simulator. Specifically, before transmitting, the object detection results are encoded based on the JavaScript Object Notation (JSON) protocol [45], which includes the locations and orientations of the objects. For data encoding at the mirror simulator, the JSON data are decomposed according to the data structure in CARLA.

For this co-simulation platform, we design the information flow based on the server–client architecture in CARLA [5]. The sequence diagram for information synchronization among components is demonstrated in Figure 5.

In the CARLA platform, the server runs the simulation (i.e., updates the information), while the client retrieves information. Specifically, as shown in Figure 5, the traffic generator (CARLA client 1-1) is responsible for the simulation initialization or stopping requests, the real-world Simulator (CARLA server 1) runs the simulation for real-world traffic and the lidar sensor, and the Detection generator (CARLA client 1-2) is designed to generate 3D object detection results by utilizing the ComplexYolo model. The detection results are encoded into postperception data and transmitted to the mirror simulator (CARLA server 2) via TCP communications. Finally, the mirror output (CARLA client 2-1) can retrieve the mirrored objects’ information, e.g., vehicle center location, bounding box dimension, and orientation, from the mirror simulator.

Vehicle-to-Everything Effect Design
Since vehicle-to-everything (V2X) communication effects are significant in CDA implementations, this section introduces the V2X effect design from three different perspectives.

Communication Delay
Owing to the co-simulation structure, a communication delay occurs in deploying real-world simulation and mirror simulation on different computers. Furthermore, to support the assessment for different delay conditions, an active communication delay (ACD) is designed at the detection generator shown in Figure 5.

The ACD aims to provide customized time delays that can be defined according to specific requirements. Hence, the total delay $T$ of the co-simulation is designed as follows:

$$T = T_{in} + T_{ACD}$$

where $T_{in}$ and $T_{ACD}$ represent the innate time delay of the co-simulation and the delay from the ACD, respectively. Specifically, for the wireless
communication applied in our case study shown in the “Case Study: Roadside Lidar-Based Vehicle Detection for Enabling CDA” section, $T_{in}$ is about 150 ms, and $T_{ACD} \sim \mathcal{N}(\mu, \sigma^2)$ is applied by setting $\mu, \sigma$ as 50 and five, respectively.

**Message Dropping**

Considering that message drops are a common issue in communications systems [46], an active message dropping (AMD) mechanism is designed in our co-simulation system. In the co-simulation system, the AMD is designed by the stochastic process and deployed right after the ACD module. The $R$ represents the factor for the message dropping, which is designed by

$$R \sim \mathcal{U}(0, 1). \quad (4)$$

For each frame, if $R$ is smaller than a certain threshold $\eta$, the message of this frame will be dropped. Specifically, for experiments in the “Functionality for Enabling CDA” section, $\eta$ is set as 0, 0.05, and 0.1, respectively.

**Case Study: Roadside Lidar-Based Vehicle Detection For Enabling CDA**

**System Setup**

Both the real-world simulator and mirror simulator run under the synchronous mode of CARLA, which means the server can update the simulator information at the same time step with the clients. In this article, the simulation frequency is set as 10 Hz. The network traffic demand is set to be 100 vehicles (driving around the town according to the default routing strategy and autopilot method in CARLA) [5]. For the 3D lidar implemented in this article, the attributes of the lidar sensor are shown in Table 1.

To implement the ComplexYolo model in our co-simulation platform, a well-defined 3D lidar dataset with the ground-truth label is required. Although CARLA provides a comprehensive Python API for data retrieving and object controlling, a built-in dataset generator is still missing. Therefore, based on the existing CARLA Python API and the KITTI dataset structure, we developed a CARLA-based 3D lidar dataset named the *CARTI dataset*. The code for generating the CARTI dataset is available at https://github.com/zwbai/carti_dataset. Although the lidar is running at 10 Hz, the data frame is recorded at 2 Hz to improve the diversity of the dataset, i.e., including more differences in certain frames of data. Specifically, a total of 11,000 frames of data are collected.

**Vehicle Detection**

**Point Cloud Preprocessing**

In this research, our target range for vehicle detection is defined as a 50 × 50-m area $\Omega$ with respect to the location of the lidar. The square size of the target range is due to the construction mechanism of the BEV feature map, which is shown in the “BEV Image Construction” section. The raw point cloud data can be described by

$$\mathcal{P} = \{ [x, y, z, \theta] | \{x, y, z\} \in \mathbb{R}^3, \theta \in [0.0, 1.0] \}. \quad (5)$$

**FIG 5** A sequence diagram for information synchronization among components.
To reduce the impact from 3D lidar points out of the target range, $\mathcal{P}$ is geofenced by

$$
\mathcal{P}_g = \{ [x, y, z]^T | x \in X, y \in Y, z \in Z \}
$$

(6)

where $\mathcal{P}_g$ represents the 3D point cloud data after geofencing, and $X$ and $Y$ are set as $[0 \text{ m}, 50 \text{ m}], [-25, 25]$ respectively. Considering the calibrated height of the roadside lidar to be 1.74 m, $Z$ is set as $[-2.74 \text{ m}, 1.56 \text{ m}]$. Additionally, ground-truth labels are collected according to the objects within the target range.

BEV Image Construction

The 3D points within the target range are then normalized by the following equation:

$$
\begin{bmatrix}
\hat{x} \\
\hat{y} \\
\hat{z}
\end{bmatrix} = 
\begin{bmatrix}
h_{\text{range}_x} & 0 & 0 \\
0 & h_{\text{range}_y} & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} + 
\begin{bmatrix}
0 \\
0.5w \\
0
\end{bmatrix}
$$

(7)

where range$_x$ and range$_y$ represent the range along the x-axis direction and y-axis direction, respectively, and $w$ and $h$ denote the weight and height of the BEV image, respectively. Specifically, $[x \ y \ z]^T$ and $[\hat{x} \ \hat{y} \ \hat{z}]^T$ represent the points in $\mathcal{P}_g$ and the points after normalization, respectively.

Then, the three feature maps, i.e., the R map, G map, and B map, can be defined according to the density, height, and intensity information, respectively:

$$
\begin{align*}
\mathcal{I}_r(S_i) &= \min (1.0, \log(N + 1)/64, N = |\mathcal{P}_i|) \\
\mathcal{I}_h(S_i) &= \max (\mathcal{P}_i \cdot [0, 0, 1]^T) \\
\mathcal{I}_g(S_i) &= \max (I(\mathcal{P}_i))
\end{align*}
$$

(8)

where $S_i$ represents a specific grid cell of the RGB map; $\mathcal{I}_r$, $\mathcal{I}_h$, and $\mathcal{I}_g$ represent three channels for the RGB map; $I$ represents the intensity of the lidar point; and $N$ describes the number of points mapped from $\mathcal{P}_i$ to $S_i$.

Model Training

Based on the CARTI dataset, we train the ComplexYoLo model from scratch via stochastic gradient descent with a weight decay of 0.0005 and momentum of 0.9. For the dataset preparation, we subdivide the training set with 80% for training and 20% for validation. The learning rate is set as 0.001 for initialization and gradually decreased along 1,000 training epochs. For regularization, we implement batch normalization. For activation functions, a leaky rectified linear activation function, defined as follows, is used except in the last layer of the CNN, where a linear activation function $f(x) = x$ is used:

$$
f(x) = \begin{cases} 
  x, & x > 0 \\
  0.1x, & \text{otherwise}
\end{cases}
$$

(9)

Functionality for AP

Quantitative Performance

For quantitative results and analysis, Figure 6 demonstrates the overall performance of the training results, including the training loss, precision, and recall. Specifically, the precision illustrates the proportion of true positive detection among all predicted detections (i.e., all of the bounding boxes generated from the model). The recall is a factor that measures the ability of the detector to find all of the relevant cases (i.e., all of the ground truths), which is the proportion of true positive detection among all ground truths (i.e., real vehicles and pedestrians). Additionally, the notations of “@50” and “@75” mean that these values are calculated based on the intersection of union (IoU) of 0.5 and 0.75, respectively.

The evaluation results of the testing dataset are demonstrated in Table 2. Specifically, the average precision is shown for each class along the different thresholds, and F1 is the harmonic mean for precision and recall, which is designed as

$$
F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
$$

(10)

It is notable that the detection results for vehicles are much better than those for pedestrians. A hypothesis is that fewer lidar points would be reflected from pedestrians.
than from vehicles and that pedestrians are more susceptible to occlusion due to their smaller sizes.

**Qualitative Performance**

For qualitative results and analysis, Figure 7 demonstrates the key pipeline for the co-simulation system. Figure 7(a) shows the top-down view of the “real-world” traffic environment, the AP results are visualized in Figure 7(b), and the traffic conditions in the “mirror world” are demonstrated by the Figure 7(c) top-down-view images. For detection visualization, vehicles and pedestrians are bounded with yellow and red boxes, respectively. In addition, the blue edges of the bounding boxes represent the forward direction of the objects.

Figure 7 validates the feasibility of our CMM-based co-simulation platform. Vehicles and pedestrians are detected via the roadside-based lidar, and the associated digital replica is reconstructed in the mobility mirror simulator. Due to the detection range and accuracy of the selected model (i.e., ComplexYolo), some objects will be missed. Nevertheless, our platform is generic and highly compatible with different detection models, and the results can be improved with advances in state-of-the-art detection methods. In general, Figure 7 validates the core concept of our CMM-based co-simulation platform, i.e., generating authentic detection results based on high-fidelity sensors and rebuilding the traffic objects for external CDA applications, which demonstrates the functionality and feasibility of the co-simulation platform.

**Functionality for Enabling CDA**

This section demonstrates two main aspects of the functionalities of our co-simulation platform in terms of enabling

---

**FIG 7** The object detection results: (a) a top-down view of the “real-world” traffic environment, (b) the visualization for the detection results, and (c) a top-down view of the “mirror world” traffic conditions.
CDA. As shown in Figure 8, a specific CDA application—infrastructure-assisted CACC—is designed in a mixed traffic scenario to illustrate: 1) why AP is important for supporting CDA algorithm development and evaluation and 2) how can the co-simulation platform supports the CDA applications.

In Figure 8, the ego vehicle (marked as the following vehicle (FV)), a CV without self-perception capacity, is trying to follow a leading vehicle (LV), a legacy vehicle without connectivity. The location and speed information of the LV are required for this scenario.

In this article, different perspectives are considered to make the assessment more comprehensive and realistic: 1) mobile occlusion, e.g., the LV in Figure 8 is occluded by the truck; 2) the time delay, and 3) the message drop. To be specific, the intelligent driver model is applied to act as a basic CACC method, and, when miss detection happens, two schemes are designed to generate continuous perception results: 1) there are no LVs and 2) the LV will keep its location as before. Since scheme 1 represents a progressive CACC style, scheme 2 represents a conservative CACC style. As demonstrated by Figure 9, ideal perception (IP) represents the ground-truth vehicle location, and AP and AP-Safe (AP-S) are the visualizations for the two schemes mentioned. Particularly, the time when detection of the LV is received is marked by a green “*” at the bottom of the figures, while a red “.” is marked for when miss detection happens. Based on the three perception results, vehicle trajectories are generated and shown in Figures 10–12. In Figure 10, FVs based on different perception results can keep a reasonable distance gap to the LV. In detail, the trajectory of AP-S has a closer fit with the trajectory of IP.

Although three perception schemes can support the FV generating feasible location trajectories, the speed and acceleration trajectories generated from AP (i.e., AP and AP-S) have significant differences from the trajectory generated by IP, which are shown in Figures 11 and 12. For velocity, the trajectories generated by AP and AP-S have many more speed fluctuations, which are mainly caused by 1) the miss detection caused by occlusion and 2) the wrong detection caused by the performance of the detection model itself, such as the detection at 18.4 s shown in Figure 9. Thus, it is quite difficult to model the deficiency of AP in a traditional way, such as the probabilistic model. For acceleration, a similar pattern can be found in Figure 12. The trajectories based on AP and AP-S have more fluctuations compared with the trajectory of IP.

Furthermore, it is notable that the fluctuations are highly correlated with the status of perception results. Specifically, AP will accelerate while the LV is missing, and AP-S tends to decelerate until the LV is detected again. These trajectory patterns illustrate that the perception process and results will significantly affect the performance of CACC in this case or similar to other CDA applications, which further demonstrates the necessity to take AP into consideration when designing subsequent CDA algorithms.

For the functionality of our platform to support the V2X effect, the time delay and message drop are considered in the “V2X Effect Design” section, and experiments are conducted accordingly in this section. Figure 15 shows the velocity trajectories based on IP, AP without delay, AP
with a 100-ms delay, and AP with a 200-ms delay. Figure 14 shows the velocity trajectories based on IP, AP without message drop, AP with a 5% message drop, and AP with a 10% message drop. The results from Figures 15 and 14 demonstrate that our platform can support the analysis of the V2X effect, which plays a crucial role in CDA implementation.

Conclusion and Discussion
To enable CDA, simulators are required to comprehensively support the design and assessment of various applications. Moreover, AP based on the high-resolution sensors is of great significance for CDA development. To the best of the authors’ knowledge, this article is the first attempt to design and develop a co-simulation platform to prove the CMM concept, which can both emulate high-resolution sensors and provide readily retrieved perception information. Specifically, the co-simulation platform consists of two main subsimulators: 1) a real-world simulator for emulating the real-world traffic environment and (roadside) sensors and generating the AP data and 2) a mirror simulator for the 3D reconstruction of traffic objects and providing a readily retrieved interface for downstream CDA applications to access the location and orientation information of target traffic objects. A case study is conducted for roadside lidar-based vehicle detection in an intersection scenario, which demonstrates the performance of AP as well as the functionality and feasibility of the co-simulation platform for enabling CDA.

In this article, we develop a preliminary framework for CMM and validate it with simulation. A natural future step would be to realize the system in the real world, but there are some open issues deserving further exploration. For instance, we need to overcome disparities in the features between the sensor data from simulators and those in reality. We need to investigate the model transferability issue, i.e., to design a model that can be trained on simulation and implemented in
real-world scenarios without the necessity for or much effort in retraining the model or fine-tuning parameters.

Acknowledgment
This research was funded by the Toyota Motor North America InfoTech Labs. The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views of Toyota Motor North America.

About the Authors
Zhengwei Bai (zbai012@ucr.edu) earned his M.S. degree from Beijing Jiaotong University, Beijing, China, in 2020. He is a Ph.D. candidate in electrical and computer engineering at the University of California, Riverside, Riverside, CA 92507 USA. His research interests include object detection and tracking, cooperative perception, decision making, motion planning, and cooperative driving automation. He serves as a review editor in Urban Transportation Systems and Mobility. He is a Student Member of IEEE and the American Society of Civil Engineers Artificial Intelligence in Transportation Committee.

Guoyuan Wu (gywu@cert.ucr.edu) earned his Ph.D. degree in mechanical engineering from the University of California, Berkeley in 2010. He is an associate researcher and an associate adjunct professor at Bourns College of Engineering–Center for Environmental Research & Technology and the Department of Electrical and Computer Engineering at the University of California, Riverside, Riverside, CA 92507 USA. His research interests include the development and evaluation of sustainable and intelligent transportation system technologies, including connected and automated transportation systems, shared mobility, transportation electrification, the optimization and control of vehicles, traffic simulation, and emissions measurement and modeling. He serves as an associate editor for several journals, including IEEE Transactions on Intelligent Transportation Systems, SAE International Journal of Connected and Automated Vehicles, and IEEE Open Journal of Intelligent Transportation Systems. He is also a member of the Vehicle-Highway Automation Standing Committee (ACP30) of the Transportation Research Board, a board member of the Chinese Institute of Engineers Southern California chapter, and a member of the Chinese Overseas Transportation Association. He is a recipient of the Vincent Bendix Automotive Electronics Engineering Award. He is a Senior Member of IEEE.

Xuewei Qi (qixuewei@gmail.com) earned his Ph.D. degree in electrical and computer engineering from the University of California, Riverside in 2016. He is a principal artificial intelligence researcher with Toyota North America Research Labs (Silicon Valley), Mountain View, CA 94043 USA. He was previously with General Motors as an artificial intelligence and machine learning research scientist. He also worked as a lead perception research engineer at Aeye.ai. His research interests include deep learning, autonomous vehicles, perception and sensor fusion, reinforcement learning, and decision making. He serves as a member of several standing committees of the Transportation Research Board. He is a Member of IEEE.

Yongkang Liu (yongkang.liu@toyota.com) earned his Ph.D. degree in electrical engineering from the University of Texas at Dallas in 2021. He is a research engineer at Toyota Motor North America, InfoTech Labs, Mountain View, CA 94043 USA. His research interests include in-vehicle systems and advancements in intelligent vehicle technologies.

Kentaro Oguchi (kentaro.oguchi@toyota.com) earned his M.S. degree in computer science from Nagoya University. He is a director at Toyota Motor North America, InfoTech Labs, Mountain View, CA 94043 USA, where his team is responsible for creating intelligent connected vehicle architecture that takes advantage of novel artificial intelligence technologies to provide real-time services to connected vehicles for smoother and efficient traffic, intelligent dynamic parking navigation, and vehicle guidance to avoid risks from anomalous drivers. His team also creates technologies to form a vehicular cloud using vehicle-to-everything technologies. Previously, he worked as a senior researcher at Toyota Central R&D Labs in Japan.

Matthew J. Barth (barth@ece.ucr.edu) earned his Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 1990. He is currently the Yeager Families Professor with the College of Engineering, University of California, Riverside, Riverside, CA 92507 USA. He also serves as the director of the Center for Environmental Research and Technology. His research interests include intelligent transportation systems and the environment, transportation/emissions modeling, vehicle activity analysis,
advanced navigation techniques, electric vehicle technology, and advanced sensing and control. He has been active in the IEEE Intelligent Transportation System Society (ITSS) for many years, serving as a senior editor for both IEEE Transactions on Intelligent Transportation Systems and IEEE Transactions on Intelligent Vehicles. He served as the ITSS president for 2014 and 2015 and is currently the ITSS vice president of education. He is a Fellow of IEEE.

References
[1] D. J. Fagnant and K. Kockelman, “Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations,” Transp. Res. A, Policy Pract., vol. 77, pp. 167–181, Jul. 2015, doi: 10.1016/j.tra.2015.04.003.
[2] J. A. Misener and S. E. Shladover, “Path investigations in vehicle-roadside cooperation and safety: A foundation for safety and vehicle-infrastructure integration research,” in Proc. 2006 IEEE Intell. Transp. Syst. Conf., pp. 9–16, doi: 10.1109/ITSC.2006.1706711.
[3] R. Stahlmann, A. Festag, A. Tomatis, I. Badusch, and F. Fischer, “Starting European field tests for car-2-x communication: The drive2x framework,” in 18th ITS World Congr. Exhib., 2011, p. 12.
[4] “Fujitsu future mobility accelerator digital twin analyzer.” Fujitsu. Accessed: Sep. 21, 2022. [Online]. Available: https://www.fujitsu.com/global/about/resouces/press-releases/202009-01.html.
[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in Proc. Conf. Robot. Learning, 2017, pp. 1–16.
[6] G. Rong et al., “LGSVL simulator: A high fidelity simulator for autonomous driving,” in Proc. 2020 IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC), pp. 1–6, doi: 10.1109/ITSC51022019.2949422.
[7] P. A. Lopez et al., “Microscopic traffic simulation using sumo,” in Proc. 2018 21st Int. Conf. Intell. Transp. Syst. (ITSC), pp. 2575–2582, doi: 10.1109/ITSC.2018.8569958.
[8] Z. Bai, G. Wu, M. J. Barth, Y. Liu, A. Sibot, and K. Oguchi, “PillarGrid: Deep learning-based cooperative perception for 3D object detection from onboard-roadside lidar,” 2022, arXiv:2203.06319.
[9] Z. Bai, W. Shangnang, B. Cai, and L. Chai, “Deep reinforcement learning based high-level driving behavior decision-making model in heterogeneous traffic,” in Proc. 2019 Chin. Control Conf. (CCC), pp. 8600–8603, doi: 10.2991/ChiCC.2018.8866095.
[10] J. Barceló et al., Fundamentals of Traffic Simulation, vol. 145. New York, NY, USA: Springer-Verlag, 2010.
[11] M. Tellfendorn and P. Vortisch, “Microscopic traffic flow simulator VIS-SIM,” in Fundamentals of Traffic Simulation, J. Barceló, Eds. New York, NY, USA: Springer, 2015.
[12] J. Barceló and J. Casas, “Dynamic network simulation with AIMSUM,” in Simulation Approaches in Transportation Analysis, R. Kitamura, and M. Kuwahara, Eds. New York, NY, USA: Springer-Verlag, 2005, pp. 57–98.
[13] M. Behrisch, L. Bieker, J. Erdmann, and D. Kraijzewicz, “Sumo-simulation of urban mobility: An overview,” in Proc. SIMUL 2011, 3rd Int. Conf. Adv. Syst. Simul., ThinkMind, 2011, pp. 25–28.
[14] A. Varga, “Omnet++,” in Modeling and Tools for Network Simulation, K. Wehrle, M. Günes, and J. Gross, Eds. New York, NY, USA: Springer-Verlag, 2010, pp. 55–59.
[15] B. R. Kain et al., “Deep reinforcement learning for autonomous driving: A survey,” IEEE Trans. Intell. Transp. Syst., vol. 25, no. 6, pp. 4809–4826, 2021, doi: 10.1109/TITS.2021.3054626.
[16] A. Juliani et al., “Unity: A general platform for intelligent agents,” 2018, arXiv:1809.02627.
[17] B. Karis and E. Games, “Real shading in unreal engine 4.” Unreal Engine. Accessed: Sep. 21, 2022. [Online]. Available: https://edu2.unrealengine.com/Resources/Files/2015SiggraphPresentations Notes-26915758.pdf.
[18] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-fidelity visual and physical simulation for autonomous vehicles,” in Field and Service Robotics, M. Hutter and R. Siegwart, Eds. New York, NY, USA: Springer-Verlag, 2018, pp. 621–655.
[19] A. Koubâa, A. Allouch, M. Alaajans, Y. Javed, A. Belghith, and M. Khalgui, “Micro air vehicle link (Malvink) in a nutshell: A survey,” IEEE Access, vol. 7, pp. 87,058–87,080, Jun. 2019, doi: 10.1109/ACCESS.2019.2924410.
[20] S. Kato et al., “Autoware on board: Enabling autonomous vehicles with embedded systems,” in Proc. 2018 ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPSS), pp. 287–296, doi: 10.1109/ICCPSS.2018.00055.
[21] F. Graf, “Apolo.” Evanston, IL, USA: Routledge, 2008.