

# Miniature Testbed for Validating Multi-Agent Cooperative Autonomous Driving

Hyunchul Bae<sup>†</sup>, Eunjae Lee<sup>†</sup>, Jehyeop Han<sup>†</sup>, Minhee Kang,  
 Jaehyeon Kim, Junggeun Seo, Minkyun Noh and Heejin Ahn<sup>\*</sup>

**Abstract**—Cooperative autonomous driving, which extends vehicle autonomy by enabling real-time collaboration between vehicles and smart roadside infrastructure, remains a challenging yet essential problem. However, none of the existing testbeds employ smart infrastructure equipped with sensing, edge computing, and communication capabilities. To address this gap, we design and implement a 1:15-scale miniature testbed, CIVAT, for validating cooperative autonomous driving, consisting of a scaled urban map, autonomous vehicles with onboard sensors, and smart infrastructure. The proposed testbed integrates V2V and V2I communication with the publish-subscribe pattern through a shared Wi-Fi and ROS2 framework, enabling information exchange between vehicles and infrastructure to realize cooperative driving functionality. As a case study, we validate the system through infrastructure-based perception and intersection management experiments.

## I. INTRODUCTION

Autonomous driving technologies have advanced significantly over the past decades, and their widespread deployment is now imminent. However, most progress has focused on stand-alone vehicles, whereas cooperative driving—where vehicles and roadside infrastructure share information and coordinate their actions—remains a challenging but essential problem. The deployment of cooperative driving requires rigorous validation in realistic settings, but such validation continues to face substantial barriers in practice.

To enable such validation, several full-scale testbeds have been established, including M-City [25], K-City [12], and AstaZero [17]. These facilities provide urban-like environments and diverse traffic scenarios, and are equipped with Vehicle-to-Everything (V2X) infrastructure for connected and automated vehicle testing. While they improve the prospects of commercialization, they also demand extensive space and massive investment, creating a significant entry barrier. For cooperative driving, realistic validation further requires smart infrastructure together with a sufficient fleet of autonomous vehicles to capture multi-agent interactions, which are often prohibitively costly and logistically demanding. As a result, researchers often turn to simulation environments, such as CARLA [6], which provide safer and more cost-effective setups. However, simulations inherently struggle to replicate the

TABLE I: Scaled vehicle testing platforms

Testbed	Vehicle Sensor	Smart Infra	Motion Capture	V2V Comm. <sup>1</sup>	V2I Comm. <sup>2</sup>
Duckietown [16]	✓	-	-	-	-
UDSSC [22]	✓	-	-	-	-
Cambridge [9]	-	-	✓	-	✓
Go-CHART [10]	✓	-	-	-	✓
CPM Lab [11]	✓	-	✓	-	✓
ICAT [23]	✓	-	-	✓	✓
MCCT [5]	✓	-	✓	-	✓
CIVAT	✓	✓	✓	✓	✓

<sup>1</sup> Direct inter-vehicle communication.

<sup>2</sup> Communication between vehicles and external agents.

physical realities of vehicle dynamics, sensor characteristics, and communication issues.

To bridge the gap between full-scale testbeds and simulations, miniature testbeds have emerged as promising alternatives. These platforms enable controlled experiments at reduced cost and with lower entry barriers. Moreover, prior studies [2] derived scaling laws based on Buckingham’s  $\pi$  theorem and experimentally validated that scaled vehicles exhibit behavior similar to full-scale vehicles. Tab. I summarizes representative miniature testbeds, comparing them in terms of vehicle sensing, smart infrastructure (infrastructure with sensing, computation, and communication capabilities), reliance on motion capture systems, and supported interaction types (Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I)). Most existing platforms rely on onboard vehicle sensors and motion capture systems, while a few explicitly incorporate infrastructure-based sensing or V2I communication. ICAT [23] and MCCT [5] introduce infrastructure nodes with V2I communication, but still depend on motion capture systems and infrastructure without sensing capabilities.

In this paper, we propose a novel 1:15-scale miniature testbed, CIVAT (**Cooperative Intelligent V2X Autonomous Testbed**), designed for the validation of cooperative autonomous driving technologies. The platform integrates miniature vehicles equipped with onboard sensors and smart infrastructure supported by 3D LiDAR and a powerful edge computing module, thereby enabling advanced perception and decision-making. Recognizing that traffic environments will, for the foreseeable future, consist of a mix of connected autonomous vehicles (CAVs) and non-connected vehicles, our testbed supports both fully CAV and such mixed-traffic scenarios, providing a versatile platform for validating cooperative driving algorithms.

The contributions of this study are as follows:

<sup>†</sup> These authors contributed equally to this work.

<sup>\*</sup> Corresponding author.

Hyunchul Bae, Eunjae Lee, Jehyeop Han, Minhee Kang, and Heejin Ahn are with the School of Electrical Engineering, and Jaehyeon Kim, Junggeun Seo, and Minkyun Noh are with the School of Mechanical Engineering at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, 34141, Republic of Korea. {bhc2675, eunjae.lee, jehyeophan, ministop, kjh714, sjg12044, minkyun.noh, heejin.ahn}@kaist.ac.kr



Fig. 1: Main components of CIVAT

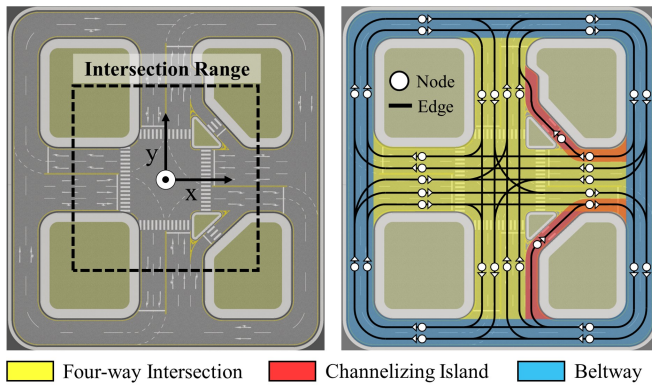


Fig. 2: Road types and segments

- We design a 1:15-scale testbed for validating cooperative autonomous driving technologies, incorporating both vehicles and smart infrastructure.
- We develop miniature vehicles equipped with onboard sensor suites and smart infrastructure equipped with 3D LiDAR and computation devices for perception, computation, and communication.
- We present an intersection management case study using smart infrastructure’s detection with 3D LiDAR and V2I communication to coordinate autonomous vehicles under mixed-traffic conditions.

## II. TESTBED DESIGN

Our testbed, CIVAT, is implemented by 1:15 scaling down a real-world four-way intersection. As shown in Fig. 1, the main components of CIVAT include the map, vehicles, and smart infrastructure. We detail the design of each component and communication in this section.

### A. Map Design

We design the map shown in Fig. 2 to capture the complexity of urban traffic and the dynamic interactions between vehicles. We construct it at a 1:15 scale, which yields a physical size of  $6\text{ m} \times 5.5\text{ m}$  corresponding to a real-world scale of  $90\text{ m} \times 82.5\text{ m}$ . At the center, we implement an unsignalized four-way intersection with 2–3 lanes per approach (yellow region in Fig. 2) and introduce channelizing islands (red regions in Fig. 2) on the approach and departure

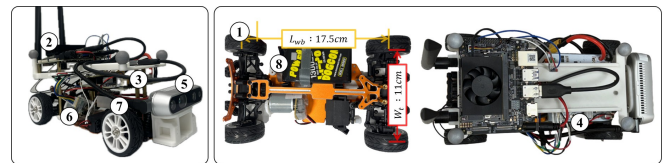


Fig. 3: Vehicle components

TABLE II: Hardware components of vehicle

Component	Item	Price (US\$)
1. Chassis & Servo motor	ZD RACING S16 4WD	117
2. Single-board Computer	Jetson ORIN NX 16GB	899
3. Custom PCB	Regulator, Motor Driver	20
4. Microcontroller Unit	NUCLEO-G431KB	13
5. Depth Camera	RealSense D435	334
6. Encoder DC Motor	M25N-2R-14 12V	18
7. IMU	myAHRS+	76
8. LiPo Battery	Vega 22.2V 1300mAh	30

The number in front of each component corresponds to Fig. 3.

lanes to induce realistic merging and diverging behaviors. To support continuous vehicle operation, we add a 2-lane beltway (blue regions in Fig. 2) that encircles the intersection and serves as a highway for high-speed driving. We model the road network as a graph, where each edge (black lines in Fig. 2) corresponds to a path segment. We store the waypoints along every segment in the memory of all agents to enable waypoint following. To replicate the frictional characteristics of asphalt, we print the map surface with material rated at 60 BPN (British Pendulum Number).

### B. Vehicle Design

1) *Hardware Components*: Fig. 3 represents the overall hardware composition of the vehicle. The vehicle is based on a 1:15-scale chassis<sup>1</sup> with Ackermann steering. It has a wheelbase of 17.5cm and a wheel track of 11cm, and its overall dimensions are  $15\text{ cm} \times 30\text{ cm} \times 13\text{ cm}$ . The onboard system includes the Jetson Orin NX for computation with GPU capabilities, equipped with an external Wi-Fi antenna for V2X communication. It also incorporates a NUCLEO Microcontroller Unit (MCU) for motor velocity control, a RealSense D435 camera for perception, a myAHRS+ IMU

<sup>1</sup>Available at [https://www.alibaba.com/product-detail/ZD-Racing-1-16-2-4G\\_1600493296884.html](https://www.alibaba.com/product-detail/ZD-Racing-1-16-2-4G_1600493296884.html)

for heading angle estimation, and a custom PCB with a power regulator and motor driver. The vehicle is driven by a DC motor with an encoder and is powered by a 1300 mAh LiPo battery. The total components are summarized in Tab. II, and the total vehicle cost amounts to \$1507.

Fig. 4 provides an overview of the vehicle system architecture. A single 22.4 V battery supplies the main power, which is regulated into two 12 V lines and one 5 V line via the custom PCB. One 12 V line powers the Jetson Orin NX, while the other supplies the motor driver. The 5 V line is used for the MCU (STM32 Nucleo), servo motor, and encoder. Regarding data flow, the Jetson Orin NX serves as the main processing unit on the vehicle. It receives global pose data from the motion capture system via Wi-Fi and sensor measurements from the camera and IMU through serial communication. The Jetson Orin NX sends velocity commands to the MCU via the I2C interface. The MCU converts these commands into PWM signals, which are forwarded to the motor driver to regulate the power supplied to the DC motor. The encoder simultaneously measures the motor rotation and returns pulse signals to the MCU for velocity estimation. In parallel, the Jetson Orin NX issues steering commands directly to the servo motor through PWM signals.

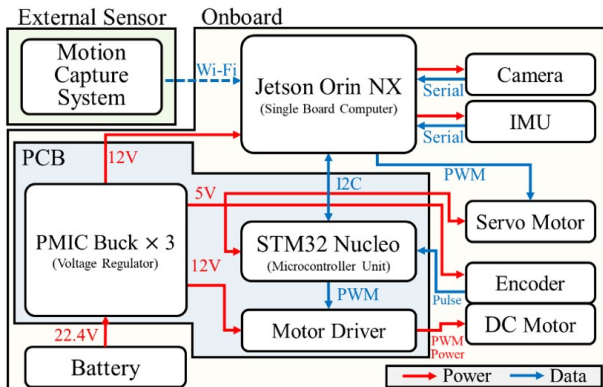


Fig. 4: Vehicle system structure

2) *Kinematic Model*: We model the vehicle with Ackermann steering using the bicycle model [20], which captures its essential kinematic behavior under the no-slip assumption:

$$\begin{aligned} \dot{x} &= v \cos(\psi), & \dot{y} &= v \sin(\psi), \\ \dot{\psi} &= \frac{v \tan(\delta)}{L}, & \dot{v} &= \alpha(v_{ref} - v), \end{aligned} \quad (1)$$

where  $x$  and  $y$  denote the vehicle position,  $\psi$  the heading angle,  $v$  the longitudinal velocity,  $L$  the wheelbase,  $\delta$  the steering angle,  $v_{ref}$  the reference velocity, and  $\alpha$  the inverse time constant. The control inputs are  $(v_{ref}, \delta)$ . For the intersection management case study in Section V, the smart infrastructure uses this kinematic model to predict future trajectories over a finite horizon  $H$ , given the current vehicle states  $(x, y, \psi, v)$  and their intended paths  $P$ .

### C. Smart Infrastructure Design

The infrastructure in CIVAT is equipped with an Ouster OS-Dome LiDAR sensor to capture the surrounding environment

and a computation device (ASUS PE6000) that supplies power to the installed sensors and provides computation resources, including a CPU (Intel i9-12900E), a GPU (NVIDIA RTX 4090), and Wi-Fi communication capabilities. The pole is set to 1 m height to better approximate the real-world infrastructure scale. The detailed components and their prices are summarized in Tab. III. We install the infrastructure near the intersection to perceive and manage objects in a complex urban environment. An infrastructure-based system allows perception and planning to leverage powerful computational resources that are difficult to equip on CAVs, thereby enabling scalability without resource limitations.

To the best of our knowledge, this is the first study to propose infrastructure as an active agent with dedicated sensing and computing modules in a miniature testbed. Prior works have been limited either to V2V communication-based methods [23] or to passive V2I information that provides only fixed guidance, such as traffic lights or road signs [5].

TABLE III: Hardware components of infrastructure

Component	Item	Price (US\$)
Computation Device	ASUS PE6000 (i9-12900E)	2988
LiDAR	Ouster OS-Dome	7152
Pole	Aluminum Profile 50 × 50 (1m)	27

### D. Communication

To enable V2V and V2I communication, our testbed establishes a network using IEEE 802.11ax (Wi-Fi 6), interconnecting all vehicles and infrastructure. While Dedicated Short-Range Communications (DSRC) [15] and 3GPP C-V2X [8] are standardized V2X technologies, they are mutually incompatible and require specialized hardware, whereas Wi-Fi offers broader compatibility, lower cost, and easier deployment. Since our focus is on cooperative driving research that leverages exchanged information between vehicles and infrastructure, we utilize a Wi-Fi-based ROS2 publish-subscribe framework to emulate the message-passing paradigm of V2X. As Fig. 2 illustrates the V2I communication range, we set the V2I communication range to cover the intersection area rather than the entire testbed, reflecting our focus on intersections where vehicle interactions occur most frequently. In addition, the V2V communication range is capped at 3 m to emulate realistic communication constraints.

## III. SOFTWARE ARCHITECTURE

We design the software architecture of CIVAT to enable cooperative autonomous driving through both V2V and V2I communication. As shown in Fig. 5, all CAVs and the smart infrastructure exchange messages within CIVAT, allowing our testbed to support diverse cooperative driving scenarios.

### A. CAV Architecture

As shown in Fig. 5(a), the CAV architecture extend the conventional autonomous driving pipeline by incorporating cooperative driving functionality: each CAV receives messages from other agents (CAVs and infrastructure) and leverages

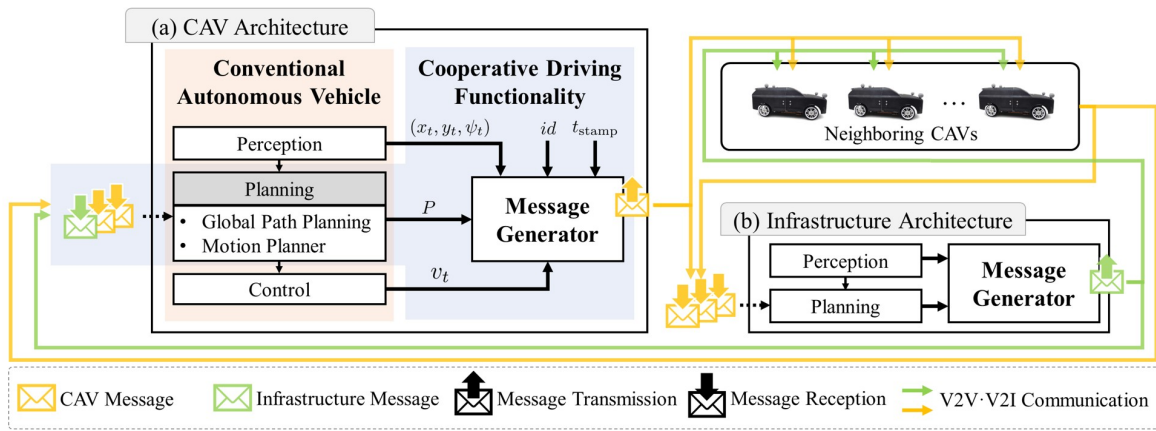


Fig. 5: Overall architecture of CIVAT

the shared information during the planning stage. At the same time, it transmits its own messages by aggregating the outputs of the perception, planning, and control modules.

In the following, we detail the planning and message generator modules. Although each vehicle is equipped with onboard sensors for perception, in the current testbed, we employ motion capture measurements to localize each CAV.

1) *Planning*: Each vehicle is assigned a destination and generates a global path using the Dijkstra algorithm [4]. The waypoints of the global path, stored in the map, provide the reference for lateral guidance. For longitudinal motion, we adopt the Intelligent Driver Model (IDM) [24]. The ego vehicle receives messages from neighboring CAVs, identifies the closest CAV on the same path, and adjusts its speed accordingly. Based on this interaction, the planner generates a trajectory that specifies both the path  $P$  through the waypoints and the velocity profile along it, thereby preserving safe following distances and ensuring collision-free motion.

2) *Message Generator*: The message generator is a core module for enabling cooperative driving. Each CAV  $n$  collects the 2D pose  $(x_t^n, y_t^n, \psi_t^n)$  from the perception module, the current path  $P^n$  from the planning module, and the estimated velocity  $v_t^n$  from the low-level controller using wheel encoder data. It then appends the CAV ID  $id^n$  and timestamp  $t_{stamp}^n$ . The message is denoted by  $M_t^n := (x_t^n, y_t^n, \psi_t^n, P^n, v_t^n, id^n, t_{stamp}^n)$  and is published at 10 Hz. The message set follows the SAE J3216 Class B standard [19], which defines the exchange of vehicle driving intents as a standardized approach to enable cooperative driving.

### B. Infrastructure Architecture

While traditional infrastructure relies on passive elements such as traffic lights and signs to manage traffic, our system enables infrastructure to participate actively and dynamically in traffic management. As illustrated in Fig. 5(b), we implement the perception and planning modules and broadcast the resulting information to nearby CAVs, thereby enabling V2X cooperative autonomous driving.

1) *Perception*: We can implement either a stand-alone infrastructure perception module or a collaborative perception

module that participates in sensor data sharing through communication. Standalone infrastructure perception outperforms vehicle perception in accuracy and robustness to noise, and in collaborative settings, V2X perception surpasses V2V perception by covering the entire intersection more comprehensively [1]. Therefore, V2X cooperative perception that leverages infrastructure is expected to play a central role in future urban autonomous driving. Collaborative perception tasks include 3D object detection [1], multi-object tracking [3], and occupancy prediction [21].

2) *Planning*: The infrastructure coordinates multiple vehicles by producing reference trajectories or direct input commands for each CAV based on perception information. Infrastructure-centric planning offers two key advantages. First, it enables globally near-optimal decisions compared to vehicle-based planning, thereby enhancing both safety and traffic efficiency [14]. Also, it supports computationally demanding planning algorithms, such as learning-based planning [7], which exceed the real-time computational capacity of individual vehicles. Our testbed enables diverse experiments in which infrastructure actively participates, including multi-vehicle coordination and intersection management.

3) *Message Generator*: The infrastructure exchanges messages with other agents through V2X communication. While we adopt the SAE J2735 standard [18] as a baseline, existing specifications primarily cover vehicle and traffic signal messages. To the best of our knowledge, there is no standardized message set dedicated to broadcasting infrastructure metadata. In CIVAT, we therefore define a customized infrastructure message set that includes control commands (e.g.,  $v_{ref}^n$  and/or  $\delta^n$ ), along with the infrastructure ID and timestamp.

## IV. CASE STUDY

In this paper, we focus on a case study of unsignalized intersection management under mixed-traffic conditions, where CAVs and human-driven vehicles (HVs) coexist on the road. As depicted in Fig. 6, the proposed intersection management framework consists of (1) infrastructure-centric 3D object detection, (2) HV identification, (3) an intersection management that determines priority and speed advisories,

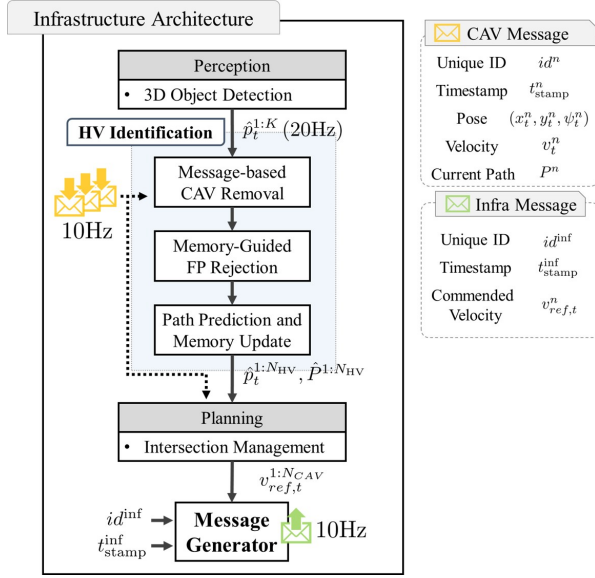


Fig. 6: Infrastructure architecture in intersection management

and (4) a message generator to broadcast both perception outputs and planning decisions to vehicles.

1) *Infrastructure-centric Object Detection*: The infrastructure must accurately localize all vehicles within the environment. To this end, we integrate an infrastructure-centric 3D object detection module into the intersection management pipeline. We employ PointPillars [13], which efficiently encodes point clouds into pseudo-images using pillar-wise feature extraction, enabling real-time inference without sacrificing accuracy. We set the LiDAR sampling rate to 20 Hz and perform object detection by predicting the corner points of 3D bounding boxes. These corner points are then converted into a 2D pose representation consisting of the center coordinates  $(\hat{x}_t^i, \hat{y}_t^i)$  and heading angle  $\hat{\psi}_t^i$ . In particular, we only consider bounding boxes that appear within the communication range, shown in Fig. 2(a), and exclude bounding boxes outside the region to reflect realistic settings.

In summary, the set of  $K$  predicted bounding boxes is denoted by  $\hat{p}_t^{1:K} := \{(\hat{x}_t^i, \hat{y}_t^i, \hat{\psi}_t^i)\}_{i=1}^K$  at time  $t$ .

2) *HV Identification*: We implement an algorithm to identify and localize HVs, as they lack communication capabilities. From the object detection results  $\hat{p}_t^{1:K}$  and CAV messages  $M_t^{1:N_{CAV}} := (M_t^1, \dots, M_t^{N_{CAV}})$  where  $N_{CAV}$  denotes the total number of CAVs at time  $t$ , we isolate HVs from a mix of CAVs, HVs, and false positives. To achieve this, we design a three-step pipeline summarized in Algorithm 1.

In **Message-based CAV Filtering**, detected bounding boxes are compared with CAV positions  $\{(x_t^n, y_t^n)\}_{n=1}^{N_{CAV}}$  obtained from messages  $M_t^{1:N_{CAV}}$ . Bounding boxes within a threshold of  $\tau_{CAV} = 13.5$  cm are removed, as they correspond to CAVs. In **Memory-Guided FP Rejection**, the remaining bounding boxes are compared with previous HV positions  $\{(\hat{x}_{t-1}^n, \hat{y}_{t-1}^n)\}_{n=1}^{N'_{HV}}$ , where  $N'_{HV}$  denotes the number of HVs at time  $t-1$ . Bounding boxes located farther than  $\tau_{FP} = 13.5$  cm from any previous HV position are discarded as false positives. In **Path Prediction and Memory Update**, the remaining

### Algorithm 1: HV Identification Procedure

---

1 **Input:**  $\hat{p}_t^{1:K}, M_t^{1:N_{CAV}}, \{(\hat{x}_{t-1}^n, \hat{y}_{t-1}^n)\}_{n=1}^{N'_{HV}}$   
2 **Output:**  $\hat{p}_t^{1:N_{HV}}, \hat{P}^{1:N_{HV}}$   
3 **Note:**  $\text{dist}(\hat{p}_t^i, (x_t^j, y_t^j)) := \sqrt{(\hat{x}_t^i - x_t^j)^2 + (\hat{y}_t^i - y_t^j)^2}$   
4 **Step 1 — Message-based CAV Filtering**  
5 **for each detection**  $\hat{p}_t^i$  **do**  
6     **if**  $\min_{(x_t^n, y_t^n)} \text{dist}(\hat{p}_t^i, (x_t^n, y_t^n)) < \tau_{CAV}$  **then**  
7         Remove  $\hat{p}_t^i$   
8 **Step 2 — Memory-Guided FP Rejection**  
9 **for each remaining detection**  $\hat{p}_t^i$  **do**  
10     **if**  $\min_{(\hat{x}_{t-1}^n, \hat{y}_{t-1}^n)} \text{dist}(\hat{p}_t^i, (\hat{x}_{t-1}^n, \hat{y}_{t-1}^n)) \geq \tau_{FP}$  **then**  
11         Remove  $\hat{p}_t^i$   
12     **else**  
13         Keep  $\hat{p}_t^i$  as HV  
14 **Step 3 — Path Prediction & Memory Update**  
15 **for each HV**  $\hat{p}_t^i$  **do**  
16     Find the nearest node and generate candidate paths  
17     Discard paths with  $\text{dist}(\hat{p}_t^i, \text{path}) > \tau_p$   
18     Store remaining candidate(s) as  $\hat{P}^i$   
19 **Update memory:**  $\hat{p}_t^{1:N_{HV}}, \hat{P}^{1:N_{HV}}$

---

### Algorithm 2: Priority-based Intersection Management

---

1 **Input:**  $M_t^{1:N_{CAV}}, \hat{p}_t^{1:N_{HV}}, \hat{P}^{1:N_{HV}}$   
2 **Output:**  $\{v_{ref,t}^i\}_{i=1}^{N_{CAV}}$   
3 **if CAV in intersection then**  
4     **if HVs present (identified using Algorithm 1) then**  
5         Assign the highest priority to HVs  
6         Predict future occupied region using  
7          $\hat{p}_t^{1:N_{HV}}, \hat{P}^{1:N_{HV}}$ , and  $v_{max}$   
8     **for each CAV**  $i$  **do**  
9         Assign priority in entry-time order  
10          $v_{ref,t}^i \leftarrow v_{max}$   
11         **while**  $v_{ref,t}^i \geq 0$  **do**  
12             Predict CAV  $i$ 's future occupied region  
13             using  $M_t^{1:N_{CAV}}$  and constant  $v_{ref,t}^i$   
14             **if Conflict higher-priority occupied region**  
15                 **then**  
16                      $v_{ref,t}^i \leftarrow v_{ref,t}^i - \Delta v_{step}$  (**Yield**)  
17                 **else**  
18                     **Break loop**  
19         **Store command:**  $\{v_{ref,t}^i\}_{i=1}^{N_{CAV}}$

---

bounding boxes are classified as HVs. Each HV position is matched to nearby entry nodes, and candidate paths are generated from the connected road segments. Paths deviating more than  $\tau_p = 20$  cm from the HV position are eliminated.

3) *Intersection Management*: To coordinate vehicles inside the intersection, the infrastructure regulates each vehicle's velocity using a priority-based rule, as summarized in Algorithm 2. It first collects neighboring-vehicle information  $M_t^{1:N_{\text{CAV}}}$  and HV states  $\hat{p}_t^{1:N_{\text{HV}}}$  and predicted paths  $\hat{P}^{1:N_{\text{HV}}}$  through HV identification. To secure the applicability of diverse experiments, we adopt different priority rules depending on the traffic composition: (1) the fully CAV condition, and (2) the mixed-traffic condition.

For the fully CAV condition, we apply a First-In-First-Serve (FIFS) rule as a deterministic conflict resolution mechanism. Vehicles are prioritized by their order of entry into the intersection. The infrastructure then predicts their future occupied regions over the prediction horizon  $H$  steps using a kinematic bicycle model (1). The highest-priority vehicle is assigned the maximum velocity  $v_{\text{max}}$ . For subsequent vehicles, the infrastructure first assigns the maximum velocity  $v_{\text{max}}$  and predicts future trajectories. If a collision is detected, the velocity is iteratively reduced by  $\Delta v_{\text{step}}$  until a collision-free trajectory is obtained. The resulting value is issued as the control command  $v_{\text{ref},t}^i$  to the  $i$ -th CAV.

For the mixed-traffic condition, when an HV enters the intersection, it is automatically assigned the highest priority, and its future occupied regions are predicted assuming motion at maximum velocity. If HV identification yields multiple candidate paths, the infrastructure predicts occupied regions for all candidates, and adjusts the commanded velocities of CAVs to ensure they avoid these regions.

4) *Message Generator*: The message to CAV  $n$  includes the unique ID of infrastructure  $id^{inf}$  to allow agents to identify the source, a timestamp  $t_{\text{stamp}}^{inf}$ , and the commanded velocity  $v_{\text{ref},t}^n$ , and is published at 10Hz. The 10 Hz update rate is chosen to reflect typical autonomous vehicle communication settings rather than limitations of our system's computation or scalability. The commanded velocity directly influences the vehicle's control, enabling efficient and collision-free passage through the intersection.

## V. EXPERIMENTS

### A. Infrastructure Perception

1) *Dataset*: For 3D object detection, we construct a dataset collected from standalone infrastructure in CIVAT, referred to as CIVAT-SI. The dataset contains scenarios with up to five vehicles and consists of 3,888 training frames and 1,120 validation frames. Raw sensor data are captured using infrastructure-mounted LiDAR, and ground-truth vehicle poses are obtained from OptiTrack measurements.

2) *Detection Accuracy*: Fig. 7 illustrates the 3D object detection results obtained from the infrastructure LiDAR. As shown in Fig. 7(a) and summarized in Tab. IV, most vehicles are detected accurately. Nonetheless, inaccurate or failed detections also occur. In particular, as indicated by the yellow dashed boxes in Fig. 7(b), vehicles located in occluded regions or areas with sparse point clouds often yield inaccurate predictions. Furthermore, as illustrated in the yellow dashed boxes Fig. 7(c), when vehicles are in close proximity, both false positives and false negatives may arise.

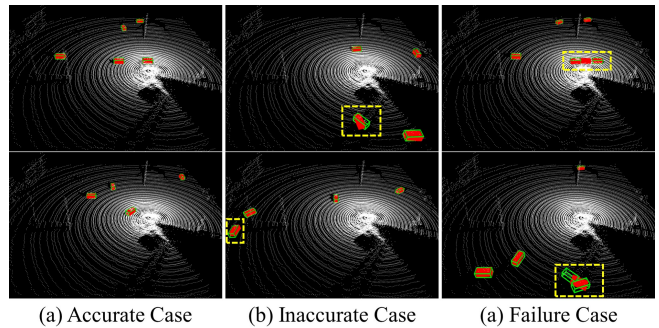


Fig. 7: Detection results of infrastructure LiDAR. Green and red boxes denote ground truth and predictions, respectively.

TABLE IV: Detection accuracy on CIVAT-SI.

Model	AP@0.3	AP@0.5	AP@0.7
PointPillars [13]	0.984	0.972	0.802

### B. Intersection Management

We validate the infrastructure-centric intersection management framework under a fully CAV condition and a mixed-traffic condition. The maximum vehicle velocity is set to  $v_{\text{max}} = 0.5$  m/s with a velocity decrement step of  $\Delta v_{\text{step}} = 0.1$  m/s. The prediction horizon is set to  $H = 30$ , with a sampling time of  $\Delta t = 0.1$  s. A safety buffer  $b_{\text{safe}} = 30$  cm is applied when predicting the future occupied region.

1) *Fully CAV Condition*: We consider a scenario in which five CAVs enter the intersection, as shown in Fig. 8. At  $t = 1.6$  s, the first-priority vehicle (white) and the second-priority vehicle (purple) follow non-conflicting paths; thus, the infrastructure assigns both the maximum allowable speed,  $v_{\text{ref},1.6}^1 = v_{\text{ref},1.6}^2 = 0.5$  m/s. In contrast, at  $t = 2.2$  s, the third-priority (orange) vehicle conflicts with the white vehicle path and therefore yields by decelerating to  $v_{\text{ref},2.2}^3 = 0.4$  m/s. At later times ( $t = 2.9$  s and  $t = 3.7$  s), the infrastructure continues to regulate vehicle velocities according to the priority, ensuring collision-free coordination.

2) *Mixed-traffic Condition*: We evaluate a mixed-traffic condition with four CAVs and one HV entering the intersection concurrently, as shown in Fig. 9. At  $t = 1.3$  s, the HV (purple) has two possible straight or left-turn candidate paths. At this stage, all CAVs stop to yield, allowing the HV to proceed first. By  $t = 3.8$  s, when the HV's position is closer to the left-turn waypoint and the distance to the straight path exceeds  $\tau_p$ , the straight path is discarded and the left-turn path is confirmed as its trajectory. After the HV completes the crossing ( $t = 7.4$  s and  $t = 9.3$  s), the infrastructure resumes priority-based management of the remaining CAVs.

As shown in Fig. 10, at  $t = 4.6$  s the detection results contain both a false positive (FP) and a false negative (FN). The FP is successfully removed by the HV identification algorithm using the previous HV bounding box, while the FN of CAV does not affect management because their positions are determined from the CAV message. These results highlight the complementary roles of V2I communication and infrastructure perception: (1) V2I communication mitigates the effects of

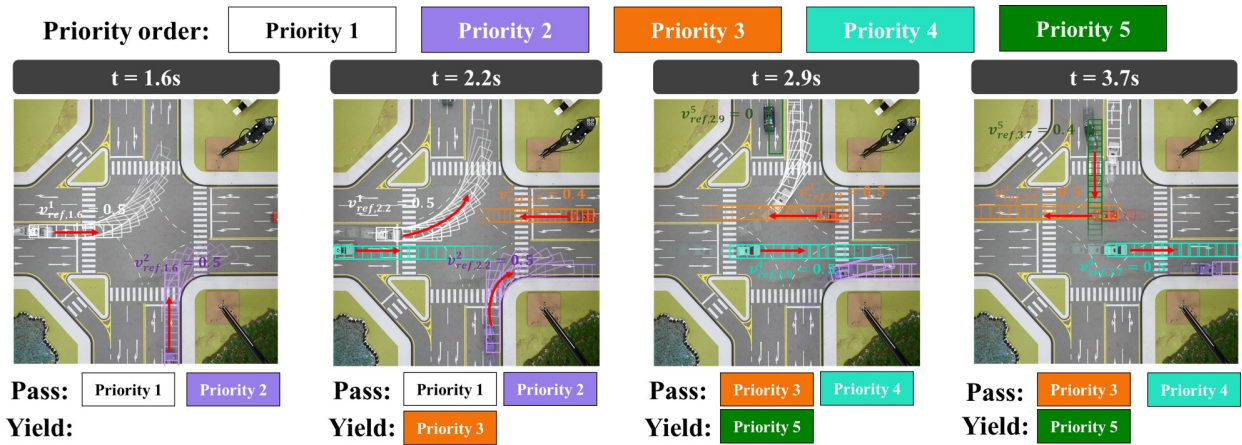


Fig. 8: Intersection management results in the fully CAV scenario. Vehicle priorities are indicated at the top. The pass indicates that vehicles approach and pass the intersection with high priority. Yield means that vehicles slow down and halt with low priority. The gradient boxes on the road indicate the occupied regions of the corresponding color-coded agents.

unstable infrastructure-based perception, while (2) perception compensates for the limitations of V2I communication in handling non-communicating agents such as HVs.

Hence, these experiments show that our testbed highlights the applicability of infrastructure intersection management systems, which have been evaluated mainly in simulations.

### C. Computation Time

Tab. V shows the computation times of object detection, HV identification, and intersection management. The computation times are measured from a scenario under the mixed-traffic conditions with four CAVs and one HV. The average total computation time of the intersection management framework is 31.79 ms, with a maximum of 40.39 ms. Given the 100 ms cycle time of a 10 Hz system, these results demonstrate that our algorithm satisfies real-time applicability.

As the number of vehicles increases, the computation time of HV identification and intersection management grows due to additional vehicle comparisons and conflict checks. However, these coordination modules are not the computational bottleneck: Tab. V shows that they require only 0.212 ms on average and 1.140 ms at maximum, whereas object detection alone takes 31.58 ms on average, accounting for 99.3% of the total runtime.

TABLE V: Computation time by modules (ms)

Module	Min.	Max.	Average
Object Detection	27.66	39.25	31.58
HV Identification	0.020	0.123	0.062
Intersection Management	0.003	1.017	0.150
<b>Total</b>	27.69	40.39	31.79

## VI. CONCLUSION

In this paper, we introduced CIVAT, a 1:15-scale testbed for cooperative autonomous driving that integrates miniature vehicles and smart infrastructure. The infrastructure provides 3D LiDAR and computation for perception, communication, and coordination. To demonstrate its utility, we presented an intersection management case study where infrastructure

perception and V2I communication coordinate CAVs under mixed-traffic conditions. CIVAT provides a cost-effective and scalable platform for evaluating cooperative driving algorithms in both fully connected and mixed-traffic scenarios.

To further demonstrate the utility of CIVAT, we deployed a second instance and hosted the first KAIST Mobility Challenge, where the platform served as an educational testbed for cooperative driving.<sup>2</sup> Future work will incorporate onboard sensor-based localization and perception, including an optional SLAM module, to enable end-to-end autonomy evaluation. We will also improve communication realism by emulating V2X impairments (delay, jitter, and packet loss) and potentially integrating standardized V2X communication such as DSRC or C-V2X.

## ACKNOWLEDGMENT

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-ITRC(Information Technology Research Center) grant funded by the Korea government(MSIT)(IITP-2026-RS-2023-00259991, 100%)

## REFERENCES

- [1] H. Bae, M. Kang, M. Song, and H. Ahn, "Rethinking the role of infrastructure in collaborative perception," in *European Conf. on Computer Vision Workshop (ECCVW)*, 2024.
- [2] S. Brennan and A. Alleyne, "A scaled testbed for vehicle control: The IRS," in *IEEE Inter. Conf. on Control Applications (CCA)*, 1999.
- [3] H.-K. Chiu, C.-Y. Wang, M.-H. Chen, and S. F. Smith, "Probabilistic 3d multi-object cooperative tracking for autonomous driving via differentiable multi-sensor kalman filter," in *IEEE Inter. Conf. on Robotics and Automation (ICRA)*, 2024, pp. 18 458–18 464.
- [4] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: his life, work, and legacy*, 2022.
- [5] J. Dong, Q. Xu, J. Wang, C. Yang, M. Cai, C. Chen, Y. Liu, J. Wang, and K. Li, "Mixed cloud control testbed: Validating vehicle-road-cloud integration via mixed digital twin," *IEEE Trans. on Intell. Vehicles*, 2023.
- [6] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conf. on Robot Learning (CoRL)*, 2017.

<sup>2</sup><https://www.youtube.com/watch?v=cH-VkOgAqHc>

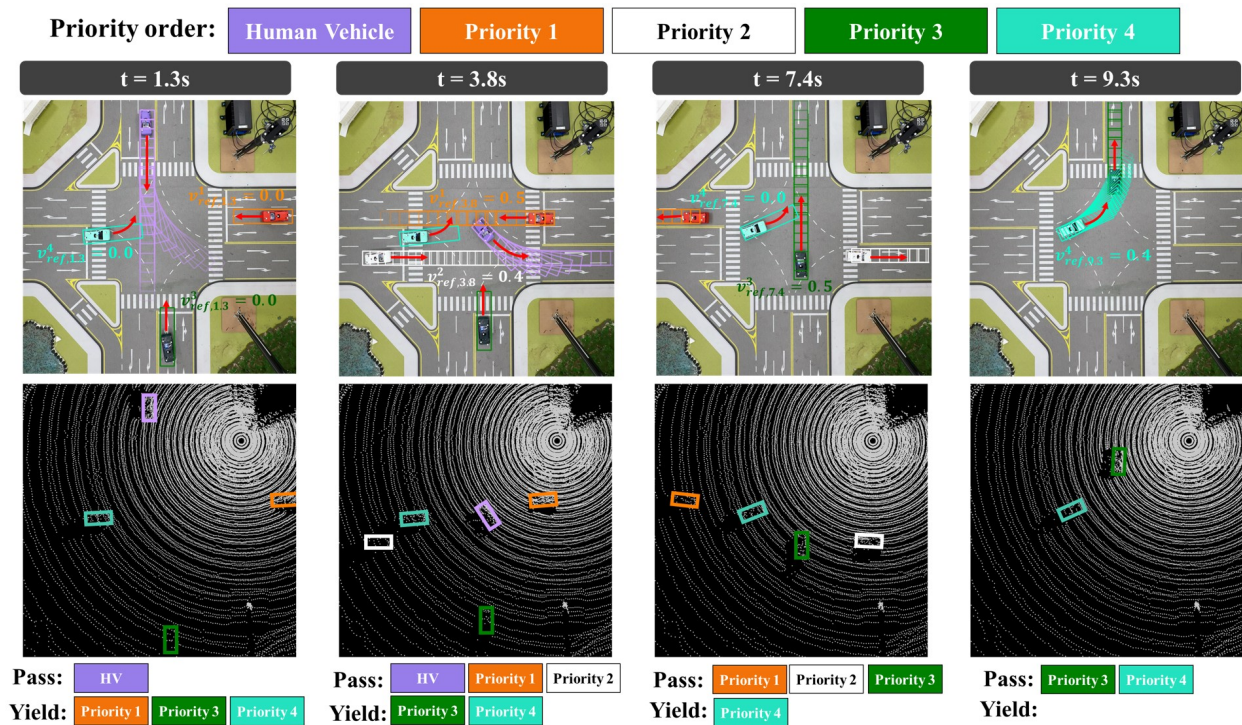


Fig. 9: Intersection management and LiDAR detection results in the mixed-traffic scenario.

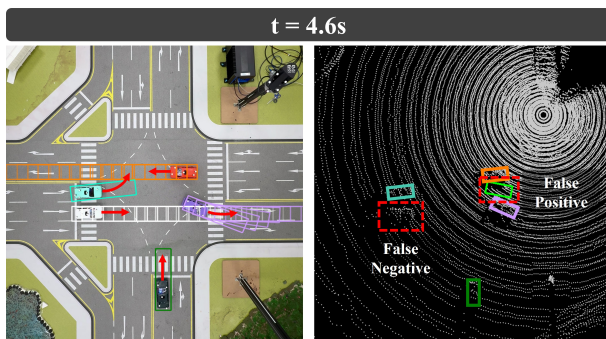


Fig. 10: Edge case in mixed-traffic intersection management

[7] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[8] M. Harounabadi, D. M. Soleymani, S. Bhadauria, M. Leyh, and E. Roth-Mandutz, "V2x in 3gpp standardization: Nr sidelink in release-16 and beyond," *IEEE Communications Standards Magazine*, vol. 5, no. 1, pp. 12–21, 2021.

[9] N. Hyldmar, Y. He, and A. Prorok, "A fleet of miniature cars for experiments in cooperative driving," in *IEEE Inter. Conf. on Robotics and Automation (ICRA)*, 2019.

[10] S. Kannapiran and S. Berman, "Go-CHART: A miniature remotely accessible self-driving car robot," in *IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[11] M. Kloock, P. Scheffe, J. Maczjowski, A. Kampmann, A. Mokhtarian, S. Kowalewski, and B. Alrifae, "Cyber-physical mobility lab: An open-source platform for networked and autonomous vehicles," in *European Control Conf. (ECC)*, 2021.

[12] Korea Transportation Safety Authority, "K-City: Korea's Autonomous Vehicle Proving Ground." [Online]. Available: <https://www.kotsa.or.kr/eng>

[13] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] E. Lee, M. Kang, Y. Choi, and H. Ahn, "A decision transformer for generalizable multi-vehicle coordination at unsignalized intersections," in *IEEE Intelligent Vehicles Symposium (IV)*, 2026.

[15] Y. Li, "An overview of the dsrc/wave technology," in *Proc. Int. Conf. Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 2010.

[16] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang, *et al.*, "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," in *IEEE Inter. Conf. on Robotics and Automation (ICRA)*, 2017.

[17] RISE Research Institutes of Sweden, "AstaZero– Active Safety Test Area and Zero." [Online]. Available: <https://www.astazero.com/>

[18] SAE J2735: V2X Communications Message Set Dictionary, SAE International Std. J2735, 2020. [Online]. Available: [https://www.sae.org/standards/content/j2735\\_202007/](https://www.sae.org/standards/content/j2735_202007/)

[19] SAE J3216: Taxonomy and Definitions for Terms Related to Cooperative Driving Automation for On-Road Motor Vehicles, SAE International Std. J3216, 2021. [Online]. Available: [https://www.sae.org/standards/content/j3216\\_202107/](https://www.sae.org/standards/content/j3216_202107/)

[20] J. M. Snider, "Automatic steering methods for autonomous automobile path tracking," Robotics Institute, Carnegie Mellon University, Tech. Rep., 2009.

[21] R. Song, C. Liang, H. Cao, Z. Yan, W. Zimmer, M. Gross, A. Festag, and A. Knoll, "Collaborative semantic occupancy prediction with hybrid feature fusion in connected automated vehicles," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[22] A. Stager, L. Bhan, A. Malikopoulos, and L. Zhao, "A scaled smart city for experimental validation of connected and automated vehicles," *IFAC-PapersOnLine*, 2018.

[23] Z. Tian, Y. He, B. Tian, R. Zhong, E. Foerginejad, and W. Shi, "ICAT: An indoor connected and autonomous testbed for vehicle computing," in *IEEE Inter. Conf. on Mobility, Operations, Services and Technologies (MOST)*, 2024.

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

[25] University of Michigan, "Mcity: University of Michigan's Test Facility for Connected and Automated Vehicles." [Online]. Available: <https://mcity.umich.edu>