
OpenCDA- ∞ : A Closed-loop Benchmarking Platform for End-to-end Evaluation of Cooperative Perception

Chia-Ju Chen¹, Runsheng Xu¹, Wei Shao², Junshan Zhang², Zhengzhong Tu³

¹UCLA, ²UC Davis, ³Texas A&M University

Abstract

Vehicle-to-vehicle (V2V) cooperative perception systems hold immense promise for surpassing the limitations of single-agent lidar-based frameworks in autonomous driving. While existing benchmarks have primarily focused on object detection accuracy, a critical gap remains in understanding how the upstream perception performance impacts the system-level behaviors—the ultimate goal of driving safety and efficiency. In this work, we address the crucial question of how the detection accuracy of cooperative detection models natively influences the downstream behavioral planning decisions in an end-to-end cooperative driving simulator. To achieve this, we introduce a novel simulation framework, **OpenCDA- ∞** , that integrates the OpenCDA cooperative driving simulator with the OpenCOOD cooperative perception toolkit. This feature bundle enables the holistic evaluation of perception models by running any 3D detection models inside OpenCDA in a real-time, online fashion. This enables a closed-loop simulation that directly assesses the impact of perception capabilities on safety-centric planning performance. To challenge and advance the state-of-the-art in V2V perception, we further introduce the **OPV2V-Safety** dataset, consisting of twelve challenging and pre-crash open scenarios designed following the National Highway Traffic Safety Administration (NHTSA) reports. Our findings reveal that OPV2V-Safety indeed challenges the prior state-of-the-art V2V detection models, while our safety benchmark yielded new insights on evaluating perception models as compared to the results on prior standard benchmarks. We envision that our end-to-end, closed-loop benchmarking platform will drive the community to rethink how perception models are being evaluated at the system level for the future development of safe and efficient autonomous systems. The code is available at <https://github.com/taco-group/opencda-loop>.

1 Introduction

Accurate, robust, and rapid perception of complex and dynamic environments is essential for responsible autonomous driving. Recent advances in robotic sensing equipped with advanced machine learning techniques have fueled perception performance, evidenced by successes in tasks such as 3D object detection, tracking, and semantic map segmentation. However, these advancements often falter in scenarios featuring extensive occlusions, small or distant objects, potentially leading to catastrophic outcomes due to insufficient sensor data coverage, which underscores the challenges inherent in single-vehicle perception systems limited by physical constraints and occlusions.

To overcome these limitations, recent studies have pivoted towards multi-vehicle cooperative frameworks that leverage Vehicle-to-Everything (V2X) or Vehicle-to-Vehicle (V2V) communication technologies. These frameworks empower Connected and Automated Vehicles (CAVs) to share a diversity of data forms—from raw sensor outputs like LiDAR point clouds, RGB images, and radar frames to processed features and detection results—thereby collaboratively enhancing perception capabilities by amalgamating multiple vehicular perspectives. Despite their potential, these technologies’ evolution

is predominantly driven by the development of diverse, large-scale, and open-sourced datasets and benchmarks. For instance, initiatives such as OPV2V [1], V2X-ViT [2], and V2X-Sim [3] have mainly utilized simulation platforms like CARLA [4] and SUMO [5] to create extensive synthetic datasets tailored for cooperative perception tasks. Yet, traditional evaluations using metrics like Average Precision (AP) fall short of capturing the full spectrum of autonomous driving requirements, particularly in ensuring safe driving behaviors and robust vehicular planning.

To this end, here we introduce a novel framework that marries OpenCDA cooperative driving co-simulation platform [6] and the OpenCOOD cooperative perception toolset [1], which we dub **OpenCDA- ∞** , allowing for holistic development and testing of cooperative perception models in a closed-loop, end-to-end fashion that mainly focuses on safety-centric evaluation. In other words, we can directly assess how the perception performance of V2V algorithms impacts the actual driving behavior and safety implications of the vehicles. To achieve this, we have made several enhancements to the OpenCDA simulation platform. First, we incorporated the OpenSCENARIO standard [7, 8] for precise actor controls (vehicles, pedestrians, etc.) in simulation, leading to more realistic and customizable scenarios. Second, we’ve added the capability to run cooperative perception models in real-time during simulation, enabling a true closed-loop evaluation. Third, we’ve integrated advanced modules for vehicle trajectory prediction and robust behavior planning to ensure that the ego vehicle (the one we’re controlling) makes intelligent decisions based on the perceived information.

Moreover, we build the **OPV2V-Safety** dataset, comprising twelve diverse and challenging pre-crash traffic scenarios cataloged by the National Highway Traffic Safety Administration (NHTSA) [9], tailored to test the robustness of V2V perception and planning algorithms under adverse conditions. This dataset, featuring 4,377 frames, serves as a critical testbed for evaluating state-of-the-art 3D object detection techniques and multi-vehicle fusion strategies from a planning perspective. We move beyond standard detection accuracy metrics and introduce a multi-tiered safety-critical evaluation suite. Our metrics encompass not only the quality of object detection but also the robustness, efficiency, and stability of the overall cooperative perception system. This holistic approach provides a deeper understanding of how different perception models impact autonomous vehicles’ system-level performance and safety. Our extensive benchmarking results on OPV2V-C, using various V2V algorithms, reveal that models that excel in traditional detection accuracy metrics do not necessarily lead to the best planning outcomes or the safest driving behaviors. This underscores the importance of our system-level evaluation approach and the value of the OPV2V-C dataset in driving the development of more robust and safety-conscious V2V autonomous driving systems.

In summary, our contributions are manifold: ❶ We propose a closed-loop, end-to-end simulation platform called **OpenCDA- ∞** that facilitates the *planning-oriented* evaluation of cooperative perception models at a system level. ❷ We extend the capabilities of OpenCDA with advanced functionalities, including realistic scenario customization and robust behavior planning, enabling real-time, online simulation, and comprehensive assessment of any perception models. ❸ We release the **OPV2V-Safety** dataset, a safety-critical testbench comprising diverse corner-case scenarios that can rigorously test existing V2V perception models and planning algorithms, which can facilitate the development of more safety-critical autonomous systems. ❹ A multi-tiered safety evaluation metric suite beyond traditional detection metrics has been provided, offering deeper insights into the safety and effectiveness of cooperative perception systems. ❺ Our extensive benchmarking results on state-of-the-art cooperative perception models highlight the importance of our benchmarking platform in regard to system-level evaluation of V2V perception.

2 Related Work

Autonomous driving datasets. Publicly available, large-scale datasets always play a fundamental role in advancing any machine learning field, and autonomous driving is no exception. The pioneering KITTI dataset [10], a trailblazer in providing multimodal sensor data, marked a significant leap towards data-driven autonomous learning with its front-facing stereo cameras and LiDAR across 22 sequences. Subsequent community efforts have escalated the scale and complexity of KITTI, including diversity in driving scenarios, sensor modalities, and data annotations that can be employed to train larger, multimodal algorithms for diverse vision and planning tasks. For example, the NuScenes [11] and Waymo Open dataset [12] are two representative multimodal datasets that consist of a significantly broader array of annotated RGB images and LiDAR point clouds, enabling more performant and robust vehicle and pedestrian detection models.

End-to-end autonomous driving. Significant progress has been made in end-to-end autonomous driving. UniAD [13] integrated full-stack driving tasks in a single network with query-unified interfaces. ReasonNet [14] improved perception by leveraging temporal and global scene information for better occlusion detection. ASAP-RL [15] proposed an efficient reinforcement learning algorithm for autonomous driving that simultaneously leverages motion skills and expert priors. InterFuser utilized a transformer-based framework for multi-modal sensor fusion [16]. Coopernaut [17] enhanced V2V cooperative driving with cross-vehicle perception and vision-based decision-making. LMDrive [18] incorporated large language models, enabling natural language interaction and improving reasoning in complex scenarios. Approaches like Latent DRL [19] and Roach [20] utilized reinforcement learning to enhance decision-making, while ScenarioNet [21] and TrafficGen [22] generated diverse driving scenarios for testing. However, this end-to-end driving automation merely focuses on single-agent-based approaches, and a system that incorporates cooperative detection methods in a closed-loop simulator is in pressing need.

V2X/V2V cooperative systems and datasets. Despite the rapid progress in single-vehicle autonomous driving, it still encounters substantial challenges in complex real-world scenarios, such as extreme occlusions and limited long-range perception capabilities [23]. Recent advancements in Vehicle-to-Everything (V2X) (including Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I)) technologies have enabled vehicles to connect, communicate, and collaborate, significantly expanding their perception range as well as compensating each other to collaboratively handling occlusion via shared viewpoints. OPV2V [1] paves the way by constructing a novel 3D cooperative detection dataset using CARLA and OpenCDA co-simulation. Other studies like V2X-ViT [2] and V2X-Sim [3] leverage the capabilities of smart infrastructure in conjunction with connected vehicles to enable Vehicle-to-Everything (V2X) perception. In contrast to these simulated datasets, DAIR-V2X [24] and V2V4Real [25] provide large-scale real-world data for cooperative detection research, establishing benchmarks on realistic and dynamic traffic scenarios.

V2X/V2V cooperative perception models. Cooperative systems have emerged as powerful tools for addressing the inherent limitations of single-vehicle perception, enabling a paradigm shift towards multi-vehicle detection. The landscape of V2V and Vehicle-to-Everything (V2X) cooperative perception can be broadly segmented into three categories: ① *Early Fusion*, where raw point clouds are shared among Connected Autonomous Vehicles (CAVs), allowing the ego vehicle to draw predictions based on the assembled raw data [26]; ② *Late Fusion*, where detection outputs (e.g., 3D bounding boxes, confidence scores) are exchanged, which are subsequently fused into a single 'consensus' prediction [27]; and ③ *Intermediate Fusion*, where intermediate feature maps or representations are derived from each agent's observation and then shared among the other CAVs [23, 28, 1, 29]. These categories encapsulate the diverse ways in which cooperative systems can be leveraged to enhance the breadth and depth of perception in autonomous driving.

Recent frontier cooperative detection models predominantly adopt intermediate fusion strategies where the intermediate neural features computed from each agent's sensor data are broadcasted, achieving the best trade-off between accuracy and bandwidth requirements. Specific examples include F-Cooper [28], which devises a simple max-pooling operation to fuse intermediate visual features, while V2VNet [23] employs graph neural networks to fuse shared features from connected vehicle nodes. Additionally, Coopernaut [17] uses Point Transformer [30] to process shared point features. Inspired by efficient axial vision Transformers [31, 32], CoBEVT [33] introduces an innovative local-global sparse attention mechanism that captures spatial interactions among different views and agents, and AttFuse [1] suggests an agent-specific self-attention module to fuse the received features. V2X-ViT [2, 34] designs a unified vision transformer optimized for multi-agent, multi-scale perception, delivering robust performance even under conditions of GPS error and communication delay. More recently, CoMamba [35] has introduced a novel state-space model-based 3D detection framework for real-time onboard vehicle perception.

3 OpenCDA- ∞ : An Online, Closed-loop, End-to-end Simulator

3.1 OpenCDA Simulation Platform

OpenCDA [6] is a simulation-integrated framework for dynamic cooperative driving automation (CDA) research, which supports a broad range of automated vehicle interactions through a benchmarking scenario database and trending CDA algorithms. As illustrated in Fig. 1, OpenCDA coherently

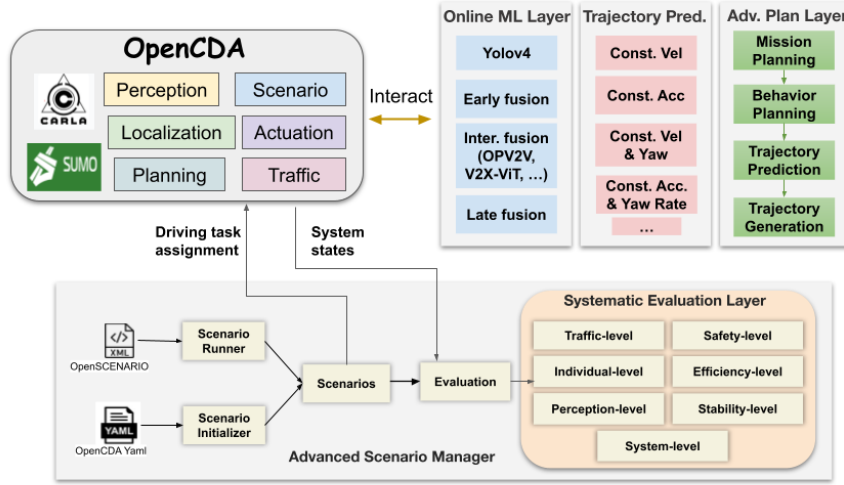


Figure 1: **OpenCDA- ∞** : a closed-loop, end-to-end simulation platform that bridges two software suites: the cooperative driving simulation platform OpenCDA and the cooperative perception toolkit OpenCOOD. We further enhance this platform with advanced modules, including OpenSCENARIO customization (Sec. 3.3), online cooperative detection (Sec. 3.2), trajectory prediction and planning (Sec. 3.4). Finally, we build **OPV2V-Safety**, a challenging, pre-crash scene dataset, equipped with a spectrum of evaluation metrics for examining cooperative perception models.

integrates several core components: simulation tools, a Python-based CDA system, and an extensive scenario manager. For the simulation tools, OpenCDA utilizes CARLA [4], a free open-source driving simulator that boasts high-quality rendering capabilities powered by the Unreal Engine. The scenario manager of OpenCDA is structured into four main elements: the configuration file, initializer, event trigger, and evaluation functions. Scenarios blend static elements, such as road structures defined by CARLA’s assets, with dynamic features managed by a YAML config file. Central to its design is the application layer, where CAVs exchange data and strategies, such as blending individual and communal sensing data for improved perception. OpenCDA provides both default and customizable protocols, enabling researchers to evaluate the entire CDA system or to conduct comparative analyses of specific algorithms. We refer the readers to the Appendix for more details regarding OpenCDA.

3.2 Online Cooperative Detection

Most simulation platforms available today, including OpenCDA, do not support the real-time operation of trained models; instead, they heavily rely on the offline evaluation of detection accuracy. This static approach fails to reflect the dynamic interplay that occurs in real-world driving scenarios where detection results continuously influence downstream planning and decision-making and, in turn, further determine the next system state for perception. This process creates a feedback loop that dynamically evolves based on real-time data and interactions. Unfortunately, current driving simulation benchmarks [1, 3, 2] fail to account for these feedback mechanisms, focusing instead on static outputs that do not measure the adaptive performance of systems under changing conditions.

To this end, we make a big step forward to fill this gap by amalgamating the current offline cooperative perception toolkit OpenCOOD [1] into the OpenCDA simulator. Specifically, compiling OpenCOOD as an additional MLManager component, our enhanced **OpenCDA- ∞** can now not only run a diverse array of cooperative detection models on the fly but also allow the outputs of these models to directly steer the planning and decision-making processes of its autonomous agents. This enriched feature makes it possible to investigate the influence of state-of-the-art cooperative perception models at a system level, which can more faithfully simulate real-world scenarios.

We would like to re-emphasize the importance of building a simulator that runs online detection models, as we have later surfaced in Sec. 5.2 that detection accuracy, although a relatively reliable metric, does not necessarily strictly reveal the overall rank in terms of planning performance. Instead, examining other metrics beyond detection, such as safety- or efficiency-level metrics, can usually offer

more informative insights into the system-level evaluation in various aspects. On the one hand, this type of real-time testing capability allows us to directly test and refine models within an end-to-end simulated environment that accurately reflects the unpredictability of real-world driving, thus largely reducing the development time before onboard deployment. On the other hand, this approach serves as a testbench that supports the crucial phase of the sim-to-real generalization research.

3.3 OpenSCENARIO Add-ons

OpenCDA, by default, utilizes the built-in CARLA traffic manager to simulate vehicle dynamics, automatically computing routes from initial spawn points to destinations. However, this approach provides limited control over the specific behaviors of individual actors, which can be restrictive when generating complex scenarios. OpenSCENARIO [7], a standardized XML-based language for driving scenarios, offers a structured method to create complex, reproducible, and configurable simulations that range from simple straight-road driving to intricate urban settings with multiple dynamic actors. This framework not only facilitates the scripting of detailed scenarios but also supports the encoding of high-level traffic rules and participant behaviors. We integrated this feature with CARLA through the ScenarioRunner extension, allowing us to construct highly challenging pre-crash scenarios through accurate agent behavior control. More details are in the Appendix.

3.4 Trajectory Prediction and Behavior Planning

OpenCDA originally did not support real-time trajectory prediction, limiting our ability to explore how predicted vehicle movements impact subsequent planning in automated driving systems. To address this, we have incorporated a trajectory prediction module capable of simulating realistic traffic scenarios and driver behaviors. We implemented various common trajectory prediction models in OpenCDA- ∞ : ❶ *Constant Velocity*: Suitable for steady traffic flow, predicting linear vehicle movements as $x = vt$. ❷ *Constant Acceleration*: Useful for scenarios of acceleration or deceleration, modeled by $v = u + at$. ❸ *Constant Speed and Yaw Rate*: Applies to vehicles moving at a constant speed but changing direction, described as $\theta = \omega t$. ❹ *Constant Acceleration and Yaw Rate*: Combines linear and angular dynamics for scenarios like exit ramps. ❺ *Physics Oracle Model*: A comprehensive model for predicting complex maneuvers in high-stakes environments.

OpenCDA utilizes a rule-based finite-state machine for planning, dynamically responding to specific traffic scenarios. This system transitions through several states, including route calculation, lane changing, overtaking, and adaptive speed regulation based on proximity to obstacles (see Appendix.A.1.1 for details). The initial planning algorithms implemented in OpenCDA constantly fail to complete the planned global route without incorporating prediction models, particularly in scenarios involving complex intersections, lane mergers, or vehicles emerging from blind spots. To enhance the planning capability, in OpenCDA- ∞ , we established a collision prediction mechanism that assesses potential future collisions within a specified lookahead time window, T seconds. Given the uncertainty of velocity and traffic conditions, we defined configurable parameters that delineate the range of possible future positions, as explained in Fig. 2, formalized as follows:

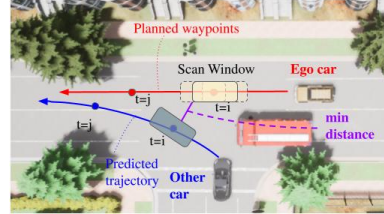


Figure 2: The diagram of our collision check model for robust planning with trajectory prediction.

$$L = \min_{t \in T} \min_{r \in [t-\tau, t+\tau]} |\mathbf{x}_r - \mathbf{y}_r|_2, \quad (1)$$

where $T = t_1, t_2, \dots, t_K$ represents a uniform time series, \mathbf{x}_r and \mathbf{y}_r are the respective positions of the ego and threat vehicles at time r , and τ accounts for prediction error. If L exceeds a predefined safe distance, the planning algorithm adjusts to mitigate collision risk.

4 The OPV2V-Safety Dataset and Benchmark

In this section, we will detail how we collect the **OPV2V-Safety** dataset and build the benchmark for end-to-end evaluation of cooperative perception models in our end-to-end **OpenCDA- ∞** simulator.

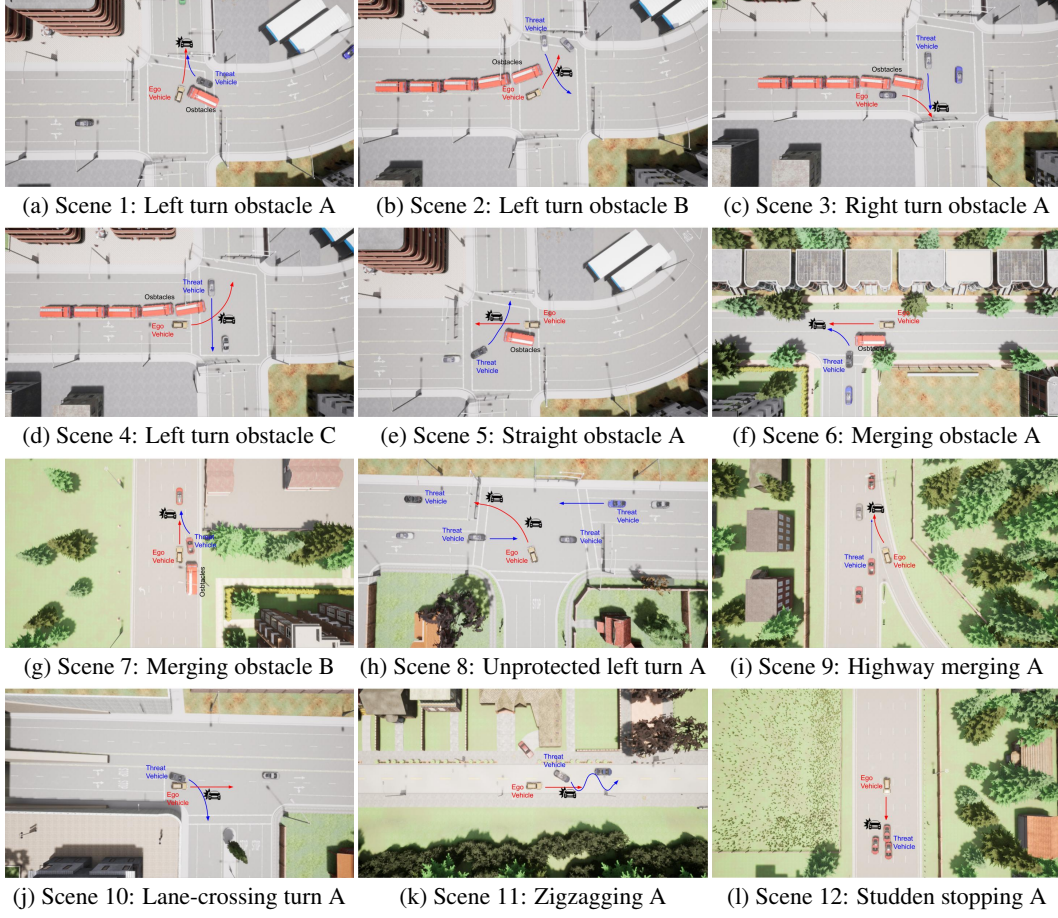


Figure 3: The visualization of potential pre-crash moments in the OPV2V-Safety Benchmark.

4.1 Data Protocols

Scenario Setting. We generate the scenarios using the eight default towns, which are directly available in CARLA for easy reproducibility. In each scenario, we follow the safety-critical pre-crash traffic from NHTSA to set up the ego vehicle’s driving route and the threat vehicle prone to colliding. We further add another layer of complexity by positioning large trucks to obstruct the sensors (LiDAR and cameras) of the ego vehicle, simulating challenging V2V co-perception conditions. Typically, each scenario features two intelligent CAVs, including a collaborating CAV that aids the ego vehicle in detecting potential collision threats.

Scenario visualization. Following NHTSA guidelines, we carefully crafted twelve pre-crash scenarios representing diverse challenging driving conditions. These scenarios are designed to test the limits of vehicle visibility and showcase the efficacy of multi-agent cooperative perception in mitigating visibility constraints and extreme occlusion. Fig. 7 illustrates critical moments before potential crashes across all scenarios, depicting a variety of hazardous driving situations. These include complex interactions such as left and right turn obstacles, straight and merging challenges, unprotected turns, highway merges, and emergency stops. The Appendix provides a detailed specification of these scenarios, ranging from urban intersections to rural roads, each demanding proactive hazard avoidance and adherence to traffic norms by the ego vehicle.

4.2 Evaluation Metrics

We employ a multi-tiered evaluation framework in the OPV2V-C scenario benchmark to comprehensively assess cooperative perception models across several dimensions:

❶ **Model Level:** We utilize Average Precision (AP) at varying Intersection-over-Union (IoU) thresholds as standard metrics to assess 3D detection accuracy within a specified range around the ego vehicle, reporting AP@0.3, @0.5, and @0.7 for each scenario.

❷ **Safety Level:** Critical for any driving system, safety is evaluated through Collision Rate (CR), Time-to-Collision (TTC), and Off-Road (OR) incidents, which provide insights into the vehicle’s ability to avoid collisions and maintain road discipline.

❸ **Efficiency Level:** Operational efficiency is measured by Time-to-Destination (TTD), Average Speed (AS), and Average Route Distance (ARD), quantifying the autonomous system’s performance in achieving its objectives effectively.

❹ **Stability Level:** Stability metrics, including average acceleration (ACC) and average yaw rate (AYR), assess the smoothness and predictability of vehicle movements, enhancing passenger comfort and trust in the autonomous system.

❺ **System Level:** An aggregate score encapsulates overall performance, calculated as a weighted sum of normalized scores from all levels: $OS = \sum_{i=1}^n w_i \times M_i$, where each metric M_i is normalized based on its optimal value: $M_i = m_i / m_i^{max}$, if m_i is the higher the better; else, $= 1 - (m_i / m_i^{max})$, if m_i is the lower the better.

These metrics collectively provide a detailed and nuanced view of the autonomous system’s capabilities, offering insights into its real-world applicability and effectiveness. Each metric has been chosen to reflect crucial aspects of autonomous operation, ensuring that our evaluations mirror the complexities and challenges of real driving scenarios.

5 Experiments

5.1 Experiment Settings

We conducted all the simulation experiments using our closed-loop simulation platform. The reference scenarios directly retrieve 3D bounding boxes from the server (i.e., 100% average precision), then run the entire simulation to get the reference metrics, such as the time-to-collision (TTC), average time spent to complete the route (TS), etc. We evaluated three types of cooperative detection methods: early fusion, intermediate fusion, and late fusion [1], all using the PointPillar [36] backbone for feature extraction. For intermediate fusion, we include two leading models, OPV2V [1] and V2X-ViT [2]. We run simulations for all the compared perception methods using the same configuration to deduce the impact of each detection module on the overall system performance.

Table 1: **Comprehensive diagnostic report of OpenCDA- ∞ simulation performance on OPV2V-Safety benchmark.** We evaluated all the cooperative perception models on all the testing scenarios and reported the metrics. 1) Safety Level. CR: collision rate, TTC: average time-to-collision, SOR: stuck on road, OR: off-road. 2) Efficiency Level. TTD: time-to-destination, AS: average speed, ARD: average route distance. 3) Stability Level: ACC: average acceleration, AYR: average yaw rate. 4) System Level. OS: an overall score that summarizes all the metrics. \uparrow / \downarrow : higher/lower the better.

Method	V2V?	Safety Level			Efficiency Level			Stability Level		Sys. Level OS \uparrow
		CR \downarrow	TTC \uparrow	OR \downarrow	TTD \downarrow	AS \uparrow	ARD \downarrow	ACC \downarrow	AYR \downarrow	
Early Fusion	No	0.500	N/A	0.000	15.50	23.91	83.71	0.295	0.132	0.516
	Yes	0.000	N/A	0.000	16.05	21.75	79.93	0.354	0.125	0.758
Late Fusion	No	0.417	6.31	0.000	15.80	22.53	79.34	0.298	0.132	0.535
	Yes	0.083	5.59	0.083	16.39	20.30	77.12	0.249	0.109	0.610
OPV2V [1]	No	0.417	6.45	0.000	16.55	20.86	81.85	0.278	0.120	0.532
	Yes	0.167	6.64	0.000	18.71	19.98	79.18	0.215	0.103	0.658
V2X-ViT [2]	No	0.583	5.28	0.000	16.42	21.73	81.02	0.293	0.136	0.440
	Yes	0.250	5.67	0.083	18.11	20.88	77.53	0.301	0.101	0.524

5.2 Quantitative Planning Results

Tab. 1 outlines the end-to-end simulation outcomes across different online cooperative perception methods as per evaluation metrics established in Sec. 4.2.

❶ **Safety Level:** we may observe that models without V2V communication consistently report high collision rates despite utilizing advanced planning and trajectory prediction (Sec. 3.4). The *Late*

Table 2: **3D cooperative detection results on in-distribution OPV2V-test dataset.** We show Average Precision (AP) at IoU=0.5. The **boldfaced** and underlined entries indicate the best and second performers for enabling and disabling V2V communication, respectively.

Method	V2V?	Scenario index																Avg.↑
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Early Fusion	No	0.92	0.75	0.91	0.79	0.70	0.93	0.67	0.32	0.28	0.85	0.81	0.72	0.72	0.57	0.65	0.54	<u>0.70</u>
	Yes	0.88	0.91	0.95	0.89	0.71	0.95	0.67	0.43	0.58	0.93	0.80	0.73	0.75	0.64	0.83	0.79	0.78
Late Fusion	No	0.91	0.75	0.95	0.82	0.63	0.82	0.67	0.29	0.35	0.84	0.71	0.72	0.64	0.47	0.65	0.63	0.68
	Yes	0.89	0.87	0.94	0.88	0.69	0.81	0.82	0.38	0.54	0.91	0.65	0.75	0.71	0.52	0.77	0.68	0.74
OPV2V [1]	No	0.92	0.77	0.93	0.74	0.79	0.80	0.78	0.36	0.33	0.82	0.69	0.74	0.76	0.55	0.59	0.57	<u>0.70</u>
	Yes	0.91	0.92	0.94	0.92	0.80	0.90	0.48	0.47	0.48	0.93	0.75	0.84	0.76	0.69	0.81	0.73	0.77
V2X-ViT [2]	No	0.95	0.75	0.95	0.78	0.84	0.91	0.73	0.34	0.50	0.85	0.80	0.75	0.71	0.68	0.73	0.61	0.74
	Yes	0.93	0.92	0.95	0.89	0.86	0.90	0.73	0.46	0.64	0.91	0.86	0.80	0.75	0.67	0.87	0.61	0.80

Fusion and *OPV2V* methods report slightly better but still inadequate CRs, over 40%, infeasible for practical deployment. In stark contrast, the *Early Fusion* method with V2V communication achieves a zero collision rate, significantly enhancing safety across all scenarios. The V2X-ViT model, however, shows a CR of 33.3%, indicating varying performance depending on the fusion method used. Evaluating against OR metrics, only *Late Fusion* and V2X-ViT with V2V communication record a metric score of 0.083; other methodologies report no such violations.

② **Efficiency Level:** Tab. 1 reveals a notable trend: the introduction of V2V communication typically results in a trade-off between safety and efficiency, often reducing Average Speed (AS) and increasing Time-to-Destination (TTD), due to early threat detection and preventive deceleration to avoid potential collision. However, Average Route Distance (ARD) tends to decrease, suggesting more efficient route planning. Moreover, it is worth mentioning that different models may outperform others across different evaluation metrics. As an illustration, while *Early Fusion* with V2V integration achieves an impressive zero CR score, emphasizing its safety, its efficiency level ARD performance doesn't quite match the performance of some other models.

③ **Stability Level:** From our observations, models present distinct behaviors in this domain. Specifically, while both *Early Fusion* and V2X-ViT excel in performance in ACC, *Late Fusion* and *OPV2V* largely enjoy the benefits of cooperative perception. In terms of AYR, the integration of V2V communication facilitates a decline in scores across all fusion methods, hinting at improved stability concerning yaw rate modifications.

④ **System Level:** Our evaluation reveals a significant trade-off between safety and efficiency metrics in single-vehicle mode. Models like *Early Fusion* achieve high Average Speed (AS) scores, indicating efficient route completion but at the cost of higher collision rates (CR). Conversely, *OPV2V* showcases low CR scores but at the expense of the slowest Time-to-Destination (TTD), highlighting a fundamental conflict between safety and efficiency as also noted in prior studies [37, 38]. However, integrating V2V cooperative perception can mitigate these trade-offs. For example, *Early Fusion* with V2V not only maintains low CR but also improves TTD, showcasing the potential of V2V systems to break the limitations of single-vehicle perception. This analysis underscores the necessity for a unified system-level metric that comprehensively evaluates all performance dimensions. The composite Overall Score (OS) metric suggests that *Early Fusion* enhanced with V2V excels in overall system performance, while the popular V2X-ViT model scores lowest within our simulation framework despite claiming high detection capabilities in standard benchmarks.

5.3 Discussions on Detection Results

We then present comparative results using standard 3D object detection performance metrics on both the OPV2V-Test set and our newly introduced OPV2V-Safety set. It is worth noting that, in contrast to prior works like [1, 2] that adopted offline evaluation, we embed these models within our OpenCDA- ∞ simulator for online AP assessment.

As indicated by Tab. 2, V2X-ViT consistently outperforms others, irrespective of V2V communication, aligning with their original study [2]. Interestingly, this result contrasts significantly with our planning-focused outcomes in Tab. 1 where V2X-ViT actually lags behind other approaches in the planning-oriented view. To further understand the root cause, we evaluated the AP scores on the OPV2V-Safety dataset, as in Tab. 3. It may be seen that all detection models, regardless of the fusion strategies, experience a marked drop in AP scores. Specifically, without V2V, OPV2V's AP@0.5 is only 0.24, significantly lower than scores reported in earlier works [1, 2]. Enabled V2V sees the

Table 3: **3D cooperative detection results on the proposed (out-of-distribution) OPV2V-Safety dataset.** We show Average Precision (AP) at IoU=0.5.

Method	V2V?	Scenario index												Avg.↑
		1	2	3	4	5	6	7	8	9	10	11	12	
Early Fusion	No	0.08	0.06	0.11	0.10	0.02	0.00	0.46	0.17	0.26	0.21	0.11	0.31	0.16
	Yes	0.35	0.56	0.16	0.37	0.29	0.29	0.42	0.38	0.53	0.54	0.33	0.43	0.39
Late Fusion	No	0.14	0.10	0.11	0.08	0.01	0.03	0.20	0.15	0.32	0.38	0.14	0.25	0.16
	Yes	0.37	0.43	0.16	0.42	0.33	0.44	0.30	0.41	0.46	0.42	0.23	0.38	0.36
OPV2V [1]	No	0.18	0.08	0.11	0.07	0.00	0.03	0.55	0.23	0.34	0.31	0.26	0.42	0.22
	Yes	0.34	0.52	0.12	0.29	0.31	0.28	0.39	0.36	0.51	0.45	0.42	0.46	<u>0.37</u>
V2X-ViT [2]	No	0.16	0.12	0.08	0.08	0.01	0.03	0.47	0.25	0.40	0.30	0.21	0.38	<u>0.21</u>
	Yes	0.33	0.33	0.18	0.18	0.27	0.16	0.38	0.32	0.54	0.61	0.29	0.54	0.34

straightforward *Early Fusion* leading with a 0.39 AP. Still, these results are rather unacceptably low compared to numbers on the in-domain test set (i.e., OPV2V-test), highlighting the challenging nature of our proposed OPV2V-Safety benchmark. Our findings call for a reevaluation of current V2V perception models and emphasize the necessity for advancements in technologies that ensure safety and reliability in cooperative autonomous driving. This rigorous analysis of detection capabilities within a realistic, dynamic environment reveals critical insights into the limitations and potential improvements for future autonomous vehicle technologies.

6 Concluding Remarks

In this paper, we introduce a comprehensive closed-loop, end-to-end simulation framework called **OpenCDA- ∞** to evaluate V2V cooperative perception systems with a focus on planning-oriented performances beyond detection accuracy. Our framework enriches OpenCDA with functionalities such as online cooperative detection, OpenSCENARIO customization, trajectory prediction, and advanced planning capabilities, enabling online evaluation of the detection model’s impact on downstream planning performance. We also introduced the OPV2V-Safety benchmark, which includes twelve complex scenarios carefully designed to challenge current cooperative systems under severe occlusions and challenging conditions. We provide a suite of evaluation metrics to assess performance across model safety, efficiency, stability, and overall system-level score. Our experiments demonstrate the effectiveness of our simulation framework in providing detailed insights into the diagnosis report of V2V perception models, highlighting their effects on planning-centric metrics like safety and efficiency levels. We hope these contributions mark a significant step forward in advancing the safety and planning-oriented benchmarks and modeling for cooperative driving systems.

Acknowledgment

We would like to thank Dr. Hao Xiang, Dr. Xin Xia, Dr. Xu Han, and Dr. Jiaqi Ma for their valuable support and contributions during the early stage of this research.

References

- [1] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2583–2589. IEEE, 2022. [2](#), [3](#), [4](#), [7](#), [8](#), [9](#), [16](#), [18](#), [21](#), [22](#)
- [2] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *ECCV*, pages 107–124. Springer, 2022. [2](#), [3](#), [4](#), [7](#), [8](#), [9](#), [18](#), [25](#)
- [3] Yiming Li, Ziyang An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: A virtual collaborative perception dataset for autonomous driving. *arXiv preprint arXiv:2202.08449*, 2022. [2](#), [3](#), [4](#), [21](#)
- [4] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. [2](#), [4](#), [15](#)

- [5] Cristina Olaverri-Monreal, Javier Errea-Moreno, Alberto Díaz-Álvarez, Carlos Biurrún-Quel, Luis Serrano-Arriezu, and Markus Kuba. Connection of the sumo microscopic traffic simulator and the unity 3d game engine to evaluate v2x communication-based systems. *Sensors*, 18(12):4399, 2018. 2, 15
- [6] Runsheng Xu, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. Opencda: an open cooperative driving automation framework integrated with co-simulation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1155–1162. IEEE, 2021. 2, 3, 15
- [7] CARLA authors. Openscenario support. https://carla-scenariorunner.readthedocs.io/en/latest/openscenario_support/, 2023. 2, 5, 18
- [8] He Chen, Hongpinng Ren, Rui Li, Guang Yang, and Shanshan Ma. Generating autonomous driving test scenarios based on openscenario. In *2022 9th International Conference on Dependable Systems and Their Applications (DSA)*, pages 650–658. IEEE, 2022. 2
- [9] Wassim G Najm, John D Smith, Mikio Yanagisawa, et al. Pre-crash scenario typology for crash avoidance research. Technical report, United States. National Highway Traffic Safety Administration, 2007. 2
- [10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012. 2
- [11] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 2
- [12] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020. 2
- [13] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 3
- [14] Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13723–13733, 2023. 3, 21
- [15] Letian Wang, Jie Liu, Hao Shao, Wenshuo Wang, Ruobing Chen, Yu Liu, and Steven L Waslander. Efficient reinforcement learning for autonomous driving with parameterized skills and priors. *arXiv preprint arXiv:2305.04412*, 2023. 3
- [16] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, pages 726–737. PMLR, 2023. 3
- [17] Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: end-to-end driving with cooperative perception for networked vehicles. In *CVPR*, pages 17252–17262, 2022. 3, 21
- [18] Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15120–15130, 2024. 3
- [19] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7153–7162, 2020. 3

- [20] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15222–15232, 2021. 3
- [21] Quanyi Li, Zhenghao Mark Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *Advances in neural information processing systems*, 36, 2024. 3
- [22] Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3567–3575. IEEE, 2023. 3
- [23] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *ECCV*, pages 605–621. Springer, 2020. 3
- [24] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *CVPR*, pages 21361–21370, 2022. 3, 16
- [25] Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, et al. V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *CVPR*, pages 13712–13722, 2023. 3
- [26] Qi Chen, Sihai Tang, Qing Yang, and Song Fu. Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 514–524. IEEE, 2019. 3
- [27] Zaydoun Yahya Rawashdeh and Zheng Wang. Collaborative automated driving: A machine learning-based method to enhance the accuracy of shared information. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3961–3966. IEEE, 2018. 3
- [28] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 88–100, 2019. 3
- [29] Yifan Lu, Quanhao Li, Baoan Liu, Mehrdad Dianati, Chen Feng, Siheng Chen, and Yanfeng Wang. Robust collaborative 3d object detection in presence of pose errors. *arXiv preprint arXiv:2211.07214*, 2022. 3
- [30] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, pages 16259–16268, 2021. 3
- [31] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, pages 459–479. Springer, 2022. 3
- [32] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5769–5780, 2022. 3
- [33] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird’s eye view semantic segmentation with sparse transformers. *arXiv preprint arXiv:2207.02202*, 2022. 3
- [34] Runsheng Xu, Chia-Ju Chen, Zhengzhong Tu, and Ming-Hsuan Yang. V2x-vity2: Improved vision transformers for vehicle-to-everything cooperative perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3, 18
- [35] Jinlong Li, Xinyu Liu, Baolu Li, Runsheng Xu, Jiachen Li, Hongkai Yu, and Zhengzhong Tu. Comamba: Real-time cooperative perception unlocked with state space models. *arXiv preprint arXiv:2409.10699*, 2024. 3

- [36] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 7, 17
- [37] Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019. 8
- [38] Zuxin Liu, Hongyi Zhou, Baiming Chen, Sicheng Zhong, Martial Hebert, and Ding Zhao. Constrained model-based reinforcement learning with robust cross-entropy method. *arXiv preprint arXiv:2010.07968*, 2020. 8
- [39] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 17
- [40] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 17
- [41] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 17
- [42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 18
- [43] Wassim G Najm, Raja Ranganathan, Gowrishankar Srinivasan, John D Smith, Samuel Toma, Elizabeth Swanson, August Burgett, et al. Description of light-vehicle pre-crash scenarios for safety applications based on vehicle-to-vehicle communications. Technical report, United States. National Highway Traffic Safety Administration, 2013. 22
- [44] Chejian Xu, Wenhao Ding, Weijie Lyu, Zuxin Liu, Shuai Wang, Yihan He, Hanjiang Hu, Ding Zhao, and Bo Li. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. *Advances in Neural Information Processing Systems*, 35:25667–25682, 2022. 24

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to **[Yes]**, **[No]**, or **[N/A]**. You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? **[Yes]** See Section A.3.1.1.
- Did you include the license to the code and datasets? **[No]** The code and the data are proprietary.
- Did you include the license to the code and datasets? **[N/A]**

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? **[Yes]** See Sec. 1.
 - (b) Did you describe the limitations of your work? **[Yes]** See Appendix A.4.
 - (c) Did you discuss any potential negative societal impacts of your work? **[Yes]** See Appendix A.5.
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? **[Yes]** We have read the ethics review guidelines and ensured that our paper conforms to them.
2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? **[N/A]**
 - (b) Did you include complete proofs of all theoretical results? **[N/A]**
3. If you ran experiments (e.g. for benchmarks)...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **[Yes]** We have included the code, data, and instructions in the supplemental material.
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[Yes]** We have specified all the details in Appendix A.1.2.
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **[N/A]**
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **[Yes]** See Appendix A.1.2.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? **[N/A]**
 - (b) Did you mention the license of the assets? **[Yes]** See Appendix.A.2.4 and the supplemental material.
 - (c) Did you include any new assets either in the supplemental material or as a URL? **[Yes]** See the supplemental material.
 - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[N/A]**
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[Yes]** See Appendix.A.2.2.
5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? **[N/A]**
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? **[N/A]**
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **[N/A]**

Contents

1	Introduction	1
2	Related Work	2
3	<i>OpenCDA-∞: An Online, Closed-loop, End-to-end Simulator</i>	3
3.1	OpenCDA Simulation Platform	3
3.2	Online Cooperative Detection	4
3.3	OpenSCENARIO Add-ons	5
3.4	Trajectory Prediction and Behavior Planning	5
4	The OPV2V-Safety Dataset and Benchmark	5
4.1	Data Protocols	6
4.2	Evaluation Metrics	6
5	Experiments	7
5.1	Experiment Settings	7
5.2	Quantitative Planning Results	7
5.3	Discussions on Detection Results	8
6	Concluding Remarks	9
A	Appendix	15
A.1	Detailed Specifications of OpenCDA- ∞	15
A.1.1	The OpenCDA Simulation Platform	15
A.1.2	The OpenCOOD Cooperative Detection Toolkit	16
A.1.3	OpenSCENARIO Add-ons	18
A.1.4	Trajectory Prediction	19
A.1.5	Robust Planning with Prediction	19
A.1.6	Robust Behavior Planning with Trajectory Prediction	19
A.2	Detailed Specifications of OPV2V-Safety	21
A.2.1	Design Details	21
A.3	Related works	21
A.4	Data Analysis and Visualization	22
A.4.1	Specifications of Scenarios	22
A.4.2	Evaluation Metrics	22
A.4.3	Data License	24
A.5	Additional Experiments	25
A.5.1	Impact of Communication Latency	25
A.6	Limitations	25
A.7	Potential Negative Societal Impacts	26

A Appendix

A.1 Detailed Specifications of OpenCDA- ∞

In this section, we will detail the specifications of our proposed OpenCDA- ∞ simulation and benchmarking platform for the planning-oriented, safety-critical evaluation of cooperative perception models in a systematic approach.

A.1.1 The OpenCDA Simulation Platform

OpenCDA [6] is an open-source co-simulation-based research and engineering framework tailored for prototyping, developing, and testing cooperative driving automation (CDA) systems. Integrating automated driving simulation tools such as CARLA [4] and SUMO [5], OpenCDA provides a versatile environment to evaluate various CDA applications. OpenCDA supports a range of functionalities, including perception, localization, planning, control, and V2X communication. As illustrated in Fig. 4, OpenCDA organically combines multiple core components: simulation tools, a Python-based Cooperative Driving Automation (CDA) system, and a comprehensive scenario manager, enabling researchers to simulate complex cooperative driving scenarios with high flexibility.

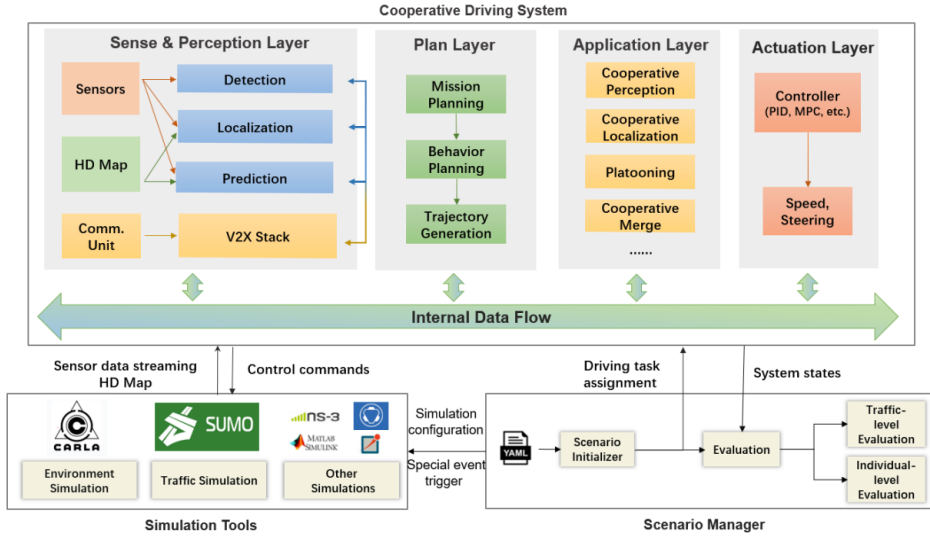


Figure 4: **The overall system design of OpenCDA.** The full-stack software of the designed open-source cooperative driving automation (OpenCDA) system interacts with simulation tools to simulate and test the system-level performance in pre-defined scenarios in Carla Towns and Culver City.

Simulation Tools Integration. OpenCDA employs CARLA [4], an open-source driving simulator powered by Unreal Engine, known for its high-quality rendering capabilities and versatile server-client architecture. While CARLA excels in detailed driving simulations, its limitations in managing large-scale traffic are further mitigated by our further integration of SUMO [5], a comprehensive traffic simulator adept at modeling realistic traffic behavior and scenarios. This dual-tool approach allows researchers to choose between detailed driving simulations with CARLA, broad traffic simulations with SUMO, or a co-simulation approach, to achieve complex, on-demand scenario and traffic simulation for each specific requirement.

Scenario Management. The scenario manager in OpenCDA is a sophisticated module segmented into four primary components: the configuration file, initializer, event trigger, and evaluation functions. It blends static elements (e.g., road structures and infrastructure defined by CARLA’s assets) with dynamic elements (e.g., traffic flow and weather conditions controlled via YAML files). The configuration loader parses these files to guide the simulation, then designating tasks for the respective Connected and Autonomous Vehicles (CAVs). This system allows for the simulation of complex scenarios, including rare events, and enables performance evaluations at both the individual vehicle level in CARLA and at the broader traffic level in SUMO.

Cooperative Driving Automation System: OpenCDA’s cooperative driving system, structured in modular layers, leverages CARLA and SUMO through a streamlined API to perform cooperative driving tasks. Sensors mounted on CAVs collect raw environmental data, which is processed through sequential layers: sensing, perception, planning, and actuation. This architecture supports simulations of both cooperative systems and single vehicle intelligence, facilitating mixed-traffic scenarios. The cooperation is primarily activated in the application layer, where CAVs exchange critical data such as positions and intents through V2X communication, aligning on cooperative strategies like platooning. This layer’s protocols dynamically adapt based on cooperative needs, such as enhancing object detection through shared sensor data fusion.

Modularity and Customization. A distinctive feature of OpenCDA is its modular framework, where each layer comes with default protocols and algorithms that can be easily replaced with advanced ones. This modularity not only allows comprehensive evaluation of the entire CDA system but also facilitates the comparison of individual algorithms within a consistent framework. The built-in algorithms, including those for cooperative platooning, serve as reproducible baselines to provide a valuable reference for emerging research.

A.1.2 The OpenCOOD Cooperative Detection Toolkit

The aforementioned OpenCDA platform does not support running cooperative detection models on the fly. Instead, it directly retrieves bounding boxes from the server, bypassing the need to run any sophisticated detection models actively within the simulation environment. This architectural choice, while efficient, presented limitations, especially for researchers and developers keen on investigating the immediate impacts of cooperative detection algorithms on autonomous driving behaviors.

The OpenCOOD [1] Cooperative Detection Toolkit offers a comprehensive suite of tools and algorithms optimized for collaborative perception. Fig. 5 visualizes the visualization of the bounding box overlay on top of the LiDAR point clouds and camera images. It mainly supports offline cooperative detection model training and testing, based on the pre-exported simulation data generated by simulation platforms like OpenCDA. It also supports training and evaluating real-world datasets like DAIR-V2X [24] if configured properly.

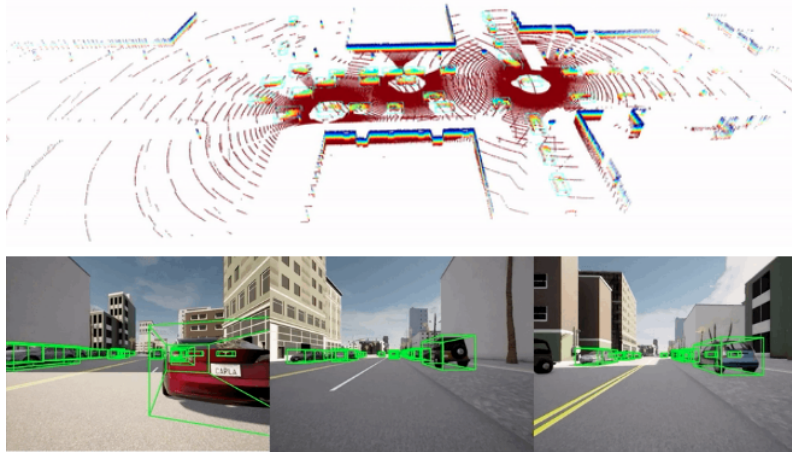


Figure 5: **Demo of OpenCOOD Cooperative Detection Toolkit.** It offers a comprehensive benchmarking platform for evaluating cooperative detection models.

The OpenCOOD framework adopts a system design that considers the V2X/V2V perception problem a multi-agent perception system, where a number of agents of different types (e.g., smart infrastructure or connected and automated vehicles (CAV)) communicate, perceive, and share information/features with each other. To simulate the real-world scenarios, OpenCOOD assumes that all the agents have imperfect locations (e.g., subject to location error) and there exists an uncertain time delay during communication (by simulating a feature transmission delay). The ultimate design goal is to develop a performant and robust V2X/V2V feature fusion module that capably generates accurate bonding boxes under the above-mentioned noise. As shown in Fig. 6, the OpenCOOD framework

consists of five major components: 1) V2X metadata sharing, 2) LiDAR feature extraction, 3) feature compression & sharing, 4) V2X fusion network, 5) a detection head.

❶ V2X metadata sharing. In the initial collaboration stage, each agent within the communication range shares metadata such as poses, extrinsics, and agent type. One agent is selected as the ego vehicle to construct a V2X graph, where nodes represent CAVs or infrastructure and edges represent directional V2X communication channels. OpenCOOD assumes metadata transmission is well-synchronized. Upon receiving the ego vehicle’s pose, connected agents project their own LiDAR point clouds to the ego vehicle’s coordinate frame before feature extraction.

❷ Feature extraction: OpenCOOD implemented various LiDAR feature extractors: PointPillar [36], Pixor [39], VoxelNet [40], and SECOND [41]. By default, the anchor-based PointPillar method is recommended to extract visual features from point clouds due to its low inference latency and optimized memory usage. Raw point clouds are converted into a stacked pillar tensor, scattered to a 2D pseudo-image, and fed into the PointPillar backbone. The backbone extracts feature maps $F_i^t \in \mathbb{R}^{H \times W \times C}$, representing agent i ’s features at time t_i with height H , width W , and channels C .

❸ Compression and sharing: To reduce transmission bandwidth, a series of 1×1 convolutions progressively compresses the feature maps along the channel dimension. The compressed features, with size (H, W, C') (where $C' \ll C$), are transmitted to the ego vehicle and projected back to (H, W, C) using a coupled 1×1 convolution decoder. Due to inevitable time gaps between data capture by connected agents and feature reception by the ego vehicle, features are often temporally misaligned. To correct this delay-induced spatial misalignment, OpenCOOD uses a spatial-temporal correction module (STCM), which employs a differential transformation and sampling operator Γ_ξ to spatially warp the feature maps. An ROI mask prevents the network from focusing on padded zeros caused by spatial warp.

❹ V2X/V2V fusion network: The intermediate features $H_i = \Gamma_\xi(F_i^t) \in \mathbb{R}^{H \times W \times C}$ aggregated from connected agents are fed into the V2X/V2V fusion network, the core component of OpenCOOD, for inter-agent and intra-agent feature fusion. High-resolution feature maps are maintained throughout the network, as the absence of high-definition features significantly impairs object detection performance.

❺ Detection head: After receiving the final fused feature maps, two 1×1 convolution layers are applied for box regression and classification. The regression output is $(x, y, z, w, l, h, \theta)$, denoting the position, size, and yaw angle of predefined anchor boxes. The classification output is the confidence score of being an object or background for each anchor box. Smooth L_1 loss is used for regression and focal loss for classification.

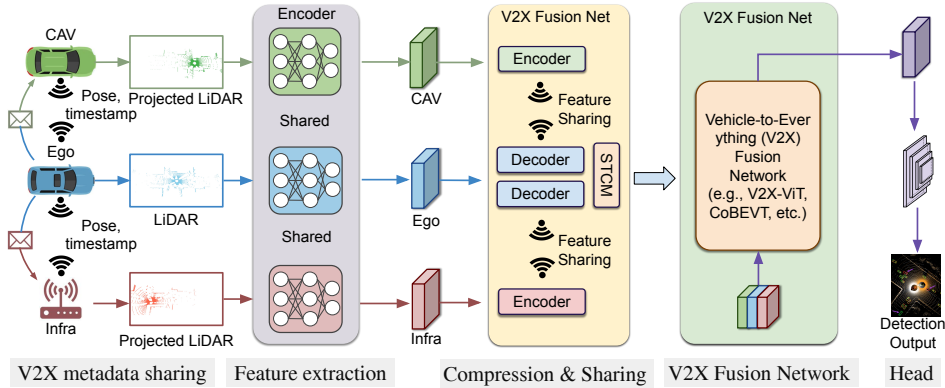


Figure 6: The system design of OpenCOOD detection architecture.

This framework effectively addresses the challenges of V2X perception, enabling robust and efficient perception in autonomous driving systems through advanced cooperative fusion strategies. We have included several baseline and state-of-the-art models to conduct the experiments on OpenCDA- ∞ :

1. **No Fusion:** only uses the ego vehicle’s own LiDAR point cloud without collaboration.
2. **Early Fusion:** directly aggregates the raw LiDAR point clouds and learns fusion on top of the raw data, requiring the highest transmission bandwidth as the raw data is large in size.

3. **Late Fusion:** receives all the final detection results and applies non-maximum impression to produce the final ego results. This method requires the least transmission bandwidth as it only requires transmitting the bounding box axis.
4. **OPV2V [1] (intermediate fusion):** transmits intermediate neural features (e.g., PointPillar features) to ego car where ego applies an attention model to fuse them.
5. **V2X-ViT [2, 34] (intermedia fusion):** transmits intermediate neural features where the ego applies a V2X-ViT transformer to fuse them.

Implementation details. All the models were trained on the OPV2V training set [1], where the set splits are 6,764/1,981/2,719 frames. The number of vehicles is between 2 and 7 for the entire dataset. During training, a random CAV was selected as the ego vehicle; during the test phase, a fixed ego was defined for a fair comparison. The communication range for each agent was set as 70 meters, ignoring CAVs beyond this radius range. The voxel resolution for the PointPillar backbone is 0.4 meters in both height and width. By default, the feature transmission compression rate is 32 for all the compared intermediate fusion methods. We follow the original papers to set up the hyperparameters for each specific model. The Adam optimizer [42] was employed for training, with an initial learning rate of 0.001, steadily decaying every 10 epochs by a factor of 0.1. We trained all the models using the Tesla V100, and all the model training took about a week to finish.

Integration into OpenCDA- ∞ . We have made further efforts to bridge the gap by amalgamation of cooperative detection models implemented in OpenCOOD into OpenCDA. Specifically, by seamlessly compiling OpenCOOD [1] as an additional MLManager component, OpenCDA can now not only run a diverse array of cooperative detection models on the fly but also allow the outputs of these models to directly steer the planning and decision-making processes of its autonomous agents. This enriched feature makes it possible to investigate the influence of state-of-the-art cooperative perception models at a system level, which can more faithfully simulate real-world scenarios. With this new module, the OpenCDA- ∞ would require at least 8GB GPU memory to run the simulation to capably load the OpenCOOD models for online detection.

A.1.3 OpenSCENARIO Add-ons

By default, OpenCDA utilizes the built-in CARLA traffic manager to simulate vehicle dynamics. During scenario initialization, routes are computed based on initial spawn and destination positions. Additionally, auxiliary vehicles are spawned randomly within predefined ranges and probability distributions, resulting in non-deterministic behavior in each run. However, this setup offers limited granular control over individual actor behavior, making specific scenario generation challenging.

OpenSCENARIO [7] is a standardized XML-based language designed for defining driving scenarios, providing a structured approach to create reproducible, configurable, and complex simulations. It enables scripting scenarios ranging from simple straight-road driving to complex urban environments with multiple dynamic actors. The framework supports encoding high-level traffic rules and participant behaviors. ScenarioRunner is an extension to CARLA that facilitates the execution and evaluation of scenarios described using OpenSCENARIO or a Python interface. It translates high-level OpenSCENARIO XML-based definitions into sequences of actions and events within the CARLA simulation, effectively bridging the gap between text-defined scenarios and the CARLA simulation environment.

Our study presents twelve reproducible and challenging scenarios developed with OpenSCENARIO, wherein cooperative perception proves more beneficial than no collaboration but still struggles due to the complexity of the scenarios. Furthermore, we introduce a programming paradigm demonstrating how OpenCDA can consume scenarios defined by OpenSCENARIO. The below listing illustrates an exemplary XML configuration file defining an OpenSCENARIO scene specification.

```
<?xml version="1.0"?>
<scenarios>
  <scenario name="Scenario_4" type="Scenario_4" town="Town03"
    enable_cav="True"> # Define the scene.
    <ego_vehicle x="23.8" y="3.7" z="0.3" yaw="0" model="vehicle.
nissan.patrol" /> # Ego vehicle
    <other_actor x="50.8" y="3.7" z="-500" yaw="0" model="vehicle.
dodge.charger_2020" /> # Other actors
```

```

    <other_actor x="58.8" y="7.4" z="-500" yaw="0" model="vehicle.
dodge.charger_2020" /> # Other actors
    <other_actor x="58.8" y="3.7" z="-500" yaw="0" model="vehicle.
dodge.charger_2020" rolename="cav_1" color="(0,0,255)" /> # CAV
vehicle
    <weather cloudiness="0" precipitation="0"
precipitation_deposits="0" wind_intensity="0" sun_azimuth_angle="0"
" sun_altitude_angle="75" /> # Define the scene specs.
</scenario>
</scenarios>

```

Listing 1: An exemplary XML config file to define an OpenSCENARIO.

A.1.4 Trajectory Prediction

In its current architecture, OpenCDA lacks inherent support for trajectory prediction. This limitation significantly constrains our ability to understand how the predictable future trajectories of other vehicles influence subsequent planning activities, which are critical in automated driving systems. To better emulate real-world traffic scenarios and simulate driver behaviors, this study introduces a trajectory prediction module supporting various commonly used behavior models. This addition aims to enhance the realism and complexity of the simulation environment.

We implement five prediction models, each designed to capture specific driving behaviors, thereby facilitating nuanced simulations. These models include:

❶ **Constant Velocity:** This model is suitable for scenarios involving consistent traffic flow on highways or steady-state vehicle motion in urban environments with light traffic, where vehicle speed remains relatively constant. The corresponding formula for position over time is expressed as $x = vt$, where x denotes displacement, v is constant velocity, and t is time.

❷ **Constant Acceleration:** Ideal for scenarios where vehicles are continuously accelerating or decelerating, such as entering or exiting highways. The motion is mathematically described by $v = u + at$, where v is the final velocity, u is the initial velocity, a is constant acceleration, and t is time.

❸ **Constant Speed and Yaw Rate:** This model applies to situations where vehicles are moving at a constant speed and direction, such as when a vehicle is steadily cornering or changing lanes at a fixed rate. It can be represented as $\theta = \omega t$, where θ is the yaw angle, ω is the constant yaw rate, and t is time.

❹ **Constant Acceleration and Yaw Rate:** Used in more complex scenarios where a vehicle is accelerating or decelerating while simultaneously changing direction at a constant rate, such as taking an exit ramp at varying speeds. This scenario combines the equations $x = vt$ and $\theta = \omega t$ to capture both linear and angular movements.

❺ **Physics Oracle Model:** This highly precise model is employed in the most complex scenarios requiring predictions of various driving behaviors, including sudden stops, emergency maneuvers, and high-speed chases. It is an ensemble of various physical laws and principles, encompassing both linear and rotational dynamics to capture the full range of possible vehicle behaviors in response to dynamic environments.

These models collectively advance the realism and fidelity of the simulation environment, providing a more comprehensive tool for understanding and predicting vehicle behaviors in automated driving systems.

A.1.5 Robust Planning with Prediction

A.1.6 Robust Behavior Planning with Trajectory Prediction

OpenCDA designs planning behavior using a rule-based finite-state machine system, conditional on specific traffic dynamics. However, without the presence of the prediction module, we have observed that the default planning behavior falls short in handling numerous challenging scenarios. These include intersections with orthogonal traffic, complex lane merging, and vehicles abruptly emerging

Algorithm 1 Robust Behavior Planning Algorithm.

```
1: Initialize the Carla world with YAML and XML configs
2: Get the current vehicle state and sensor data
3: Calculate the global route
4: while vehicle is running and far from destination do
5:   Update vehicle state and sensor data
6:   Run online cooperative perception models
7:   Do collision check for observed vehicles
8:   Run trajectory prediction for actors
9:   Do collision check for predicted trajectories
10:  if red light detected ahead then
11:    Brake immediately
12:  end if
13:  if in intersection then
14:    Decelerate
15:  end if
16:  if slow vehicle detected in front then
17:    Do collision check for overtaking
18:    if overtake allowed and safe to overtake then
19:      Perform overtaking
20:    else
21:      follow the front vehicle
22:    end if
23:  else if safe to push then
24:    Acceralte to temporary waypoints
25:  else
26:    Continue current action or follow the trajectory
27:  end if
28:  Update control commands to actuators
29: end while
```

from blind spots. These findings highlight the limitations of the current planning algorithm. Thus, we are implementing new planning components to accommodate the existence of predicted trajectories.

Algo. 1 depicts the detailed robust behavior planning algorithm in our proposed simulation platform, OpenCDA- ∞ . The system consists of several major states: calculating a global route, performing lane changes, formulating and executing temporary routes, overtaking, following the front vehicle, and decelerating when the ego vehicle is proximate to other obstacles. Upon integrating the proposed prediction model, future trajectories of visible actors become predictable. Consequently, we designed a mechanism to determine if the ego vehicle will collide with other obstacle vehicles within the next k seconds. Given the uncertainty of velocity and traffic conditions, we propose configurable parameters that delineate the range of possible future positions, as detailed in Fig. 2 of main paper, to account for variations in locations.

Specifically, suppose the planned waypoints of the ego car are $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}$, and the predicted trajectory of a threat car is $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K\}$. We run a collision check at a uniform time series $T = \{t_1, t_2, \dots, t_K\}$ and check the minimum possible distance between the two vehicles at future locations within a given scanning window that accounts for prediction error:

$$L = \min_{t \in T} \min_{r \in [t-\tau, t+\tau]} \|\mathbf{x}_r - \mathbf{y}_r\|_2. \quad (2)$$

If L is less than some predefined distance, then there is a potential collision, and the planning algorithm will adjust accordingly.

This enhancement in trajectory prediction and behavior planning bridges the gap between the simulation environment and real-world driving behavior. The simulation platform can now handle more complex scenarios, such as those defined in our proposed OPV2V-Safety benchmarks.

A.2 Detailed Specifications of OPV2V-Safety

In this section, we will detail the proposed OPV2V-Safety dataset, a supplementary set of challenging scenarios for evaluating the safety of cooperative perception models in our proposed closed-loop OpenCDA- ∞ simulation benchmark.

A.2.1 Design Details

Simulation Platform. We utilize our proposed OpenCDA- ∞ , the enhanced co-simulation platform for V2X/V2V cooperative automation research based on CARLA, as our simulation toolkit. Integrated with the OpenSCENARIO add-on, we customized the behaviors of all the other actors within the simulation and used the XML config file to control the behavior of the ego vehicle and CAVs. All the detection results from the compiled OpenCOOD models (e.g., OPV2V, V2X-ViT) are loaded prior to simulation and inference at per-frame to generate bounding boxes on the fly. These detection results are employed as input for the downstream trajectory prediction and robust planning. After the simulation has concluded, we record the evaluation metrics and generate our comparison report.

Scenario Setting. We generate the scenarios based on the eight default towns, which are directly available in CARLA, for easier reproduction. In each scenario, we follow the safety-critical pre-crash traffic from NHTSA to set up the driving route of the ego vehicle and the vehicle prone to colliding with it. We also spawn multiple large trucks as obstacles to block the sensors (LiDAR and cameras) in the ego car, making the scenarios even more challenging. To tailor them in the context of V2V co-perception, we initiate another CAV in collaboration with ego to capably ‘see’ the collision vehicle ahead of the ego vehicle. Our dataset has an average of two intelligent vehicles in each scenario.

Sensor Configuration. Similar to previous works [1, 3], we configured each CAV to be equipped with 4 RGB cameras facing front, back, left, and right respectively, so that they collectively cover the whole 360° panoramic view. Each camera has 110° field-of-view (FOV) and reads out 800×600 RGB frames. A 64-channel LiDAR with 1.3M points per second, 120-meter capturing range is mounted on the middle of the vehicle roof. We record LiDAR point clouds and camera frames at 10 Hz and save the corresponding positional data and timestamp as additional metadata.

A.3 Related works

There exist two highly relevant end-to-end benchmarks and datasets: ReasonNet [14] and Coopernaut [17]. Both ReasonNet and Coopernaut are significant contributions to the field, each with unique approaches to addressing challenges in cooperative perception and autonomous driving. Here we discuss the striking difference of our proposed framework as compared to theirs. ReasonNet focuses on global and temporal reasoning for handling occlusions and predicting the future behavior of objects, which is crucial for navigating complex urban environments. However, we’d like to highlight that our work fundamentally differs from ReasonNet: ReasonNet leverages global and temporal reasoning to solve occlusion problems, while our OpenCDA- ∞ offers a unique contribution to employ multi-agent cooperative/V2V perception, where multiple vehicles are connected to share view sights within a collaborative network, for obstacle handling. The things in common are both to handle severe occlusion, but our methodologies and use cases are strikingly different. Additionally, our OPV2V-Safety dataset specifically targets pre-crash situations, adding another layer of stress testing under extreme conditions that complement the scenarios addressed by ReasonNet. We have designed the scenarios in such a way that without connected vehicles, the crash will likely happen, but with connected vehicles, the ego car will be able to observe the occluded regions, thus yielding smoother and better planning results. Coopernaut introduces an end-to-end learning model using cross-vehicle perception via V2V communications, with a focus on vision-based cooperative driving. It employs the AUTOCASIM framework to evaluate the model in challenging, accident-prone scenarios. Coopernaut focuses on imitation learning-based planning algorithms for end-to-end driving. In contrast, our OpenCDA- ∞ supports both modular components (that are widely deployed by the industry), as well as end-to-end approaches. Empowered by OpenCDA, our frameworks feature more advanced mobility features like platooning. Lastly, OpenCDA- ∞ provides a comprehensive platform to evaluate the system-wide impacts of these perception models on autonomous driving safety and performance.

A.4 Data Analysis and Visualization

Inspired by the guidelines provided by the National Highway Traffic Safety Administration (NHTSA) [43], we carefully crafted 12 pre-crash scenarios, representing challenging driving conditions. To push the envelope further, we introduced additional obstacles in each scenario, simulating situations where a single vehicle’s visibility is critically hindered, making it vulnerable to accidents. This serves as a foundation to explore the potential of multi-agent cooperative perception in ameliorating these visibility issues.

Figure Fig. 7 presents the pivotal moments just before potential crashes for all 12 scenarios, encapsulating a diverse range of daily hazardous driving situations. In particular, our designs include: 1) Left turn obstacles A/B/C, 2) Right turn obstacle A, 3) Straight obstacle A, 4) Merging obstacle A/B, 5) Unprotected left turn A, 6) Highway merging A, 7) Lane-crossing turn A, 8) Zigzagging A, and 9) Sudden stopping A. A comprehensive breakdown of these scenarios can be found in Tab. 4. The scenarios span various road configurations, including 4-way intersections, T-intersections, straight segments, midblocks, entrance ramps, highways, rural roads, and more. In each setting, the ego vehicle embarks on a pre-set path, relying on planning algorithms implemented in OpenCDA to react to unforeseen road events. Adhering to traffic norms and proactively avoiding potential hazards is paramount for the ego vehicle.

A.4.1 Specifications of Scenarios

The detailed descriptions and specifications of our carefully curated scenarios are presented in Tab. 4. It should be noted that we have pre-designed a larger space of scenarios, but have conducted pre-screening to filter out non-interesting scenarios. The remaining ones in OPV2V-Safety can indeed challenge existing cooperative perception systems, as shown in Tab. 1 of the main paper. We should note that we don’t use any personal or sensitive information as our simulation are purely based on Carla simulation using OpenCDA- ∞ . We also did not include any pedestrians in our benchmark.

A.4.2 Evaluation Metrics

Here we detail the evaluation metrics as used in our OPV2V-Safety benchmark, as supplementary to our main paper. Specifically, we evaluate the performance on four levels: **Model level**, **Safety level**, **Efficiency level**, and **Stability level**. Finally, we propose a new **System level** overall score method to summarize the overall performance as a weighted sum of all the evaluation metrics.

❶ **Model Level:** The de facto evaluation metrics to compare 3D detection models are Average Precision (AP) at different Intersection-over-Union (IoU) thresholds. These metrics merely assess the accuracy and robustness of the model in detecting objects/vehicles in 3D space. Following [1], detection performance is evaluated in the range of $x \in [-140, 140]m$, $y \in [-40, 40]m$ near the ego vehicle. The broadcast range is 70 meters, and messages beyond this communication range will be discarded by ego. We report AP@0.3, @0.5, and @0.7 for each individual scenario respectively.

❷ **Safety Level:** Ensuring safety remains critical for any automated driving system. To effectively gauge safety, we consider two key metrics:

- **Collision Rate (CR):** This metric represents the frequency of collisions encountered by the vehicle, typically expressed as a ratio of collisions to total driving time or distance. A lower CR indicates safer driving behavior.
- **Time-to-Collision (TTC):** TTC quantifies the time it would take for two vehicles to collide if they continue at their current speed and trajectory. A smaller TTC suggests a higher risk situation. Monitoring this metric helps in determining how often an autonomous system comes close to a potential collision.
- **Off-Road (OR):** Measures instances when the vehicle unintentionally departs from the designated roadway, indicating potential control or perception failures. Any non-zero OR is generally considered undesirable in automated driving.

❸ **Efficiency Level:** Metrics in this category gauge the operational efficiency of the autonomous system. We include metrics like 1) Time-to-Destination (TTD), which evaluates the time taken for an autonomous vehicle to reach its destination, 2) Average Speed (AS) which records the average speed magnitude when completing the route, and 3) Average Route Distance (ARD) that calculates

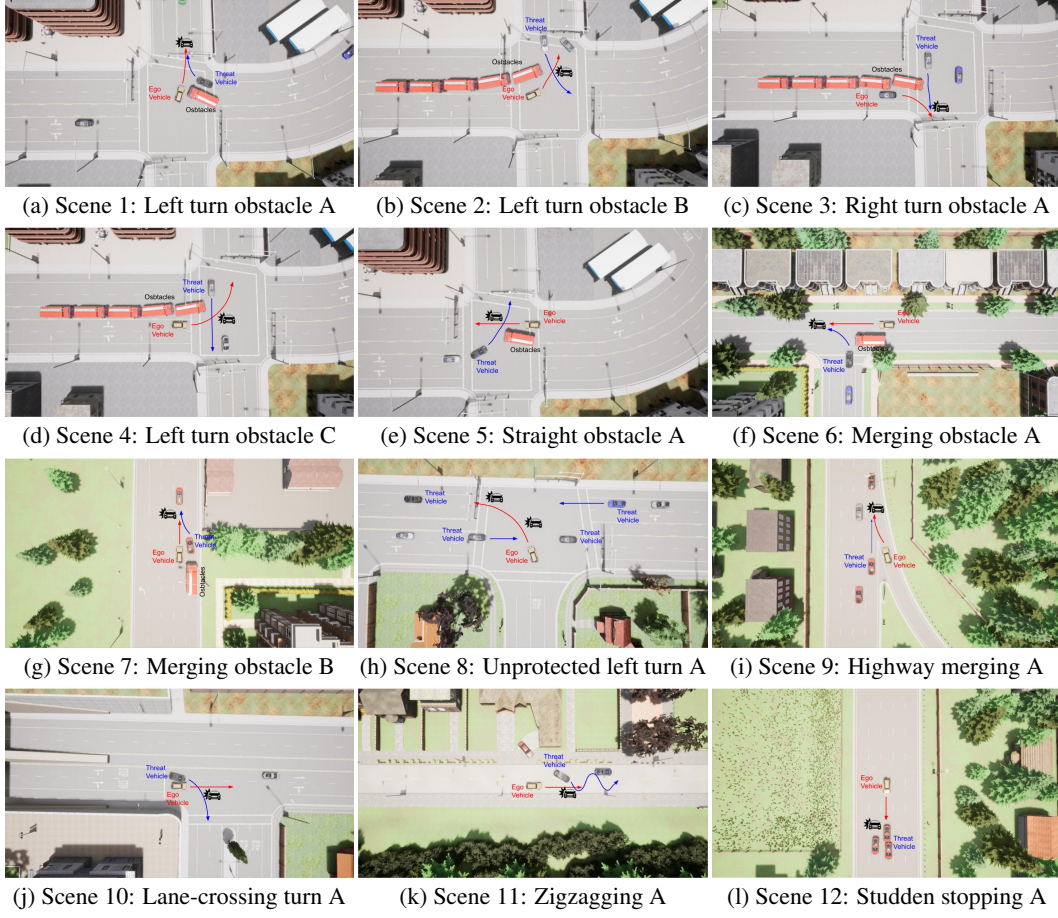


Figure 7: The visualization of potential pre-crash moments in the OPV2V-Safety Benchmark.

the overall route distance in meters. These metrics are effective tools to compare the efficiency of the autonomous driving algorithms during simulation.

④ Stability Level: The stability level metrics are an indicator of the driving skills of AD models. Ensuring that an autonomous vehicle operates smoothly and predictably not only heightens passenger comfort but also augments the trust that other road users place in autonomous systems. We follow existing works and consider two metrics accordingly: average acceleration (ACC), and average yaw rate (AYR). The average acceleration gauges the consistency in speed adjustments, effectively preventing abrupt stops or rapid accelerations that can be jarring for occupants. On the other hand, the average yaw rate is a testament to the vehicle’s steadiness during turns, ensuring that lane changes and cornering are executed with precision and grace. Collectively, these metrics provide a holistic assessment of the vehicle’s ability to maneuver safely, confidently, and comfortably in diverse driving conditions.

⑤ System Level: This level amalgamates metrics from the preceding tiers to produce a comprehensive performance indicator. By employing a weighted summation of these metrics, we gain an integrated perspective of the autonomous system’s efficacy across diverse conditions. The weighting coefficients can be tailored to align with the unique criteria and emphases of any given evaluation. Specifically, we calculate the overall score as:

$$OS = \sum_{i=1}^n w_i \times M_i, \quad (3)$$

where M_i is the normalized metric score for i – th evaluation metric:

$$M_i = \begin{cases} m_i/m_i^{max} & \text{if } m_i \text{ is the higher the better} \\ 1 - (m_i/m_i^{max}) & \text{if } m_i \text{ is the lower the better} \end{cases} \quad (4)$$

Table 4: Detailed descriptions of carefully designed scenarios in the OPV2V-Safety dataset.

SCENARIO	Len (s)	Description
1: Left turn obstacle A	17.5	The ego vehicle attempts a left turn at an intersection, with its line of sight to opposite lanes obstructed by a truck, masking potentially hazardous oncoming vehicles from the right.
2: Left turn obstacle B	17.2	The ego vehicle makes a left turn at an intersection, whose view to the left is blocked by obstacle trucks, while a threat vehicle initiates a left turn from its perpendicular left.
3: Right turn obstacle A	13.6	The ego car begins a right turn at an intersection with view blocked by trucks, hindering the view of a potentially hazardous vehicle approaching from a perpendicular direction.
4: Left turn obstacle C	23.8	The ego vehicle initiates a right turn at an intersection, with its vision hampered by trucks, while a threatening vehicle approaches straight on from a perpendicular path.
5: Straight obstacle A	15.1	The ego vehicle advances straight through an intersection, but its forward view is blocked by obstacle trucks as a vehicle from the opposing direction attempts a left turn.
6: Merging obstacle A	21.3	Driving straight on a T-intersection, the ego vehicle faces a challenging situation as a vehicle from a perpendicular direction attempts a left merge, all while trucks block its view.
7: Merging obstacle B	16.9	As the ego vehicle cruises on a straight segment, an abrupt merge attempt by a vehicle from an adjacent lane takes place, with the ego’s situational awareness hampered by trucks.
8: Unprotected left turn A	15.1	The ego vehicle, aiming for a left merge on a T-intersection, has to navigate through a chaotic mix of bidirectional traffic.
9: Highway merging A	18.8	Approaching the main lanes of a highway via an entrance ramp, the ego vehicle confronts fast-paced traffic densely packed with vehicles.
10: Lane-crossing turn A	20.3	As the ego vehicle moves straight ahead on a T-intersection, an erratic vehicle from the left lane suddenly cuts across, aiming to access a perpendicular lane.
11: Zigzagging A	15.9	The ego vehicle maintains a straight trajectory when a neighboring vehicle engages in a perilous double lane-change, switching lanes back and forth, typically without signaling.
12: Sudden stopping A	19.0	While the ego vehicle travels on a straight path, the leading vehicle unexpectedly halts, posing an immediate rear-end collision risk.

Table 5: Constants and weights used to compute the overall score.

Symbol	Safety Level			Efficiency Level			Stability Level	
	CR↓	TTC↑	OR↓	TTD↓	AS↑	ARD↓	ACC↓	AYR↓
m_i^{\max}	1	8	0.1	20	25	85	0.5	0.15
w_i	0.495	0.099	0.099	0.05	0.05	0.05	0.02	0.02

where m_i^{\max} is a constant suggesting the maximum allowed value of this metric. We follow prior works [44] and set the constants and weights used to calculate the overall metrics as shown in Tab. 5.

A.4.3 Data License

The dataset will be released under the MIT License (MIT):

The MIT License (MIT)

Copyright (c) 2024: OpenCDA-Loop Team.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

A.5 Additional Experiments

A.5.1 Impact of Communication Latency

An essential aspect of autonomous driving systems is their real-time responsiveness, especially when evaluating detection accuracy, safety, and other critical metrics. Communication latency can introduce inaccurate coordinate transformations and delayed sensing information. Failure to properly handle these challenges can make the system vulnerable [2]. To address this, we investigated the impact of time delays in communication, focusing on a range of delays from 0 to 400 milliseconds. For our case study, we utilized the balanced model OPV2V, which achieves top-tier performance regardless of V2V communication status. As illustrated in Fig. 8(a), detection accuracy (measured as AP0.5) significantly declines with increased time delay, showing a 51% reduction in performance with a 400 ms delay compared to no delay.

In terms of safety metrics, the reduced detection accuracy does not correlate straightforwardly with degraded collision risk (CR) and time-to-collision (TTC). Interestingly, CR initially decreases and then increases at 100 ms and 300 ms delays, respectively. TTC fluctuates without a clear pattern, showing no consistent relationship with time delay. The occurrence rate (OR) spikes at 200 ms delay but remains steady at higher latencies. Efficiency metrics, such as time to detection (TTD), average speed (AS), and average route deviation (ARD), remain relatively stable and are minimally impacted by time delay. Regarding stability, acceleration (ACC) shows random effects, while average yaw rate (ARY) increases with delays up to 300 ms before dropping at 400 ms.

Overall, the system-level operational safety (OS) metric exhibits a general declining trend as time delay increases from 0 to 400 ms. However, the OS score surprisingly improves at 100 ms latency. This suggests that while minor delays in V2V communication can significantly reduce the detection accuracy of a cooperative model, they may not detrimentally affect overall planning performance when evaluated from a system perspective. This finding underscores the significance of our novel planning-oriented benchmarking pipeline for cooperative perception, which offers a more holistic view of testing and evaluation. It provides critical insights into the safety and operational performance of V2V algorithms, emphasizing the importance of comprehensive system-level assessments.

A.6 Limitations

Although our proposed OPV2V-Safety benchmark includes a variety of challenging scenarios, it may still lack coverage of some critical edge cases and rare events that could significantly impact safety. Real-world driving involves a vast array of unpredictable situations that are challenging to

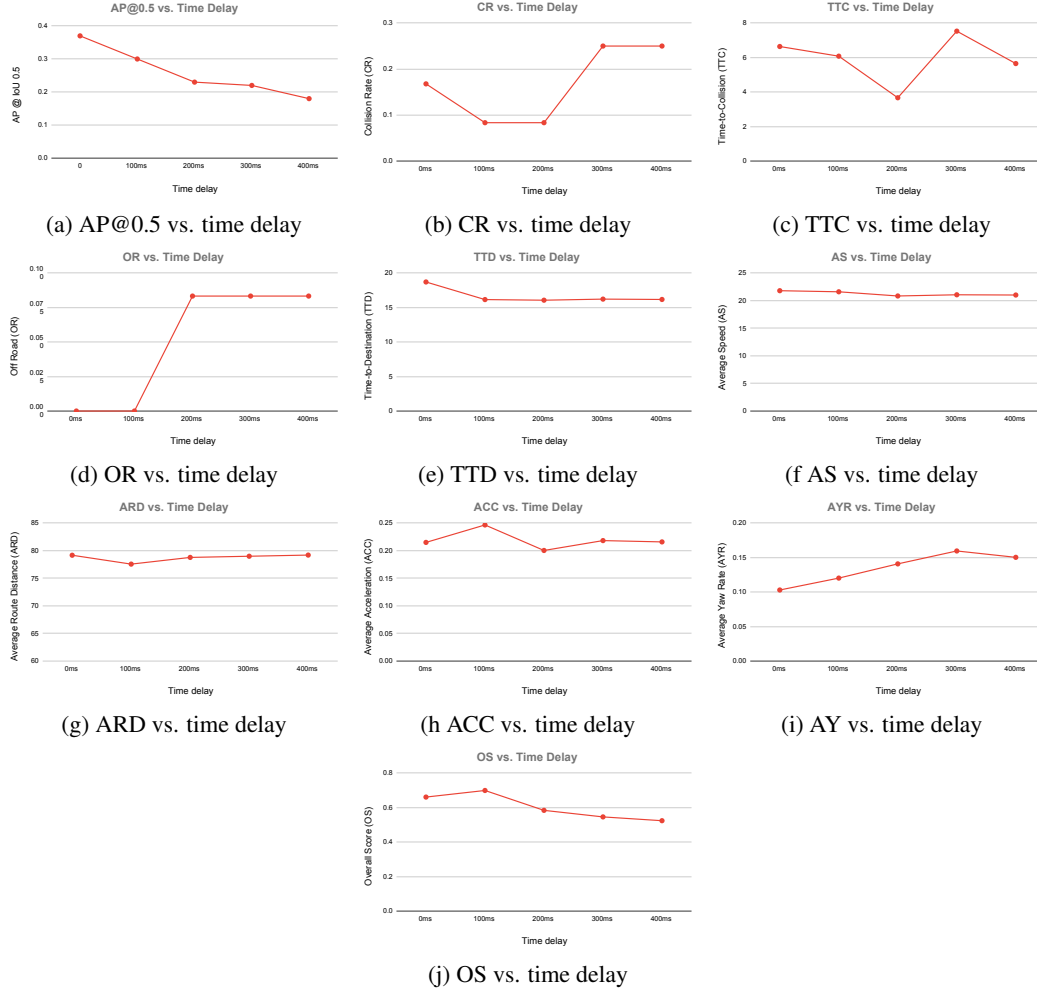


Figure 8: **Investigating the impact of time delay on the proposed evaluation metrics using the OPV2V model as a case study.** We observe that (a) AP@0.5 is decreasing gradually as the time delay increases from 0 to 400 ms, while the collision rate (CR) and other safety-related metrics like TTC do not necessarily deteriorate. Suprisingly, the overall score (OS) increases at 100 ms time delay but drops largely when latency surpasses 200 ms.

comprehensively model and test in a simulation environment. While simulation is an effective tool for synthesizing data, there is always a gap between simulation results and real-world performance. The benchmark’s results might not fully generalize to real-world deployments due to differences in sensor configurations, vehicle dynamics, and environmental interactions that are difficult to model accurately in a simulation. Addressing these limitations through continuous refinement and incorporating real-world data and scenarios will be crucial for the reliability and safety of autonomous driving systems. We call for rigorous on-road testing to ensure that models trained on simulation datasets would improve the safety of real-world complex scenarios.

A.7 Potential Negative Societal Impacts

We carefully designed twelve safety-critical scenarios to challenge the current cooperative perception systems. Releasing this source data would provide malicious hackers with additional resources to design adversarial or backdoor attacks in real-world scenarios. Moreover, our V2V cooperative systems would require to collect vast amounts of data for each vehicle, including detailed information about vehicle locations, passenger identities, and surrounding environments. This can raise significant privacy concerns regarding who has access to this data and how it is used. Failure to properly

handle this sensitive data can lead to misuse or unauthorized surveillance. The integration of V2V or V2X communication systems introduces new cybersecurity vulnerabilities. Malicious actors could potentially hack into these systems from a weak node in the network, causing entire graph malfunctions or data breaches. This networked risk can pose serious threats to public safety and security. Despite showing impressive performance and potential, future researchers should explore such techniques more responsibly.