Invited Paper: Cooperative Perception for Road Safety in Vehicular and VRU Microclouds

Stefano Ravera[1][0009-0000-5051-2221], Francesco Raviglione[1][0000-0003-1136-8049], Corrado Puligheddu[1][0000-0001-9961-0412], Ziqi Zhou[2][0009-0008-3559-8985], Ahmed Hasan Ansari[2][0009-0002-5485-1452], Diego Gasco[1][0009-0008-0837-7490], Marco Rapelli[1][0000-0001-9259-5387], Falko Dressler[2][0000-0002-1989-1750], Claudio Casetti[1][0000-0002-9507-8526], and Carla Fabiana Chiasserini[1][0000-0003-1410-660X]

[1] Politecnico di Torino, Turin, Italy
{name.surname}@polito.it
[2] TU Berlin, Telecommunication Networks group (TKN), Berlin, Germany
{zhou,ansari,dressler}@ccs-labs.org

Abstract. Vulnerable Road Users (VRUs) have benefited far less than car occupants from the recent road fatalities drop, motivating cooperative perception, in which connected vehicles and infrastructure share sensing information about VRUs. Existing solutions, however, rely on communication metrics that are only implicitly linked to safety, or place a collision avoidance application in the loop while abstracting perception. To fill this gap, we present a modular simulation framework combining CARLA/OpenCDA sensor-level perception, a trajectory-based Collision Avoidance Algorithm (CAA), an ETSI-compliant VRU Awareness Basic Service, and 5G network simulation based on ns-3 and 5G-LENA, enabling end-to-end evaluation of a road safety application that uses real sensor-level detections. We use our framework to compare three policies with increasing degree of cooperation within a vehicular and VRU edge cloud at an urban intersection: local collision avoidance based on onboard perception only, centralized collision avoidance at an edge server fusing vehicle detections, and an infrastructure-assisted policy that also integrates VRU Awareness Messages. Results show that cooperation reduces the number of missed VRUs within a 40 m reference range by a factor > 2, and improves the collision-risk detection probability by up to 48.5%, at the cost of a slightly higher end-to-end latency.

Keywords: Cooperative perception · Vehicular network · VRU Safety

1 Introduction

The "2023 Statistics of Road Traffic Accidents" report [37] identifies a general decline in road traffic accident fatalities across European countries over the decade 2011–2021. In many countries, however, this reduction has been substantially greater for car occupants than for Vulnerable Road Users (VRUs), largely because pedestrians, cyclists, and motorcyclists are intrinsically exposed and fragile in collisions, not being protected by the structural enclosure of a vehicle. Given their critical role in traffic safety, VRUs have been incorporated in the ETSI standards for Cooperative Intelligent Transportation Systems (C-ITS) [11, 10, 12]. However, since current commercial smartphones do not support direct vehicle-to-everything (V2X) communication in the 5.8–5.9 GHz ITS bands, VRUs must be detected by vehicle- or infrastructure-mounted sensors, although smartphone position updates over cellular networks may provide an additional source of information. This motivates cooperative perception, in which vehicles, VRUs, and infrastructure nodes share sensor data, intermediate features, or object-level detections [18]. A natural way to organize such cooperation is to group road users gathered around a common location, such as an intersection, into a microcloud: a dynamic cluster of *connected* nodes that share information and computing resources, possibly with the support of a nearby edge server [8, 7].

In this context, simulation plays an essential role: large-scale real-world experiments involving VRUs pose substantial safety, logistic, and cost constraints, whereas simulation enables controlled, repeatable, and scalable evaluations. The combination of the CARLA [6] autonomous driving simulator and OpenCDA [42], which runs on top of CARLA providing perception functionalities, is particularly well suited to sensor-level cooperative perception studies, and user applications can easily be developed on top of it. Despite the maturity of these simulation ecosystems, existing evaluations of cooperative VRU safety remain fragmented along two lines. On the one hand, studies of ETSI Collective Perception and VRU Awareness services typically adopt communication-centric metrics, such as awareness ratio or age of information, to quantify the benefit of exchanging Collective Perception Messages (CPMs) and VRU Awareness Messages (VAMs) [38, 19, 16]; such metrics are only implicitly connected to actual VRU safety, since no safety application consumes the improved awareness. On the other hand, works that place a collision avoidance application in the loop either abstract the perception process, feeding it with idealized positions extracted from awareness messages [24, 29], or evaluate a single, fixed cooperation architecture [15]. It thus remains an open challenge to determine the optimal service placement (centralized or decentralized) and VRU detection architecture for end-to-end cooperative perception and communication pipelines supporting VRU safety: indeed, these choices involve trade-offs between information completeness and collision detection reliability that have not yet been fully evaluated in a sensor-level simulation with a safety application in the loop.

To fill this gap, we present an end-to-end simulation framework for distributed cooperative driving and VRU protection, and use it to evaluate three policies with increasing levels of cooperation. First, we integrate a Collision Avoidance Algorithm (CAA), adapted from [3], into OpenCDA/CARLA, combining cooperative perception with a complete VRU safety service. Second, to the best of our knowledge, we provide the first application-in-the-loop comparison of three policies for VRU safety: *(i)* a distributed policy, where each vehicle runs a local CAA on its own VRU detections; *(ii)* a centralized policy, where vehicles send detections, via 5G, to an edge server running the CAA; and *(iii)* an infrastructure-assisted policy, where VRUs also broadcast their positions through VAMs generated by a smartphone application. Using a regulated-intersection scenario with vehicles and cyclists, we quantify the safety gains of each cooperation level and compare cooperative perception with non-cooperative approaches.

2 Related Work

VRU safety is a key objective of smart cities, particularly in the interaction between VRUs and Connected and Automated Vehicles (CAVs). Several research projects and living labs have addressed it through cooperative and data-driven approaches [36, 13, 2, 39] and demonstrated the feasibility of VRU-oriented services based on infrastructure sensing and edge computing [4, 17, 21, 34]. A large body of work has also studied learning-based cooperative perception for CAVs, exchanging raw data, intermediate features, or object-level detections [43, 41, 20, 22], and datasets such as V2X-Real [40] have brought it closer to realistic operating conditions. These works show that cooperative fusion improves detection under occlusions and limited sensing coverage, but they mainly evaluate perception-oriented metrics (e.g., average precision), treat VRUs as perceived objects rather than active participants, and do not connect the fused output to a safety application; hence, unlike our work, they do not quantify whether perception improvements ultimately translate into fewer collisions or earlier interventions. Our work is complementary: rather than proposing a new fusion network, we investigate how the placement and availability of perception information affect an operational collision avoidance service.

In parallel, collision avoidance services have been investigated as pivotal solutions for road safety: [3] presents a MEC-hosted, trajectory-based collision avoidance service for intersections fed by Cooperative Awareness Messages (CAMs), while [24] extends it to smartphone-equipped VRUs. Notably, [24] already established, for the same CAA and under positions self-declared by connected road users, that a centralized edge-based deployment outperforms a purely distributed one; in our work, instead, we assess whether, and to which extent, a centralized approach remains convenient when high-frequency sensor-based perception replaces self-declared positions used in [3, 24]. On the standardization side, ETSI has enabled active VRU participation through the VRU Awareness Basic Service and VAM messages [10]; nevertheless, purely VAM-based services are limited by the unpredictable mobility of VRUs, the insufficient accuracy of smartphone GNSS for safety-critical applications [44], VRUs without mobile devices, and the lack of direct V2X support in consumer devices, making cooperative detection by vehicles crucial for VRUs that do not reliably run a VRU Awareness Service.

Finally, co-simulation frameworks combining microscopic traffic and discrete-event network simulators dominate the state of the art: Veins [35] couples SUMO and OMNeT++, Artery [31] adds a standard-compliant ETSI ITS-G5 stack, recent extensions model active VRUs as connected C-ITS stations [19], and the ns-3-based VaN3Twin framework [28, 30] has been recently integrated with

CARLA to simulate connected, sensorized vehicles [32]. Yet, no single tool combines high-fidelity sensor-level perception, an easy-to-use cooperative perception pipeline, and a real-world standalone CAA. We fill this gap with a modular OpenCDA/CARLA framework, designed for future integration into co-simulation pipelines such as VaN3Twin. Additionally, with respect to [32], which already couples CARLA, OpenCDA-based fusion, and ns-3, we add a field-tested CAA safety application in the loop (on board or at the edge), a dedicated scenario configuration, and active VRU participation.

3 Reference Scenario and System Model

This section introduces the reference scenario and the system model used to compare the VRU safety policies under study: we first describe the road safety scenario and the possible information flows, and then formalize the system model.

3.1 Network and Application Scenario

We consider a safety-critical, four-way, regulated urban intersection where CAVs interact with VRUs, namely, pedestrians and cyclists. These entities have asymmetric capabilities: CAVs carry environmental sensors and on-board computing resources, and can therefore run models to detect other road users; VRUs carry no environmental sensors, and can either remain passive objects of vehicle perception or report their own kinematic state through standardized messages sent by a mobile device. Our scenario is represented in Figure 1, which shows the reference urban intersection in the CARLA graphical user interface and an example of a fused camera and LiDAR detection from a CAV, detecting another vehicle and a VRU. All participating road users are within the coverage of a 5G New Radio Standalone (NR SA) cell serving the intersection: a gNB provides Uu connectivity, and a co-located edge-computing node runs the safety service. The CAVs, connected VRUs, and edge node form a vehicular and VRU microcloud [8]: a dynamic group that makes locally generated information and computing resources available to run the CAA [45]. Co-location removes any relevant gNB-edge node delay, although the two remain logically distinct, performing communication and computing functions, respectively.

Protecting a VRU requires the safety service to maintain up-to-date knowledge of its kinematic state: a vehicle-VRU collision risk can only be assessed for VRUs whose position and motion are known to the service. Such information can reach the service through two complementary paths: a CAV can detect the VRU via its on-board sensors and either exploit the resulting object-level description locally or upload it to the edge node via 5G, or the VRU itself can upload its own position, speed, and heading via its smartphone by means of VAMs, as standardized by ETSI [12]. Since consumer smartphones do not yet integrate automotive V2X technologies (e.g., IEEE 802.11p or C-V2X), VAMs can be transmitted over the cellular Uu interface, with a generation frequency between 0.2 and 10 Hz, depending on how quickly the VRU changes its state [12].

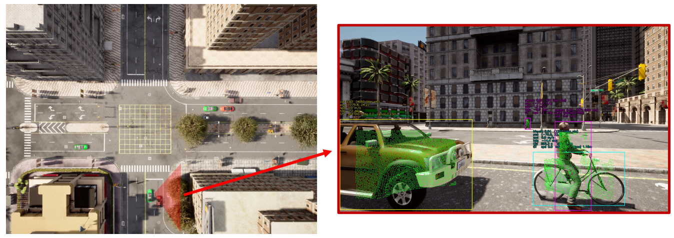


Fig. 1: Reference scenario and example of lateral fused detection by a CAV, with a detected vehicle and VRU.

The CAA, instead, can run either locally, on each CAV, or on the edge node. It consumes the most recent available states of road users, projects their trajectories, and evaluates whether or not a vehicle-VRU pair may be at risk. A local instance can react without communication but is limited to the objects perceived by its host vehicle. Conversely, an edge instance incurs network delay but can combine detections from multiple CAVs and, when available, VAMs.

3.2 System Model

The abstractions below are assumed to be common to all cooperative perception policies under study. Section 4 then specializes this common model into three VRU safety policies, each defined by a different CAA placement and a different set of inputs available to the CAA.

Road Users and Mobility Model. Let $\mathcal{V}$ be the set of CAVs and let $\mathcal{U} = \mathcal{U}^{\text{ped}} \cup \mathcal{U}^{\text{cyc}}$ contain the pedestrian and cyclist VRUs. The state of any road user $k \in \mathcal{V} \cup \mathcal{U}$ at time t is $\mathbf{z}_k(t) = \big(\mathbf{p}_k(t), \mathbf{v}_k(t), \mathbf{a}_k(t), h_k(t), c_k\big)$, where $\mathbf{p}_k$, $\mathbf{v}_k$, and $\mathbf{a}_k$ are the two-dimensional position, velocity, and acceleration, h_k is the heading, and $c_k \in \{\text{CAV}, \text{pedestrian}, \text{cyclist}\}$ is the road user type. Each CAV or cyclist follows a route compatible with the lane geometry and traffic regulation, while pedestrians move along walkable areas and crossings.

Communication Network Model. Let g denote the gNB and e the co-located edge node, so that the complete set of nodes is $\mathcal{N} = \mathcal{V} \cup \mathcal{U} \cup \{g, e\}$. Every road user $k \in \mathcal{V} \cup \mathcal{U}$ is attached to the gNB through the 5G NR SA Uu interface; communication between road users thus takes place through the cellular infrastructure. The underlying 5G model supports scheduling, propagation, retransmissions, and packet loss, mechanisms that need not be reproduced analytically here because the application consumes their end-to-end outcome. For a message of D bits generated by node k, let $T_k^{\text{ul}}(D, t)$ be the measured uplink delivery time to the edge, including radio access and the negligible gNB-edge transport component. $T_i^{\text{dl}}(D, t)$ is the delivery time of an edge-generated warning message to CAV i, and only successfully received packets contribute to the maps defined below. The relevant payload sizes are D_j^{obj} for the object list produced by CAV j, D^{vam} for a VAM, and D^{alert} for a CAA warning.

Computing Model. Each node $n \in \mathcal{N}$ has a computing capacity Λ_n, so that a task with workload F experiences a processing time F/Λ_n. We distinguish the perception workload $F^{\mathrm{det}}(m)$, the VAM-generation workload F^{vam}, and the collision-check workload F^{caa}. CAVs execute perception and maintain a Local Dynamic Map (LDM) and, depending on the policy, run the CAA. A connected VRU only obtains its own state and generates VAMs. The edge node maintains a centralized LDM (i.e., a Server LDM (S-LDM [33])) and executes the centralized CAA; the gNB only forwards traffic.

Perception Model. The sensors of all CAVs are synchronized to a common clock and sample at the constant frequency $f_s = 1/\tau$, where τ is the perception period. At every sample, each CAV i collects its sensor observations and executes the complete perception pipeline. The selected detection model complexity m captures the accuracy-inference-time trade-off among the available model variants, with computational effort $F^{\mathrm{det}}(m)$. The output produced by CAV i at perception cycle r is the set

$$\mathcal{O}_i[r] = \big\{o = (\mathbf{p}_o, \mathbf{v}_o, h_o, c_o, \gamma_o, t_o)\big\}, \tag{1}$$

where $\mathbf{p}_o$, $\mathbf{v}_o$, h_o, and c_o are the position, velocity, heading, and class assigned to object o, $\gamma_o \in [0, 1]$ is the confidence score returned by the detector, and t_o is the sample timestamp. Each CAV stores its latest object tracks in an LDM $\mathcal{L}_i(t)$. The LDM acts as the interface between perception and the safety service, and a CAA instance consumes the same object representation irrespective of where it runs (locally or at the edge).

Information Sharing and Object Collection at the Edge. When cooperation is enabled, two types of information are delivered to the edge. (i) *Vehicle detections*: a CAV i transmits the object-level content of $\mathcal{L}_i(t)$, including kinematics, road user class, timestamp, and confidence. (ii) *VRU awareness*: a connected VRU u transmits its self-declared state $\mathbf{y}_u(t) = (\mathbf{p}_u, \mathbf{v}_u, h_u, c_u, t_u)$ in a VAM generated according to the standard frequency rules [12].

The edge maintains a centralized LDM $\mathcal{L}_e(t)$, following the concept of the S-LDM in [33], which collects only successfully received contributions. Let $\mathcal{S} \subseteq \mathcal{V} \cup \mathcal{U}$ be the set of nodes allowed to provide information to the edge under a given policy, i.e., the CAVs whose object lists are uploaded and the connected VRUs whose VAMs are integrated. Hence, each policy in Section 4 corresponds to a different choice of $\mathcal{S}$. Given $\mathcal{S}$, an S-LDM content can be abstracted as

$$\mathcal{L}_e(t) = \varPhi(\{\mathcal{L}_i(t - \delta_i)\}_{i \in \mathcal{S} \cap \mathcal{V}}, \{\mathbf{y}_u(t - \delta_u)\}_{u \in \mathcal{S} \cap \mathcal{U}})\,, \tag{2}$$

where $\varPhi(\cdot)$ matches entries referring to the same physical object and retains their most recent information, while δ_k denotes the age of the most recent information from source k available at the edge at time t, accounting for both the source's generation timing and the uplink delivery delay. The confidence attached to vehicle detections is stored but it does not currently affect $\varPhi$. A VRU is present in $\mathcal{L}_e(t)$ when it appears in at least one successfully delivered CAV object list, or in its own successfully delivered VAMs. Completion of a local perception cycle updates $\mathcal{L}_i$, whereas reception of an object list or VAM updates $\mathcal{L}_e$; each update immediately triggers the relevant CAA instance, with no batching delay.

Collision avoidance service. The safety service is a trajectory-based CAA adapted from [3, 25], which can be executed locally or centrally at an edge server, depending on the considered policy. Given an input LDM $\mathcal{L}$, either a local $\mathcal{L}_i$ or the edge map $\mathcal{L}_e$, it maintains a set $\mathcal{R}$ containing the latest information of the tracked road users. Each tracked entry of $\mathcal{L}$ is mapped to an array $\mathbf{r}_k = (\mathbf{p}_k, \mathbf{v}_k, \mathbf{a}_k, \alpha_k, t_k^{last})$, expressed in Cartesian (projected) coordinates, where α_k is the heading of road user k with respect to the North and t_k^{last} is the timestamp of the last update of the entry. For entries originating from on-board perception, these quantities are derived from the estimates $(\mathbf{p}_o, \mathbf{v}_o, h_o)$, with the acceleration estimated from consecutive track updates. For entries originating from VAMs, instead, they are the self-declared states from the road users' navigation systems.

Consistently with the event-driven operation described above, the CAA is executed every time an array $\mathbf{r}_k$ is created or updated. For every other tracked road user $\mathbf{r}_w \in \mathcal{R} \setminus \{\mathbf{r}_k\}$, entries whose information is outdated (i.e., $t_{now} - t_w^{last}$ exceeding a staleness threshold) are ignored. For users who are nearer than a given distance threshold, a locally uniformly accelerated motion is assumed, and the squared distance between road users k and w is computed as:

$$D_{k,w}(t) = \left\| \mathbf{p}_k + \mathbf{v}_k\, t + \tfrac{1}{2}\mathbf{a}_k\, t^2 - \left(\mathbf{p}_w + \mathbf{v}_w\, t + \tfrac{1}{2}\mathbf{a}_w\, t^2\right) \right\|^2. \tag{3}$$

Starting from the definition of $D_{k,w}(t)$, we define the Time-to-Collision $t2c$ as the earliest non-negative stationary point of the projected squared separation, i.e., $t2c = \min\{t \geq 0 : \frac{\mathrm{d}D_{k,w}(t)}{\mathrm{d}t} = 0\}$ (with a k,w pair being discarded if no such instant exists, or if $D_{k,w}$ is constant, so that the condition holds for every t), and the Space-to-Collision $s2c = \sqrt{D_{k,w}(t2c)}$ as the projected distance at that instant. In the general accelerated case $D_{k,w}$ is quartic, hence its earliest non-negative stationary point may not necessarily be its minimum. However, in the evaluation in Section 6, the CAA is fed with $\mathbf{a}_k = \mathbf{0}$, so that $D_{k,w}$ is quadratic, admits at most one stationary point, and $t2c$ is exactly the instant of minimum distance at which two road users are expected to approach each other. A collision risk between v and w is then declared as such if both $t2c$ and $s2c$ are below two tunable thresholds, $t2c_{th}$ and $s2c_{th}$. Both can be tuned for balancing false positives and false negatives, and, consistently with the sensitivity analysis in [23] and in line with the fine tuning performed in the 5G-TRANSFORMER Horizon 2020 project, can be set to 4.8 s and 4.2 m, respectively, for a typical urban scenario. Upon risk detection, the CAA issues an alert to the endangered vehicle, which can start an emergency maneuver after an actuation delay T_i^{act}.

End-to-End Latency Model. The safety benefit of the CAA depends on the delay between the instant at which a dangerous situation becomes physically observable and the instant at which an involved vehicle starts its avoidance maneuver. Consider a VRU u and an involved CAV i, i.e., a CAV that must receive an alert, and let $h \in \{i, e\}$ denote the CAA host. Information about u may originate from any CAV $j \in \mathcal{V}$ that detects it: $j = i$ when the involved vehicle perceives the VRU itself, whereas $j \neq i$ is possible only when detections are pooled at the edge ($h = e$).

The latency of the path through detecting CAV j is

$$T^{\mathrm{e2e}}_{j,i} = W^{\mathrm{sens}}_j + \frac{F^{\mathrm{det}}(m)}{\Lambda_j} + \mathbb{1}\{h=e\}\left[T^{\mathrm{ul}}_j(D^{\mathrm{obj}}_j) + T^{\mathrm{dl}}_i(D^{\mathrm{alert}})\right] + \frac{F^{\mathrm{caa}}}{\Lambda_h} + T^{\mathrm{act}}_i, \quad (4)$$

where $0 \leq W^{\mathrm{sens}}_j \leq \tau$ is the wait until the next synchronized sample, $\mathbb{1}\{\cdot\}$ is the indicator function, and T^{ul}_j and T^{dl}_i are the Uu delivery times introduced in the network model. When the CAA is local ($h = i$), only the host own detections are available, thus $j = i$ and the indicator removes both transfer terms. When $h = e$, the object list and the alert traverse the uplink and downlink, respectively. Thus, the same expression covers both local and edge CAA service placement.

If a VRU is revealed by its own VAM, perception is replaced by VAM generation and the corresponding latency becomes

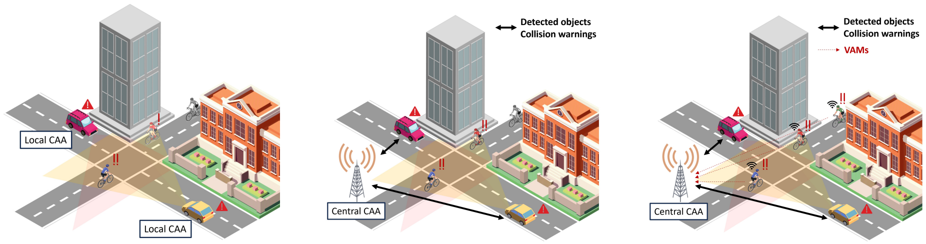
$$T^{\mathrm{e2e}}_{u,i} = W^{\mathrm{vam}}_u + \frac{F^{\mathrm{vam}}}{\Lambda_u} + T^{\mathrm{ul}}_u(D^{\mathrm{vam}}) + \frac{F^{\mathrm{caa}}}{\Lambda_e} + T^{\mathrm{dl}}_i(D^{\mathrm{alert}}) + T^{\mathrm{act}}_i, \quad (5)$$

where W^{vam}_u is bounded by the current VAM interval selected by the standard generation rules. Since every completed update immediately triggers a CAA check, no batching or LDM data gathering time appears in either expression, and the effective reaction latency of the service to the risk is that of the fastest available path, i.e., $T^{\mathrm{react}}_{i,u} = \min_{j\in\mathcal{S}_u} T^{\mathrm{e2e}}_{j,i}$, where $\mathcal{S}_u \subseteq \mathcal{S}\cup\{i\}$ is the set of sources whose updates can reveal u to the CAA instance in use: only vehicle i under the distributed policy, the detecting vehicles under the centralized policy, and additionally u itself under the infrastructure-assisted policy. Thus, increasing the cooperation level enlarges $\mathcal{S}_u$, trading additional communication delay for a higher probability that at least one fast delivery path exists.

4 Distributed Computing for Safe Autonomous Driving

Our work focuses on the implementation and performance assessment of three main policies for VRU collision avoidance, aimed at improving safety for Connected Cooperative and Automated Mobility (CCAM). The three policies act on the same scenario (road users, perception pipeline, and CAA) defined in Section 3; what changes is how far locally generated information is allowed to travel before a safety decision is made. Specifically, *NoCoop* keeps perception and collision avoidance within each CAV; *VehCoop* merges together vehicle detections at the edge; and *FullCoop* further allows VRUs to report their own state using standardized messages delivered through the cellular network. The policies therefore correspond to increasing levels of cooperation without changing the underlying safety logic, as depicted in Figure 2.

The first policy, *NoCoop*, Figure 2a, yields a fully distributed, non-cooperative scenario, in which each CAV operates as a self-contained safety system: it processes its sensor samples, updates its on-board LDM, and immediately runs the CAA on the locally available objects. No detection is uploaded and no remote warning is required, so the uplink and downlink terms of Eq. (4) are zero, while



(a) *NoCoop* (b) *VehCoop* (c) *FullCoop*

Fig. 2: The VRU protection policies with increasing levels of cooperation: (a) locally, the blue cyclist is detected by both vehicles, while the red cyclist only by the yellow vehicle; (b) combining vehicle detections at the edge, both cyclists are known; (c) with VAMs, the fully hidden cyclist can also report its own state.

perception and collision checking both use the on-board computing capacity. This is the shortest information path and does not depend on network coverage; however, the CAA can only reason about what its host vehicle has detected, and a VRU hidden from that CAV, like the red cyclist hidden from the red vehicle in Figure 2a, will be absent from its LDM and hence unable to trigger an alert.

The second policy, *VehCoop*, Figure 2b, leads instead to a centralized scenario in which CAVs upload their detections to the edge, which aligns the received contributions in a centralized LDM and runs a single CAA instance. A VRU hidden from an endangered CAV (e.g., the red cyclist hidden from the red car in Figure 2b) may still be considered by the CAA if another vehicle has detected it (e.g., the yellow vehicle). When the CAA identifies a risk, the edge returns a warning to the affected vehicle. This policy turns CAVs into cooperating viewpoints and moves collision checking to the more capable edge node, at the cost of the uplink and downlink communication latency. Moreover, VRUs remain solely perceived objects, and may be invisible when unobservable to all nearby vehicles.

Finally, the third policy, *FullCoop*, Figure 2c, retains vehicle cooperation and adds direct VRU participation: pedestrians or cyclists periodically send VAMs containing their own states to the edge server, where the decoded information is inserted into the same centralized LDM as the vehicle detections. VAMs complement rather than replace perception: the CAA instance at the edge leverages whichever fresh contributions are available. Therefore, a VRU hidden from every vehicle sensor can still become known to the CAA through its own VAMs. This policy corresponds to the highest cooperation level and the most demanding deployment, as it assumes active VRU participation, even though the latter do not employ any camera or LiDAR. This helps improve the perception completeness, and the performance of the collision avoidance system; however, it also introduces additional delays due to communication and active VRU participation. Note that VAMs experience a generation wait bounded by 5 s, due to standard triggering rules [12], while the perception delay is bounded by 50 ms. Hence, they help improve completeness rather than reducing the reaction time.

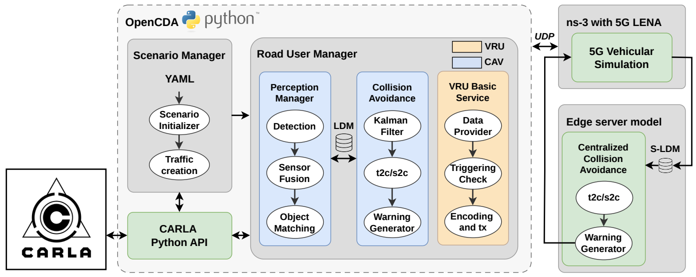


Fig. 3: High-level architecture of our framework, combining CARLA, 5G network simulation, the CAA and a VRU Basic Service for the transmission of VAMs.

5 Simulation Framework

The high-level architecture of our integrated simulation framework is depicted in Figure 3, with the main novelty consisting of the component integration, the Python VRU Basic Service, the porting and adaptation of the CAA, the scenario and ground-truth metric collection, and the 5G simulation harness. Our modular framework couples *(i)* CARLA [6] and OpenCDA [42] for CAV/VRU mobility, sensors, and perception; *(ii)* the trajectory-based CAA, deployed locally or at the edge; and *(iii)* an ns-3.46/5G-LENA 4.1.1 harness that simulates 5G communication using CARLA/OpenCDA positions and parameters. Each block can be replaced independently, enabling future integration into full co-simulation pipelines based on ns-3, such as VaN3Twin [32]. As a concrete example, the Python VRU Basic Service can be removed and the CAA interfaced directly, via UDP, with the VRU Basic Service of VaN3Twin [28].

Within OpenCDA, the *Scenario Manager* instantiates the simulation from a YAML description, managing the map, all CAVs, background traffic, and VRUs. Each CAV or VRU is handled by a *Road User Manager*, which comprises three submodules. First, only for CAVs, the pre-existing *Perception Manager* inherited from OpenCDA [42], and already exploited for perception in [32], implements the perception model, in which RGB camera frames are processed by the YOLOv8 model, detections are fused with LiDAR data to obtain 3D object positions, and an object-matching step associates detections across frames and sensors into consistent sensed objects. Second, only for CAVs, we introduced a novel local *Collision Avoidance* module, based on the code released by [3]. This module includes a newly implemented Kalman Filter for speed and heading estimation, besides all the logic for $t2c$ and $s2c$ computation and local warning generation. Third, only for VRUs, a VRU Basic Service module implements the transmission side of the ETSI VRU Awareness Basic Service [12]. This module, loosely based on the service available in VaN3Twin [28, 30, 16], has been ported to Python, along with novel logging and metrics collection capabilities. Depending on the VRU mobility, VAMs are sent between 0.1 and 5 s as per the standards.

The edge server model, instead, implements the collision avoidance service, which is used for the *VehCoop* and *FullCoop* policies. Object-level detections are uploaded by vehicles and, when enabled, VAMs received from VRUs are collected into a centralized LDM which follows the concept of the S-LDM [33] but has been re-implemented to store information about road users without the need of connecting to an AMQP 1.0 broker. Each new update, either from detections or from VAMs, immediately triggers the centralized CAA instance, whose warning generator delivers downlink alerts to vehicles involved in possible collisions.

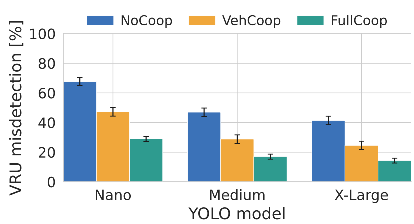
All message exchanges traverse the simulated 5G network detailed in Section 6, implemented by a dedicated ns-3 bridge application (labeled as *5G Vehicular Simulation* in Figure 3), based on the 5G-LENA module [27] and taking its time reference from OpenCDA, so that the network simulation evolves alongside CARLA. The *CARLA Python API* connects OpenCDA with CARLA, while a dedicated interface links ns-3 with OpenCDA, via a lightweight JSON-over-UDP protocol. Notably, CARLA/OpenCDA provides ns-3 with road user positions and the types of messages to be transmitted over the network, while ns-3 returns which warning messages were successfully delivered to vehicles potentially involved in VRU collisions. Each packet is generated inside the simulation and delivered over the 5G NR Uu interface, while the measured one-way latency is logged together with the simulation time, message type and size.

6 Performance Evaluation

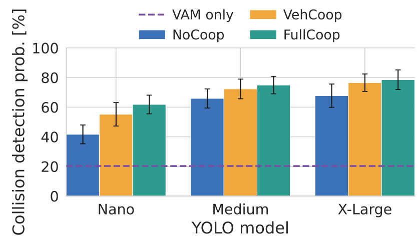
Simulation Settings. We configured the simulation framework to simulate a 5G NR Standalone (SA) network, in a typical urban microcell (3.5 GHz (n78), with 60 MHz bandwidth and numerology $\mu = 1$). A 40 dBm gNB on a 10 m mast serves vehicle and VRU UEs (23 dBm, power class 3) at 1.5 m average height, using the 3GPP TR 38.901 Urban Micro (UMi) channel model. The system uses a Time Division Multiple Access (TDMA) round-robin scheduler, a non-Guaranteed Bit Rate (non-GBR) low-latency Enhanced Mobile Broadband (eMBB) bearer carrying UDP/IPv4, and a symmetric slot-based Time Division Duplex (TDD) pattern with 4 downlink, 4 uplink, and 2 special slots. Vehicle-to-Network messages with detections are sent to an edge server co-located with the gNB and connected to the 5G Core by 100 Gb/s fiber. Radio Link Control uses 5G-LENA's default unacknowledged mode, so packets failing all Hybrid Automatic Repeat reQuest retransmissions are lost and the results capture both latency and radio-induced loss. Transmitted payload sizes depend on the scenario and the number of reported objects; their measured averages are 886, 83, 432 bytes for D_j^{obj}, D^{vam}, and D^{alert}. In the CAA, data older than 0.5 s is discarded, and detections within 2 m are clustered and treated as the same object. As the reference scenario in CARLA, we select the most relevant urban intersection of the `Town10` map, representing a realistic inner-city environment (Figure 1). Mobility is simulated using the autopilot model, which drives road users following the realistic patterns expected from real drivers, pedestrians, and cyclists. We consider 4 CAVs equipped with 5 synchronized sensors, i.e., 4 RGB cameras (1 front, 2 lateral, 1 rear) and 1 LiDAR, a combination shown to support accurate object detection and scene understanding [5]. All sensors sample at $f_s = 20$ Hz (i.e., $\tau = 50$ ms). Also, we select YOLOv8n, YOLOv8m, and YOLOv8x, which offer different detection accuracy-inference time trade-offs. For object detection, we use 1920×1080 frames and the stock YOLOv8n/m/x weights, without dedicated fine-tuning, as our aim is to assess cooperation under a given detection model, not to optimize the model itself. Sensor-equipped CAVs approach the reference intersection from four different directions, while pedestrians and cyclists are generated at random spawn points around it, following realistic mobility patterns. Cyclists share the roadway with vehicles, whereas pedestrians are restricted to sidewalks and may cross only at designated crosswalks.

Perception is executed inside the simulation loop, i.e., the CAA consumes the detections produced by the sensor pipeline, and each object list carries the timestamp of the sample that generated it. To speed-up simulation time, CARLA is stepped synchronously and all CAVs are processed sequentially within a step, so the simulated clock does not advance while a detection model runs. No simulation frame is therefore dropped, and each local LDM is refreshed once per step, i.e., at $f_s = 20$ Hz in simulated time, for all three models. The edge LDM instead is updated on delivery, so its update period additionally depends on VAM triggering, transmission delay, and losses. As a consequence, the measured inference time does not delay LDM freshness at runtime and it is currently accounted for a posteriori, in the latency breakdown of Figure 4c. This corresponds to assuming an on-board platform whose throughput is sufficient to keep up with the sampling rate. Such an assumption is compatible with the per-frame latencies of Figure 4c, provided that the per-camera inference streams are pipelined: a per-frame latency larger than the sampling period (50 ms) then delays each detection without reducing the rate at which detections are produced. The only residual effect is that each frame is processed with an age equal to the perception time, which is identical for all policies at a given m.

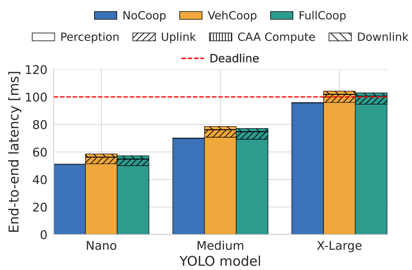
All 9 configurations (3 policies for 3 model sizes) were simulated under 10 different VRU patterns, obtained by varying the simulation seed, with each run lasting 80 s, long enough for CAVs to clear the intersection. Results were averaged over the 10 runs and are reported below with 90% confidence intervals. All gains reported in the following are relative changes, while, for the main results, we also report the absolute difference in percentage points. We also recorded ground truth values (i.e., position and kinematic state of each road user every 50 ms) directly from CARLA and fed them to the CAA implementation, obtaining the ground truth collision warnings expected under ideal conditions in which global information about all road users is available to the CAA. This ground truth includes 10 risk events per run on average (within a range 6–13). We remark that no actual collisions occur, as the CARLA autopilot actively avoids collisions following realistic mobility models. Hence, we define the ground truth risk events, against which collision detection probability is measured, as the events that are potential collision events detected by the Collision Avoidance Algorithm (CAA) when it is fed with complete and error-free knowledge of all road users in the intersection. Finally, the framework and full scenario configuration have been made available under an open source license [9].



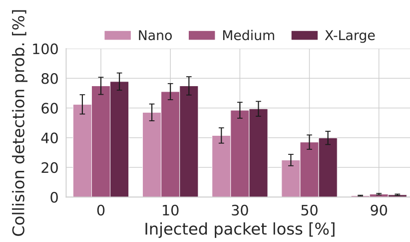
(a) Misdetected VRUs within 40 m



(b) Collision detection probability



(c) End-to-end latency and latency contributions



(d) Collision detection probability under injected packet loss for FullCoop

Fig. 4: Evaluation results of the three policies, for each selected YOLOv8 size.

Policy Evaluation. Our evaluation assesses the performance of the three policies described in Section 4, focusing on three main metrics: *(i)* the percentage of misdetected VRUs within a 40 m reference range, i.e., the fraction of in-range VRUs present in the ground truth but absent from the knowledge of a given policy, averaged over all CAVs and all 50 ms time steps; *(ii)* the collision detection probability, measured against the ground truth; and *(iii)* the resulting end-to-end latency, assessed against the 300 ms end-to-end requirement set by ETSI for VRU protection [10]; a 100 ms figure associated with the 5GAA Intersection Movement Assist and VRU Collision Risk Warning use cases [1] is also reported, as the perception and network latency should stay below the shortest VAM generation period foreseen by ETSI.

Figure 4a reports the percentage of misdetected VRUs within a 40 m radius, a distance at which a VRU is already relevant for the collision avoidance service [14, 26]: at speeds typically held when approaching a regulated urban intersection (around 30 km/h in a scenario with a 50 km/h speed limit), 40 m are covered in about 4.8 s, i.e., our selected $t2c$ threshold. As can be seen, cooperation is consistently more effective than additional model capacity: with the X-Large model, *VehCoop* and *FullCoop* reduce misdetections, in relative terms, by up to 40.7% and 65.4% with respect to *NoCoop*, down to just 14.3% of misdetections in the best case, against the 67.7% of the Nano model without cooperation.

Figure 4b reports the collision detection probability, i.e., the recall of the safety service, as the fraction of collision risks identified on ground-truth data that are also detected by each policy. Two main insights emerge. First, model size helps mainly up to Medium (improving *NoCoop*, *VehCoop*, and *FullCoop*, in relative terms, by 58.2%, 31.1%, and 21.2%, respectively), while the difference between X-Large and Medium is well within the confidence intervals. Thus, a medium-sized model may already be sufficient for reliable collision avoidance. Second, cooperation improves recall in all tested configurations, most evidently with the smallest model (Nano), for which *VehCoop* and *FullCoop* outperform *NoCoop* by 32.5% and 48.5% in relative terms (+13.6 and +20.2 percentage points), raising the detection probability from 41.6% to 55.2% and 61.8%, respectively; merging detections at the edge accounts for the largest share of the gain, with VAMs providing a further contribution. Furthermore, as a reference we also report a *VAM only* baseline, represented by the violet dashed line, which relies exclusively on cooperative broadcasts from all VRUs without onboard perception. Although assuming all VRUs as connected is an ideal situation (not all VRUs are likely to carry VAM-enabled mobile devices), it remains suboptimal: standard-compliant VAM transmission is inherently less frequent than local sensor updates, more often leading to outdated state tracks under the implemented kinematic triggering conditions [12]. For completeness, we also measured the precision of the CAA, which proved comparable across model sizes and lower for the cooperative policies (on average 65% for *NoCoop*, down to 48% for *FullCoop*), consistently with [24]. This is due to precision being evaluated per 50 ms frame, and to each vehicle being exposed to many more object estimates under cooperation (on average 15 to 22 against 3.9 VRU entries checked per vehicle and per frame). Some of these LDM entries are redundant views of the same road user caused by noisy detections and, as the number of real hazards is fixed, false positives grow when the LDM contains more entries. An improved kinematic state representation, together with improved object fusion at the edge, could enhance precision. Nevertheless, the most critical metric for safety is the false negative rate, i.e., the share of missed collision risks, which cooperation very substantially reduces, as depicted in Figure 4b.

Figure 4c breaks down the end-to-end latency into its perception, uplink and downlink communication, and CAA computation contributions. To measure the perception time, we considered CAVs equipped with Jetson Orin units, a good tradeoff between power consumption and computing capability. The sensing and actuation components in Eq. (4) are, on average, constant and policy-independent, and are thus not considered in our analysis. However, they can be easily summed to the measured values (in our case, it would add an average 25 ms W_j^{sens} for the 20-Hz sensor sampling, plus a CAV braking actuation time). For consistency, the VAM waiting time W_u^{vam} of Eq. (5) is excluded as well. We stress that this only affects *FullCoop* and is bounded by 5 s. However, in our scenario, VAMs are on average sent more frequently, due to the kinematics of VRUs [12]: the VAM inter-generation interval measured for the cyclists is 750 ms on average, up to 1500 ms at the 95th percentile, i.e., of the same order as or larger than the maximum 50 ms perception delay. Hence, since VAMs are used to improve completeness and the reaction time remains determined by W_j^{sens}, the results remain valid even though W_u^{vam} is neglected. Perception clearly dominates and is the only component clearly affected by the model size, growing from 51 ms (Nano) to 70 ms (Medium), and 96 ms (X-Large). The CAA computation time is negligible (0.3 ms in the worst case, with X-Large and *FullCoop*), while communication adds on average 5.3 ms in the uplink and 2.2 ms in the downlink. Cooperation increases latency by at most 8.9 ms (*VehCoop* with X-Large), which is an acceptable cost for the awareness and collision detection gains discussed earlier. As mentioned, the plotted values exclude W_j^{sens} and T_i^{act} and, hence, are a lower bound on Eq. (4). The 100 ms deadline is nevertheless relevant as the network and perception latency should stay below the minimum generation period set by ETSI and 5GAA [1, 12]. With X-Large, cooperation makes latency exceed the 100 ms target. Conversely, against the 300 ms ETSI requirement, adding $W_j^{\text{sens}} = 25$ ms leaves a residual budget for T_i^{act} of 224/205/179 ms (*NoCoop*) and 218/198/172 ms (*FullCoop*) for Nano/Medium/X-Large. For a representative $T_i^{\text{act}} = 200$ ms, Nano complies under all policies, X-Large under none, and Medium only under *NoCoop*.

At last, we underline that packet loss on the 5G network is always zero, as the 60 MHz cell is largely unused. However, to analyze the policy degradation under network congestion, we inject a uniform packet loss of 10, 30, 50, and 90% on detections, VAMs, and warnings, taking *FullCoop* as a reference. Figure 4d reports the resulting collision detection probability. The service degrades gracefully up to 10% loss, where recall drops by only 2.9%, 3.9% and 5.4% for X-Large, Medium and Nano respectively. It then decreases almost linearly up to 50%, where it is roughly halved for every model, and finally collapses at 90%, with fewer than 2% of the risks detected. Also, since *FullCoop* evaluates the CAA at the edge, it retains no onboard fallback, suggesting that a hybrid policy keeping a local CAA instance alive would be advisable whenever the network cannot be dimensioned to support the required load.

7 Conclusion and Future Work

We presented a novel CARLA/OpenCDA framework for evaluating cooperative perception strategies for VRU safety, supporting 5G-connected CAV and VRU microclouds, edge computing, sensor fusion, and YOLO-based detection. Using the simulator, our analysis of three policies with increasing cooperation at an urban intersection has shown that cooperation can reduce missed VRUs by up to 65.4% and the gap with an ideal-information reference by up to 20.2 percentage points, with a negligible latency increase. Future work will integrate our framework with VaN3Twin [28], develop an advanced autopilot supporting actual frontal collisions, and support advanced computing paradigms and perception models.

Acknowledgments. This work was supported by Fondazione CRT and Politecnico di Torino via "Bando Internazionalizzazione della Ricerca 2025" (CUP: E13C25002890005), CARS@POLITO, and Regione Piemonte through the Infra+ PATH project.

Disclosure of Interests. The authors declare that they have no competing interests.

References

1. 5G Automotive Association: C-V2X Use Cases and Service Level Requirements: Volume I. Tech. Rep. Version 2.0, 5G Automotive Association (Jan 2025)
2. AI4CCAM Consortium: AI4CCAM: Trustworthy AI for Connected and Cooperative Automated Mobility. https://www.ai4ccam.eu/ (2022–2025), project website
3. Avino, G., Bande, P., Frangoudis, P.A., Vitale, C., Casetti, C., Chiasserini, C.F., Gebru, K., Ksentini, A., Zennaro, G.: A mec-based extended virtual sensing for automotive services. IEEE Transactions on Network and Service Management **16**(4), 1450–1463 (2019). https://doi.org/10.1109/TNSM.2019.2931878
4. C-Roads Platform: M4.3 Annual deployment overview report 2023. Tech. rep., C-Roads Platform (2023)
5. Cui, Y., Chen, R., Chu, W., Chen, L., Tian, D., Li, Y., Cao, D.: Deep learning for image and point cloud fusion in autonomous driving: A review. Trans. Intell. Transport. Sys. **23**(2), 722–739 (Feb 2022). https://doi.org/10.1109/TITS.2020.3023541
6. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. pp. 1–16 (2017)
7. Dressler, F., Chiasserini, C.F., Fitzek, F.H.P., Karl, H., Lo Cigno, R., Capone, A., Casetti, C.E., Malandrino, F., Mancuso, V., Klingler, F., Rizzo, G.A.: V-Edge: Virtual Edge Computing as an Enabler for Novel Microservices and Cooperative Computing. IEEE Network **36**(3), 24–31 (5 2022). https://doi.org/10.1109/MNET.001.2100491
8. Dressler, F., Pannu, G.S., Hagenauer, F., Gerla, M., Higuchi, T., Altintas, O.: Virtual Edge Computing Using Vehicular Micro Clouds. In: IEEE International Conference on Computing, Networking and Communications (ICNC 2019). IEEE, Honolulu, HI (2 2019). https://doi.org/10.1109/ICCNC.2019.8685481
9. DriveX-devs: CARLA-CA-framework: Cooperative perception and collision avoidance simulation framework for vru safety. https://github.com/DriveX-devs/CARLA-CA-framework (2026)
10. ETSI: ETSI TS 103 300-2 V2.2.1 - Intelligent Transport Systems (ITS); Vulnerable Road Users (VRU) awareness; Part 2: Functional Architecture and Requirements definition; Release 2. Technical specification, European Telecommunication Standard Institute (2021)
11. ETSI: ETSI TR 103 300-1 V2.3.1 - Intelligent Transport Systems (ITS); Vulnerable Road Users (VRU) awareness; Part 1: Use Cases definition; Release 2. Technical report, European Telecommunication Standard Institute (2022)
12. ETSI: ETSI TS 103 300-3 V2.2.1 - Intelligent Transport Systems (ITS); Vulnerable Road Users (VRU) awareness; Part 3: Specification of VRU awareness basic service; Release 2. Technical specification, European Telecommunication Standard Institute (2023)
13. European Commission, CORDIS: Holistic and adaptivE Interface Design for human-technology Interactions (HEIDI). https://cordis.europa.eu/project/id/101069538 (nd)
14. Fernandez, A.E., Fallgren, M., Brahmi, N., Servel, A., Tiphene, J., Sun, W., Ström, E., Svensson, T., Bernardez, D., Vilalta, R., Alonso-Zarate, J., Kousaridas, A., Boban, M., Dillinger, M., Condoluci, M., Mahmoodi, T., Li, Z., Otterbach, J., Lefebvre, M., Vivier, G., Abbas, T., Wingård, P.: 5GCAR scenarios, use cases, requirements and KPIs. Deliverable D2.1, v2.0, 5GCAR Project (Feb 2019)

15. Gamerdinger, J., Teufel, S., Roller, S., Bringmann, O.: Carlancap: A framework for quantifying the safety of vulnerable road users in infrastructure-assisted collective perception using euroncap scenarios. In: 2025 IEEE 101st Vehicular Technology Conference (VTC2025-Spring). pp. 1–6 (2025). https://doi.org/10.1109/VTC2025-Spring65109.2025.11174321
16. Genovese, A., Rapelli, M., Raviglione, F., Casetti, C.: Etsi-compliant protection of connected vulnerable road users: Simulation and field trial. In: 2024 IEEE Vehicular Networking Conference (VNC). pp. 343–350 (2024). https://doi.org/10.1109/VNC61989.2024.10576014
17. Gkemou, M., Biral, F., Gkragkopoulos, I., Valenti, G., Tsetsinas, I., Bekiaris, E., Steccanella, A.: Cooperation between Roads and Vehicles: Field Validation of a Novel Infrastructure-Based Solution for All Road Users' Safety. Safety **7**(2) (2021). https://doi.org/10.3390/safety7020039
18. Han, Y., Zhang, H., Li, H., Jin, Y., Lang, C., Li, Y.: Collaborative perception in autonomous driving: Methods, datasets, and challenges. IEEE Intelligent Transportation Systems Magazine **15**(6), 131–151 (2023). https://doi.org/10.1109/MITS.2023.3298534
19. Hejazi, H., Bokor, L.: Modeling and evaluation of cooperative vulnerable road user protection schemes in realistic c-its environments. Computer Networks **246**, 110396 (2024). https://doi.org/10.1016/j.comnet.2024.110396
20. Hu, Y., Fang, S., Lei, Z., Zhong, Y., Chen, S.: Where2comm: Communication-efficient collaborative perception via spatial confidence maps. In: Advances in Neural Information Processing Systems. vol. 35, pp. 4874–4886 (2022)
21. Kunkel, J., Decker, J., Mülhausen, S.: DECICE: Device-Edge-Cloud Intelligent Collaboration Framework (2023)
22. Lu, Y., Hu, Y., Zhong, Y., Wang, D., Wang, Y., Chen, S.: An extensible framework for open heterogeneous collaborative perception. In: Proceedings of the International Conference on Learning Representations (2024)
23. Malinverno, M., Avino, G., Casetti, C., Chiasserini, C.F., Malandrino, F., Scarpina, S.: Performance analysis of c-v2i-based automotive collision avoidance. In: 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM). pp. 1–9 (2018). https://doi.org/10.1109/WoWMoM.2018.8449772
24. Malinverno, M., Avino, G., Casetti, C., Chiasserini, C.F., Malandrino, F., Scarpina, S.: Edge-based collision avoidance for vehicles and vulnerable users: An architecture based on mec. IEEE Vehicular Technology Magazine **15**(1), 27–35 (2020). https://doi.org/10.1109/MVT.2019.2953770
25. Miller, R., Huang, Q.: An adaptive peer-to-peer collision warning system. In: Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367). vol. 1, pp. 317–321 vol.1 (2002). https://doi.org/10.1109/VTC.2002.1002718
26. Oczko, M.C.H., Basaran, O.T., Dressler, F.: Explainable LSTM-Based Cyclist Intention Prediction at Intersections. In: 16th IEEE Vehicular Networking Conference (VNC 2025). IEEE, Porto, Portugal (6 2025). https://doi.org/10.1109/VNC64509.2025.11054246
27. Patriciello, N., Lagen, S., Bojovic, B., Giupponi, L.: An e2e simulator for 5g nr networks. Simulation Modelling Practice and Theory **96**, 101933 (2019). https://doi.org/10.1016/j.simpat.2019.101933
28. Pegurri, R., Gasco, D., Linsalata, F., Rapelli, M., Moro, E., Raviglione, F., Casetti, C.: Van3twin: the multi-technology v2x digital twin with ray-tracing in the loop (2025), https://arxiv.org/abs/2505.14184

29. Pereira, J., Kurunathan, H., Filho, E., Santos, P.M.: Edge-aided v2x collision avoidance with platoons: Towards a hybrid evaluation toolset. In: 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 68–71 (2023). https://doi.org/10.1109/DSN-W58399.2023.00032
30. Raviglione, F., Risma Carletti, C.M., Malinverno, M., Casetti, C., Chiasserini, C.F.: ms-van3t: An integrated multi-stack framework for virtual validation of V2X communication and services. Computer Communications **217**, 70–86 (2024). https://doi.org/10.1016/j.comcom.2024.01.022
31. Riebl, R., Günther, H.J., Facchi, C., Wolf, L.: Artery: Extending Veins for VANET applications. In: 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). pp. 450–456 (2015). https://doi.org/10.1109/MTITS.2015.7223293
32. Risma Carletti, C.M., Casetti, C., Härri, J., Risso, F.: MS-VAN3T-CARLA: An Open-Source Co-Simulation Framework for Cooperative Perception Evaluation. In: 2024 19th Wireless On-Demand Network Systems and Services Conference (WONS). pp. 93–96 (2024). https://doi.org/10.23919/WONS60642.2024.10449533
33. Risma Carletti, C., Raviglione, F., Casetti, C., Stoffella, F., Yilma, G., Visintainer, F.: S-LDM: Server local dynamic map for 5G-based centralized enhanced collective perception. Vehicular Communications **49**, 100819 (2024). https://doi.org/10.1016/j.vehcom.2024.100819
34. Rito, P., Almeida, A., Figueiredo, A., Gomes, C., Teixeira, P., Rosmaninho, R., Lopes, R., Dias, D., Vítor, G., Perna, G., Silva, M., Senna, C., Raposo, D., Luís, M., Sargento, S., Oliveira, A., Carvalho, N.: Aveiro Tech City Living Lab: A Communication, Sensing and Computing Platform for City Environments (07 2022). https://doi.org/10.48550/arXiv.2207.12200
35. Sommer, C., German, R., Dressler, F.: Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. IEEE Transactions on Mobile Computing **10**(1), 3–15 (January 2011). https://doi.org/10.1109/TMC.2010.133
36. Staff writers: Motorcyclist safety: Connected motorcycle pilot. https://imoveaustralia.com/project/motorcyclist-safety-connected-motorcycle-pilot/ (mar 2023)
37. United Nations. Economic Commission for Europe: 2023 Statistics of Road Traffic Accidents in Europe and North America, vol. 58. UN (2023)
38. Wolff, V.A., Xhoxhi, E.: Mitigating vulnerable road users occlusion risk via collective perception: An empirical analysis (2024), https://arxiv.org/abs/2404.07753
39. of Wolverhampton, U.: VRUs and micro-mobility safety services V1 (may 2024). https://doi.org/10.5281/zenodo.16260181
40. Xiang, H., Zheng, Z., Xia, X., Xu, R., Gao, L., Zhou, Z., Han, X., Ji, X., Li, M., Meng, Z., Jin, L., Lei, M., Ma, Z., He, Z., Ma, H., Yuan, Y., Zhao, Y., Ma, J.: V2X-Real: A large-scale dataset for vehicle-to-everything cooperative perception. In: Proceedings of the European Conference on Computer Vision (2024)
41. Xu, R., Tu, Z., Xiang, H., Shao, W., Zhou, B., Ma, J.: CoBEVT: Cooperative bird's eye view semantic segmentation with sparse transformers. In: Proceedings of the Conference on Robot Learning (2022)
42. Xu, R., Xiang, H., Han, X., Xia, X., Meng, Z., Chen, C.J., Correa-Jullian, C., Ma, J.: The openCDA Open-Source Ecosystem for Cooperative Driving Automation research. IEEE Transactions on Intelligent Vehicles **8**(4), 2698–2711 (2023)
43. Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M.H., Ma, J.: V2X-ViT: Vehicle-to-everything cooperative perception with vision transformer. In: Proceedings of the European Conference on Computer Vision (2022)

44. Yun, J., Lim, C., Park, B.: Inherent Limitations of Smartphone GNSS Positioning and Effective Methods to Increase the Accuracy Utilizing Dual-Frequency Measurements. Sensors **22**(24) (2022). https://doi.org/10.3390/s22249879
45. Zhou, Z., Memedi, A., Lee, C., Ucar, S., Altintas, O., Dressler, F.: Fairness-Aware Multi-Agent Learning-based Task Offloading in Dynamic Vehicular Scenarios. In: IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom 2025). pp. 261–269. IEEE, Seoul, South Korea (8 2025). https://doi.org/10.1109/MetaCom65502.2025.00049