CIAK-CP: Camera feed Injection AttacK in Collaborative Perception

Marco Calipari
marco.calipari@tum.de
Technical University of Munich
Munich, Germany

Fabian Maximilian Schmidt
fabian1.schmidt@tum.de
Technical University of Munich
Munich, Germany

Mohammad Hamad
mohammad.hamad@tum.de
Technical University of Munich
Munich, Germany

Sebastian Steinhorst
sebastian.steinhorst@tum.de
Technical University of Munich
Munich, Germany

Abstract

Collaborative Perception (CP) enhances the capabilities of single-vehicle perception by producing collaborative outputs derived from information shared among vehicles. This helps overcome critical issues such as occlusions and sensor range limitations. However, as Connected Autonomous Vehicles (CAVs) achieve higher levels of autonomy, competition for resources becomes a realistic concern, and malicious CAVs may fabricate false information to gain an advantage despite being trusted nodes in the collaborative network. This paper proposes *CIAK-CP*, a framework for data-injection attacks against intermediate, camera-based collaborative perception systems. Our framework enables the extraction of vehicle image snippets from normal driving scenarios, which can later be injected into camera feeds and recognized as authentic vehicles by the perception module, even when they are not present in the real world. We present a detailed analysis of two safety-critical scenarios, a comprehensive evaluation of the attack's effectiveness on 205 samples, and a discussion of state-of-the-art defenses and future directions. Our results reveal a severe limitation in the current CP defense research, and point out a possible direction for researchers to focus on.

CCS Concepts

• **Security and privacy → Distributed systems security**.

Keywords

Collaborative Perception, Security, Connected and Autonomous Vehicles

ACM Reference Format:
Marco Calipari, Fabian Maximilian Schmidt, Mohammad Hamad, and Sebastian Steinhorst. 2026. CIAK-CP: Camera feed Injection AttacK in Collaborative Perception. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), March 23–27, 2026, Thessaloniki, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3748522.3779847

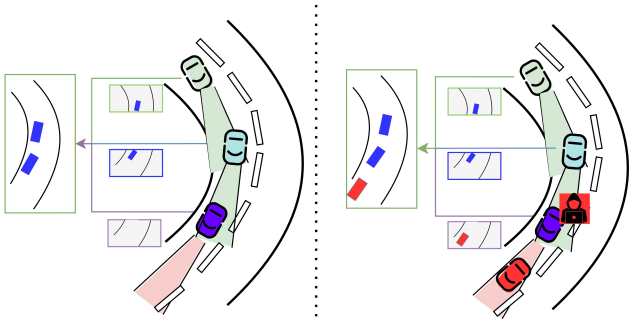


Figure 1: In the presence of a dishonest participant in the CP network, injecting false information for the sake of maximizing the resources is a concrete risk. If the attacker (purple vehicle) is able to inject false data inside its own camera feed, it can result in a critical threat for other road users performing other maneuvers, such as overtaking.

1 Introduction

CAVs have experienced a significant surge in popularity over the last decade, culminating in the development of the first fully autonomous vehicles that do not require human intervention [22]. What enables higher levels of autonomous driving (L3-L4-L5, as defined by [13]) is the *Perception* module [28]. This module serves as the initial stage in the autonomous driving pipeline [18] and is fundamentally responsible for tasks such as object detection, classification, and semantic segmentation [27].

Despite advancements, the perception capabilities of a single CAV are inherently constrained by its onboard sensors. Limitations such as *occlusions* (where objects are hidden from view) or a *limited sensing range* can significantly degrade the performance of detection algorithms [33]. To overcome these challenges, the concept of CP has been proposed [9]. By leveraging Vehicle To Everything (V2X) communication, CAVs can share their perception

SAC '26, Thessaloniki, Greece

ACM ISBN 979-8-4007-2294-3/2026/03
https://doi.org/10.1145/3748522.3779847

data, effectively extending the individual sensing range of each vehicle in the network. This shared awareness allows for the observation of a larger area and can preemptively identify dangers that might be occluded from a single vehicle's perspective.

When it comes to security, many Cyber-Physical Systems (CPS) rely mainly on the security of their V2X communication. While V2X has been well studied, and many solutions have been proposed to prevent attackers from intercepting or modifying messages [1], most of this research focuses only on external threats. However, these studies often ignore a critical case, when the attacker is actually one of the collaborative peers. This risk comes from the nature of CAV systems, where many vehicles may collaborate, but still compete with each other for road space, priority, or other resources [8]. In such cases, a vehicle may choose to falsify its data while still using valid V2X credentials, making the attack hard to detect. This has been proven true in Light Detection and Ranging (LiDAR) based perception in [38], concluding that attacking directly the source of information can defy most of the defenses developed.

Motivated by this, we turn our attention to CPS systems that rely on camera-based perception. Cameras are widely used due to their low cost, ease of deployment, and rich visual information. In fact, some commercial systems already use vision-only perception to support autonomous driving tasks such as Tesla [3]. Cameras also play a central role in collaborative perception [7, 11, 14, 25, 37], where vehicles share images or visual detections to improve their understanding of the environment. However, similar to LiDAR, camera data can be manipulated at the source. A malicious vehicle can alter its own image before sharing it, without breaking any communication protocols or cryptographic protections. Despite the growing use of camera-based collaborative systems, there are no existing attacks that specifically target this type of setup. In this paper, we make the following contributions:

- We introduce, to the best of our knowledge, the first data injection attack for intermediate camera-based CP, elaborated further (Section 3). To perform our attack, we developed `CIAK-CP`[1], an end-to-end framework that executes the steps required to inject an adversarial image aiming to cause a bogus detection within the camera feed of the attacker, to be shared and potentially cause danger to the normal traffic circulation.
- We evaluate `CIAK-CP` against CoBEVT [34], a state-of-the-art CP framework, as a case study to prove our hypothesis (Section 4). Additionally, we propose and discuss defense mechanisms based on the attack scenarios (Section 5).

2 System and Threat Model

In this section, we present CP in subsection 2.1, the system model in Subsection 2.2, and the Threat model in Subsection 2.3

2.1 Collaborative Perception

CP extends the perception capabilities of a single vehicle by incorporating viewpoints from multiple vehicles and, therefore, reduces risks and enhances the inference capabilities of CAVs. Similar to sensor fusion, CP distinguishes between different stages of collaboration, depending on which data are exchanged (see Figure 2).

[1]code available at https://github.com/tum-esi/CIAK

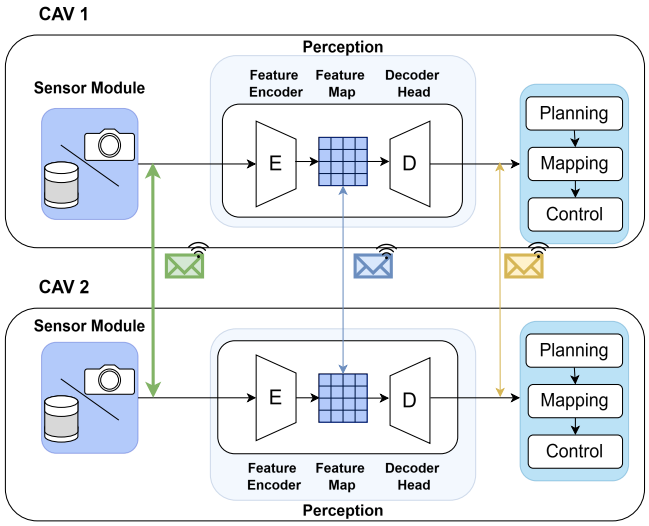


Figure 2: Conceptual scheme of collaborative perception between two CAVs. Sensor data (cameras, LiDARs) flow into the Perception Module, where the encoder (E) generates a *Feature Map* and the decoder (D) outputs results for Planning, Mapping, and Control. Collaboration is color-coded: green (Early), blue (Intermediate), yellow (Late), with line width showing bandwidth demand.

With *early* CP, the CAVs will fuse the raw sensor readings from other CAVs before the encoding stage [6, 10, 26, 31]. Despite being the most informative, early CP requires a higher communication bandwidth because the sensor measurements are large and minimally processed. Consequently, this generates a transmission overhead for the V2X network. In *intermediate* CP, each CAV extracts features locally and shares the resulting *feature map* with others [17, 20, 24, 30, 36, 39, 41]. A feature map is a lower-dimensional representation of the original source, in which features from the higher-dimensional input are encoded, thereby drastically reducing message dimensions; however, this approach may introduce information loss. Finally, in *late* CP, each CAV exchanges only the result of the local perception (i.e., the bounding boxes, segmentation maps) [4, 15, 21]. This stage is the most efficient in terms of bandwidth requirements, since it sends only compact outputs (such as bounding box coordinates); however, it is more error-prone and can be trivially exploited [29, 38].

2.2 System model

We consider a CP system with a set of CAVs denoted as $V = \{v_1, \ldots, v_n\}$, where $n \geq 2$. These CAVs operate in an environment $\mathcal{E} \subset \mathbb{R}^d$ with $d \geq 2$, where d denotes the dimensionality of the sensed data. The CAVs collaborate using intermediate CP by sharing feature maps after local feature extraction. Each vehicle fuses its own feature map with those received from others and inputs the result into a decoder, which outputs the final CP result. In this work, we assume that each one of the CAVs is equipped with the same CP framework, specifically CoBEVT [34]. Additionally, we assume

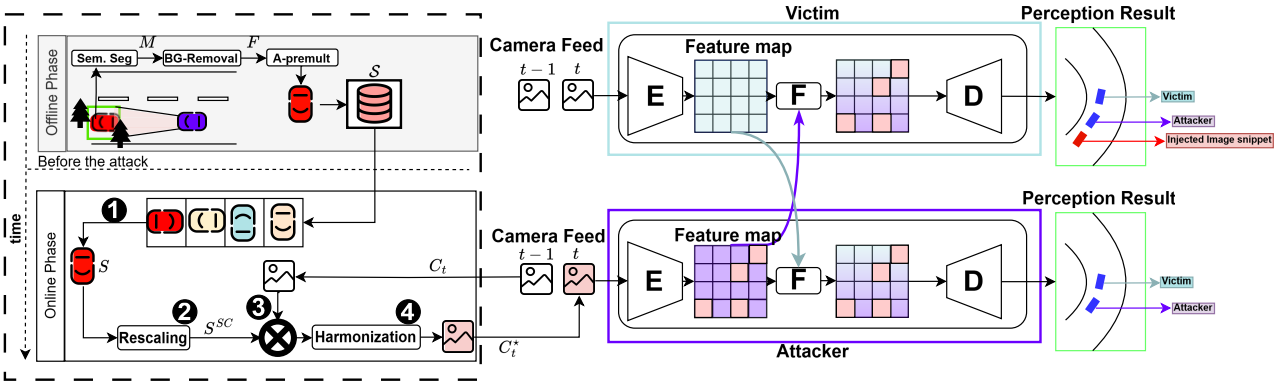


Figure 3: Online phase of CIAK-CP. In step ❶, the attacker will select the snippet to inject into its own view, scale it in ❷, and composite it within the frame C_t in step ❸. Finally, after the harmonization step ❹, the snippet is injected within the original frame and fed into the *Feature Encoder* E. Consequently, the compromised feature map will be transmitted, leading the injected image snippet to be recognized as authentic, compromising the CP result

that all inter-vehicle communications are secure and authenticated, ensuring the confidentiality and integrity of the exchanged feature maps [19].

To formally define sensor coverage, we adopt the concept of *Field Of View (FoV)* from the SAE-J1050 standard [5]. In this work, the FoV of a camera sensor S on a CAV v_i at time t denoted $\mathcal{F}_i(t)$ is the set of points $(x, y) \in \mathcal{E}$ that lie within the camera's sensing range r and horizontal viewing angle α. The goal of CP is to expand the FoV of each CAV by merging information from multiple vehicles. We define this as the *collaborative FoV*:

$$\mathcal{F}_{\text{collab}}(t) = \bigcup_{i=1}^{n} \mathcal{F}_i(t) \tag{1}$$

where $\mathcal{F}_i(t)$ is the individual FoV of CAV v_i at time t.

We divide the set of CAVs into two groups: honest vehicles V_h and attacking vehicles V_a, such that $V = V_h \cup V_a$. The collaborative FoV then becomes:

$$\mathcal{F}_{\text{collab}}(t) = \left(\bigcup_{v \in V_h} \mathcal{F}_v(t) \right) \cup \left(\bigcup_{a \in V_a} \mathcal{F}_a(t) \right) \tag{2}$$

Also, we assume that each attacker CAV $a \in V_a$ is positioned ahead of the honest CAV $v_i \in V_h$, i.e., $(x_{v_i}, y_{v_i}) < (x_a, y_a)$. This reflects a realistic scenario where an attacker may obstruct the victim's view. Finally, each CAV produces a shared perception result $Y(t, X)$ at time t, modeled as:

$$Y(t, X) = P(X(t, \mathcal{F})) = D(F(E(X(t, \mathcal{F})))) \tag{3}$$

where $X(t, \mathcal{F})$ is the sensor input, E is the Encoder, F is the fusion function, and D is the Decoder. This chain defines the collaborative perception pipeline from individual sensor input to collaborative final output.

2.3 Threat Model

We consider an adversary to be an authenticated, *dishonest* CAV within the CP network. Concretely, the adversary is fully compliant with the network's Identity and Access Management (IAM) and secure channel establishment mechanisms, but it retains complete control over its own *sensor module*. This gives the attacker full access to its local sensor stream and the ability to manipulate that stream arbitrarily *prior* to any transmission to other CAV.

In this work, we restrict the adversary to digital-only manipulations of its own sensor feed, without any physical tampering with other agents or their sensors. Permitted attacks include virtual object insertion (spoofing) and virtual object removal (occlusion), which may depend on the victim's transmitted state (for example, the relative pose) as in [38]. Also, we assume that the attacker has black-box access to the perception stacks. Thus, it does not know model weights, architectures, or feature maps. It is very important to note that the attacker will not modify the internal design or implementation of the Encoder, Decoder, or fusion algorithms, neither for itself nor for the victims. In addition, since we assume that all inter-CAV communications are secure and authenticated, we exclude external adversaries who could tamper with, forge, or inject feature maps into legitimate messages.

The attacker's goal is to disrupt the CP result by injecting, removing, or extending existing objects that are in its own FoV. Our case study evaluation in Section 4.2 happens considering two safety-critical scenarios: I) *Occlusion attack* where the attacker is positioned in front of the victim and exploits this positional advantage to hide or insert objects; II) *Sensing range limitation exploitation* where the attacker can take advantage of the victim's limited FoV where imminent objects are beyond the victim's sensing range. In this case, the attacker can insert or remove objects, leaving the victim unable to verify the manipulation. Even though the collaborative perception output may contain injected objects, the attacker, having access to its own ground truth, will ignore them, leaving only the victim affected.

3 CIAK-CP Framework

In this section, we present CIAK-CP, an open-source data injection framework for intermediate camera-based CP, which operates in two phases: offline and online. The framework and its two phases are presented in Figure 3.

Algorithm 1 CIAK-CP Offline Phase

Require: Image set IS (cars in normal traffic)
Ensure: Snippet library $\mathcal{S}$
1: **for all** $Img \in IS$ **do**
2: $M \leftarrow \text{U}^2\text{-Net}(Img)$ % semantic seg., class *Car*
3: $F \leftarrow Img \odot M$ % remove background
4: $S \leftarrow \text{alpha\_premultiply}(S)$ % avoid white halos
5: add S to $\mathcal{S}$
6: **end for**
7: **return** $\mathcal{S}$

3.1 Offline phase

During the offline phase, described in detail in Algorithm 1, the adversary prepares a local library of vehicle snippets for later use. The process begins by collecting a dataset IS of car images captured in normal traffic scenarios. Each image is processed using a semantic segmentation model, specifically U^2-*Net* [23], line 2–3, pretrained for the Car class, to isolate the vehicle silhouette and remove the background. The segmented foreground is then cropped into an RGBA snippet, line 4, and an alpha-premultiplication step is applied to eliminate unwanted white halo artifacts that can appear when compositing, shown in line 5. The resulting snippets are stored in the local repository $\mathcal{S}$ for later injection, as shown in line 6. If system resources are limited, a predefined set of snippets can be loaded instead of generating new ones.

3.2 Online Phase

During this phase, the adversary executes the attack in real time by injecting pre-generated vehicle snippets into live camera frames. Snippets are scaled and placed to match the scene both geometrically and visually. This phase is described in Algorithm 2 and shown in Figure 3. To instantiate the attack within a camera frame $C_t^\star$ at time t, the adversary first selects a frame from the camera feed C_{t-1} at time $t-1$. A snippet is then chosen from the snippet collection in step ❶, and rescaled in step ❷. When the intrinsic camera parameters are available (i.e., the calibration matrix K, which converts 3D camera coordinates to 2D homogeneous image coordinates under the pinhole camera model) and the desired placement distance from the front camera is known, the snippet width in pixels is computed using the focal length from K, the intended placement distance d, and the real-world object width W, as shown in lines 3–8 of Algorithm 2:

$$w_{px} = f_{px} \cdot \frac{W}{d} \tag{4}$$

Because the injected object is always a vehicle, canonical vehicle dimensions may be used as an approximation for W. In scenarios where the attack is executed at a fixed distance, the snippet can be alternatively rescaled using a predefined bounding box. After rescaling, the snippet is composited with the original image in step ❸, line 9 and then harmonized via histogram normalization in step ❹; line 10. Finally, the processed (tampered) image $C_t^\star$ is inserted into the feed and processed by the CP framework. To ensure temporal consistency, the snippet is injected continually within a sequence, maintaining a constant distance, allowing the attacker to maintain consistency both in space and time.

Algorithm 2 CIAK-CP Online Phase

Require: Snippets $\mathcal{S}$ (premult., bg removed), optional intrinsics K, width W
Ensure: $C_t^\star$
1: Acquire frame C_{t-1} % at time t
2: Pick $S \in \mathcal{S}$
3: **if** K, W, and d known **then** % transformation
4: $w_{\text{px}} \leftarrow f_{px} \cdot W/d$
5: Scale $S \rightarrow S^{\text{sc}}$ to w_{px}
6: **else**
7: Scale $S \rightarrow S^{\text{sc}}$ to (w_0, h_0)
8: **end if**
9: Composite S^{sc} into C_t at $d \rightarrow C_t^\star$
10: Harmonize S^{sc} to C_t % using histogram normalization
11: **return** $C_t^\star$

4 Implementation and Evaluation

We instantiate `CIAK-CP` against CoBEVT [34], a state-of-the-art camera-based intermediate CP framework for collaborative BEV semantic segmentation trained on OPV2V [35], a CARLA-derived dataset. In a multi-view configuration, each vehicle mounts four RGB cameras (front, rear, left, right) to provide 360° coverage. Vehicles exchange BEV feature maps that CoBEVT fuses via Fused Axial Attention to produce a single, consistent Bird's Eye View (BEV) over lanes (static) and vehicles (dynamic). Our objective is to induce false vehicle segments in the collective *dynamic* BEV output by tempering one of the camera feed views. This will introduce one of the multi-view features, the injected vehicle, resulting in a tampered fused map, which will induce the vehicle to be recognized as an authentic vehicle within the scene.

For semantic segmentation tasks, the standard evaluation metric used is the Intersection Over Union (IoU), which measures pixel-level consistency between model predictions and ground truth. Unlike detection tasks that rely on bounding boxes, BEV segmentation produces dense class maps where each pixel is assigned to a semantic class. Accordingly, IoU, as shown in Eq. (5) evaluates the fraction of correctly classified pixels over the union of predicted and true class regions.

$$IoU = \frac{|\text{Prediction} \cap \text{GT}|}{|\text{Prediction} \cup \text{GT}|} \tag{5}$$

CoBEVT provides two separate models: (i) a static model for segmenting Street and Lane classes, and (ii) a dynamic model for segmenting the Car class. The dynamic BEV output is thus a binary mask labeling pixels as either Car or Not-Car. If the adversarial snippet we inject is successfully recognized as a vehicle, it manifests as a decrease in IoU relative to the benign case.

We deem an attack successful if, for a given frame, the relative dynamic IoU degrades compared to the benign baseline [2], as in Eq. (6). Formally, given benign IoU IoU_b (no injection) and adversarial IoU IoU_a, we compute:

$$\Delta IoU_r = \frac{(IoU_b - IoU_a)}{IoU_b}. \tag{6}$$

We consider the attack successful when the decrease in ΔIoU_r is at least 10%. This indicates that the injected vehicle is being classified

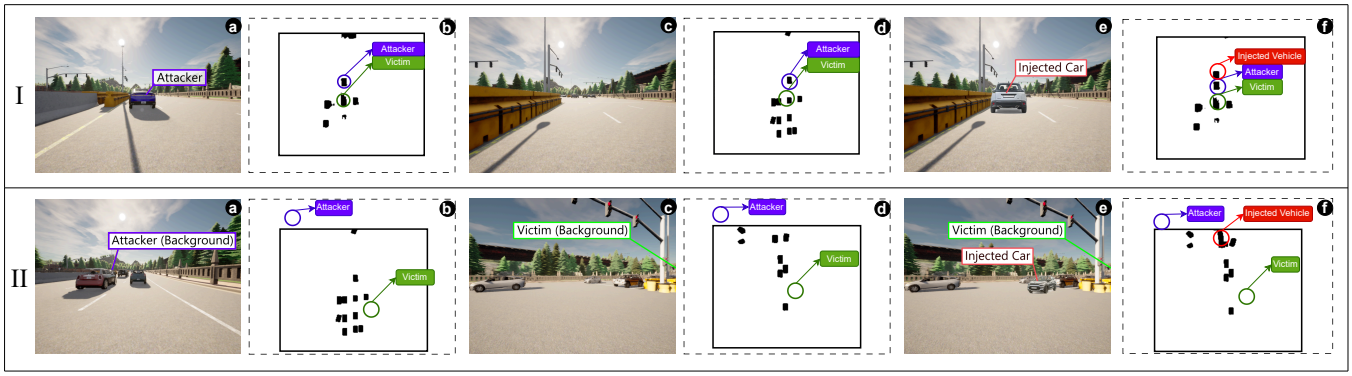
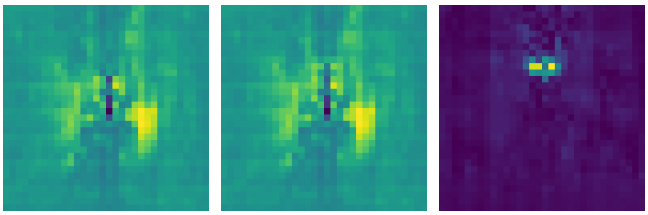


Figure 4: Scenario I is reported on the top half of the figure, whereas scenario II is on the bottom half. From left to right: *a* shows the victim's PoV, occluded by the attacker; *b*, the ground truth provided from CoBEVT; *c*, the attacker's PoV with corresponding collaborative semantic segmentation in *d*. Subfigure *e* displays the attacker's tampered feed with the injected car, and the corresponding successful attack is shown in subfigure *f*.



(a) Benign Feature Map **(b) Attack Feature Map** **(c) Difference Map**

Figure 5: Comparison of a feature map without (Figure 5a) and with (Figure 5b) the effect of CIAK-CP. Figure 5c shows the effect of CIAK-CP in the output of the fusion model. By changing one of the multiview feature maps, the attack propagates across the other encoding layers, causing misdetection at the decoder head.

or segmented under the Car class, and that the resulting deviation in segmentation performance exceeds the model's baseline variability.

4.1 Experimental Setup

To demonstrate the requirements of our framework CIAK-CP, we evaluated the framework on a workstation equipped with an Intel Xeon Gold 5220R CPU, a PNY NVIDIA RTX A5000 GPU running CUDA 12.8, and Ubuntu 24.04.3. The inference is performed using the pre-trained models provided by the CoBEVT developers, trained on 90 epochs.

4.2 Case Study

To showcase the effect of CIAK-CP, we extracted two safety-critical scenarios from the test dataset, depicted in Figure 4 to highlight the attack's impact. Namely, we considered a full attacker obstruction, with the attacker being in front of the victim described in § 4.2.1, and occlusion by non-collaborative vehicles at an intersection described in § 4.2.2.

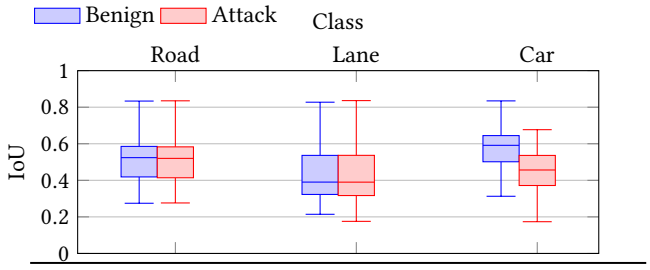
4.2.1 Scenario I: Attacker-Induced Occlusion. Row I illustrates the first attack scenario, which exploits a fundamental limitation of line-of-sight perception in multi-agent cooperative perception systems. In this setup, the victim vehicle is positioned directly behind the attacker and aligned in the same direction, as shown in a. Consequently, the attacker's physical body naturally occludes part of the victim's forward field of view. The attacker, facing away from the victim, is thus strategically placed to conceal portions of the surrounding environment.

To exploit this occlusion, we inject a synthetic white vehicle into the attacker's *frontal* camera view; as shown in c. This positioning ensures that the injected object occupies a region invisible to the victim's sensors due to the attacker's physical obstruction. As a result, the collaborative BEV map includes a phantom vehicle—perceived as existing directly in front of the attacker—since the injected snippet is encoded within the attacker's feature map.

This behavior is confirmed in Figure 5, where we compare the fusion network's output under benign (Figure 5a) and malicious (Figure 5b) conditions. The difference map (Figure 5c) validates our hypothesis: the injected snippet is successfully embedded within the feature representation, and the encoder network correctly classifies it as a "Car." Notably, f shows that this phantom object appears despite its absence in the physical environment, but is present in the tampered feed e

We also observe a discrepancy between the dataset-provided Ground Truth (GT) map in b, and the benign collaborative BEV output in d. This inconsistency further motivates our use of the *relative IoU degradation* as a robust performance metric to quantify the impact of our attack.

4.2.2 Scenario II: Occlusion by Surrounding Traffic. Row II of Figure 4 depicts the second attack scenario, which leverages occlusions caused by surrounding traffic rather than the attacker's own body. Specifically, we model a common urban driving situation where a queue of vehicles forms at an intersection. In this configuration, the victim's forward-facing sensors are obstructed by intervening vehicles; depicted in a, while the attacker, positioned with a



Class	Road	Lane	Car
$\overline{IoU_b}$	0.537993993	0.440453112	0.570402509
$\overline{IoU_a}$	0.53425954	0.437181687	0.448499767
$\Delta IoU_a(\%)$	0.782194009 %	0.975982574 %	21.45225977 %

Figure 6: Results of the attack effectiveness evaluation. In blue we have the IoU from the runs without data injection, which are the baseline of our comparison, and in red the runs with `CIAK-CP` in place. It is possible to see that the IoU of the class Car degrades noticeably, whereas for the other classes it remains practically stable. The table describes the average degradation among all the classes, with the Road and Lane class remaining under 1%, which is an expected variability caused by the stochastic nature of AI. On the other hand, the Car class degraded by over 21%.

clear line of sight beyond the queue in ⓒ, retains visibility of the unobstructed roadway ahead.

As before, we inject a vehicle snippet into the attacker's frontal view ⓔ, placing it immediately ahead of the occluding traffic. By design, this injected object remains invisible to the victim's sensors but is treated by the attacker as a genuine detection. The resulting collaborative BEV map mirrors the outcome of Scenario I: the benign map ⓓ, lacks the phantom object, whereas the adversarial map ⓕ includes it. For reference, the dataset-provided collaborative ground-truth BEV map is shown in ⓑ.

4.3 Results

We evaluated our attack on 205 samples from the test dataset of OPV2V. For each of the frames, we computed ΔIoU_r for all the classes included in CoBEVT. To ensure the success of our attack, we evaluated both the `Static` and `Dynamic` segmentation to ensure that the ΔIoU_r degradation was caused by the injection result, and not from poor performances of the model. Concerning the `static` segmentation, for both the `Road` and `Lane`, the outliers present are the byproduct of the stochastic nature of the AI model. However, for the `Dynamic` segmentation, the tampered feeds substantially degrade the IoU by approximately 22%. This result, shown in Figure 6, confirms the efficacy of our framework in injecting highly believable vehicles within a camera feed, and to alter the sole result of the `dynamic` segmentation. While we decided to concentrate our work on CoBEVT due to its popularity, with over 340 citations as of October 2025, we are confident that our framework's effectiveness also extends to other CP frameworks.

5 Possible Mitigation

To mitigate `CIAK-CP`, under the system and threat model we provided, there are different possible scenarios, depending on the attack setting, and in particular, the $\mathcal{F}_{\text{collab}}$, and the number of honest collaborators available within the collaborative perception network. This section aims to analyse the defense problem under variable conditions.

Consider a region $\mathcal{R}(t)$ that encompasses the overlapping regions of the FoV of the attacking vehicles ($V_a(t)$) and the honest-near-attacker vehicles ($V_{hn}(t)$) at time t. The set of honest-near-attacker vehicles is defined $V_{hn}(t) = \{\, u \in V_h \mid \exists a \in V_a : \mathcal{F}_u(t) \cap \mathcal{F}_a(t) \neq \varnothing \,\}$. The region $\mathcal{R}(t)$ is then given by:

$$\mathcal{R}(t) = \Big(\bigcup_{v \in V_{hn}} \mathcal{F}_v(t) \Big) \cap \Big(\bigcup_{a \in V_a} \mathcal{F}_a(t) \Big) \tag{7}$$

In relation to this, we also define the area in sole control of the attacker as

$$\mathcal{A}(t) = \Big(\bigcup_{a \in V_a} \mathcal{F}_a(t) \Big) \setminus \Big(\bigcup_{u \in V_{hn}(t)} \mathcal{F}_u(t) \Big) \tag{8}$$

Based on this definition, we can compare different scenarios for possible mitigations.

5.1 Case 1: $\mathcal{R} \neq \emptyset$ and the injected object lies inside $\mathcal{R}$

If the overlapping region is not empty, this case corresponds to the setting described in [38]. In this scenario, the ratio between honest and attacker vehicles plays a crucial role. If the number of honest-near-attacker vehicles exceeds that of attackers, i.e., $| V_{hn} | > | V_a |$, the system remains resilient. Here, more than two CAVs are present, and the shared region $\mathcal{R}$ is non-empty. Each CAV constructs an *Occupancy Map* to distinguish between free and occupied spaces. If one vehicle perceives a region as free while another perceives the same region as occupied, this inconsistency indicates a potential data fabrication attempt.

However, this mitigation strategy is effective only when the injected snippet S is placed within the region $\mathcal{R}$. When the number of attacking vehicles exceeds that of honest ones, deploying any defense strategy becomes practically infeasible, as discussed in more detail in Section 5.3. This is because the portion of the FoV controlled by the attacker, defined in Eq. (8), allows the injection of the snippet into a blind spot for all honest vehicles, thereby preventing detection.

Importantly, the attacker's relative position does not affect the detection capability: the existence of at least one additional honest CAV guarantees verification. This leads to our first observation:

> **Observation 1:** Redundancy is essential for mitigating data fabrication attacks. The greater the number of honest vehicles observing the same region, the easier it becomes to detect inconsistencies. Moreover, increased overlap in sensing reduces the attack surface and strengthens detection guarantees.

5.2 Case 2: $\mathcal{R} \neq \emptyset$ and the injected object lies inside $\mathcal{A}$

When the injected object lies within $\mathcal{A}$, even if some overlap exists (i.e., $\mathcal{R} \neq 0$) with honest-near-attacker vehicles, the attackers have full control over this area. Consequently, the victim cannot rely on other honest-near CAVs to confirm or contradict the fabricated information. In this setting, data fabrication attacks remain undetectable within $\mathcal{A}$, despite the use of technologies such as anomaly detection systems or sensor fingerprinting.

A highly skilled and well-resourced attacker group might generate a high-fidelity digital object that introduces no anomalies in the final composite image while preserving all necessary features to evade fingerprinting. However, this is possible only if the extent of $\mathcal{A}$ is sufficient for performing the injection. If $\mathcal{R} \gg \mathcal{A}$, such an attack may not be practically feasible. This leads to our second observation:

> **Observation 2:** Control over a non-overlapping region enables undetectable data fabrication. The smaller the attack-controlled area relative to the shared region, the lower the feasibility of a successful injection. Ensuring that $\mathcal{R}$ dominates $\mathcal{A}$ limits the attacker's capacity to conceal fabricated data.

5.3 Case 3: $\mathcal{R} = \emptyset$

This corresponds to the worst-case scenario, where the honest-near-attacker vehicles present a blind spot that is not limited to a portion of the attacker's collaborative FoV but encompasses its entirety, i.e., $\mathcal{A} = \bigcup_{a \in V_a} \mathcal{F}_a(t)$. In this situation, the considerations made in 5.1 do not hold, as it becomes impossible to construct an occupancy map due to the absence of any overlap between the honest and attacker FoVs. This eliminates redundancy.

Similarly, the considerations in 5.2 also fail to apply. Without any secondary source of information, neither an occupancy map nor any other traditional defense mechanism can be built, making it impossible to verify the integrity of a single camera feed. In this setting, data fabrication attacks remain undetectable. This leads to our final observation:

> **Observation 3:** When $\mathcal{R} = \emptyset$, redundancy is lost, and no defense mechanism based on cross-verification can be deployed.

6 Discussion and Related Work

While we acknowledge previous studies tackling data fabrication attacks [38], we are the first to investigate it specifically for camera-based intermediate CP. Although intermediate collaboration offers benefits such as reduced bandwidth requirements and independence from specific sensor modalities, it remains vulnerable to attacks that temper the single CAV raw data. This vulnerability is not unexpected: when an attacker injects an object into its own sensing domain and the encoder subsequently detects it, the object becomes embedded in the resulting feature map, as shown previously in Figure 5, thereby compromising the shared representation.

Several defense mechanisms have been proposed for intermediate CP based on LiDARs, including the comparison of feature maps against benign references [40] and consensus-based validation schemes [16]. However, these approaches inherently assume the availability of a reliable reference or a sufficient number of honest collaborators capable of verifying falsified information. We argue that this assumption highlights a fundamental limitation of intermediate CP, where the very properties that confer efficiency and scalability simultaneously impose constraints that undermine robustness in safety-critical scenarios.

In contrast, early CP could enable the development of defense mechanisms that exploit data-specific characteristics, because by design it requires the fusion of the raw data captured by the sensors. Camera data, in contrast to other sensor data, such as LiDAR, offers a potential alternative. To the best of our knowledge, only one prior work [12] has explored early collaboration among camera-equipped vehicles, employing image stitching techniques [32]. To confirm the claims of the authors, we re-implemented the pipeline described in [12], as well as evaluating the performances of pre-implemented stitching algorithms such as `PANORAMA`, and `SCANS`, from the Python library OpenCV.

We generated controlled test scenes in CARLA. Starting from a simple configuration—two vehicles positioned side by side and oriented in the same direction—we applied all three pipelines. Even in this benign setting, stitching failed: our re-implementation exhibited severe ghosting without expanding the field of view, while OpenCV produced no valid panorama. With the two cameras separated by approximately 3 m, the induced parallax could not be reconciled by a homography, resulting in visible artifacts and invalid composite outputs. We therefore conclude that current image-stitching techniques are unsuitable for camera-based CAV scenarios and, consequently, that early camera-based CP remains impractical with existing methodologies, and that the only way to do CP is by following the intermediate paradigm.

Nevertheless, future research should focus on developing defense mechanisms that exploit the intrinsic characteristics of exchanged data.

7 Conclusion

In this paper, we introduced `CIAK-CP`, a two-step (offline and online) framework for conducting data fabrication attacks in intermediate collaborative perception through snippet injection. We evaluated `CIAK` against CoBEVT, an intermediate collaboration framework that fuses multi-view feature maps to generate a BEV semantic segmentation map. We evaluated our attack in 205 frames from the OPV2V dataset, demonstrating a substantial decrease in the relative ΔIoU for the class Car, but not for the class Lane or Road. We further discussed possible mitigations, based on the collaborative FoV, and the number of honest vehicles in the scene, concluding that, while redundancy-based mechanisms can mitigate some attacks, they fail in situations where there is not an overlapping region of the FoVs, or if the spoofed object lies within the area fully controlled by the attacker. Finally, we discussed the state of the art in camera-based CP, emphasizing the critical role of raw data in developing robust defenses. We hope this work encourages the community to advance beyond current limitations and design security mechanisms that safeguard collaborative perception systems against fabrication attacks.

Acknowledgments

This work is supported by the Horizon Europe project PANDORA under grant agreement number 101135775.

References

[1] Aljawharah Alnasser, Hongjian Sun, and Jing Jiang. 2019. Cyber security challenges and solutions for V2X communications: A survey. *Computer Networks* 151 (2019), 52–67.

[2] Anurag Arnab, Ondrej Miksik, and Philip HS Torr. 2018. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 888–897.

[3] Benjamin Bauchwitz and M. L. Cummings. 2020. *Evaluating Reliability of Tesla Model 3 Driver Assist Functions.* Research Report CSCRS-R{X}. Duke University. Humans and Autonomy Laboratory. https://rosap.ntl.bts.gov/view/dot/73716 Corporate contributors: Collaborative Sciences Center for Road Safety; United States Department of Transportation: Federal Highway Administration, University Transportation Centers (UTC) Program, Office of the Assistant Secretary for Research and Technology.

[4] Bernardo Camajori Tedeschini, Mattia Brambilla, Luca Barbieri, Gabriele Balducci, and Monica Nicoli. 2023. Cooperative Lidar Sensing for Pedestrian Detection: Data Association Based on Message Passing Neural Networks. *IEEE Transactions on Signal Processing* 71 (2023), 3028–3042. https://doi.org/10.1109/TSP.2023.3304002

[5] Driver Vision Standards Committee. 2009. J1050_200902: Describing and Measuring the Driver's Field of View. https://doi.org/10.4271/J1050_200902 SAE Standard.

[6] Siqi Fan, Haibao Yu, Wenxian Yang, Jirui Yuan, and Zaiqing Nie. 2024. QUEST: Query Stream for Practical Cooperative Perception. arXiv:2308.01804 (May 2024). https://doi.org/10.48550/arXiv.2308.01804 arXiv:2308.01804 [cs].

[7] Xiangbo Gao, Yuheng Wu, Rujia Wang, Chenxi Liu, Yang Zhou, and Zhengzhong Tu. 2025. LangCoop: Collaborative Driving with Language. (4 2025). https://arxiv.org/abs/2504.13406v2

[8] Mohammad Hamad, Christian Prehofer, Mikael Asplund, Tobias Löhr, Lucas Bublitz, Alexander Zeh, Mridula Singh, and Sebastian Steinhorst. 2025. Cybersecurity Challenges of Autonomous Systems. In *2025 Design, Automation & Test in Europe Conference (DATE)*. IEEE, 1–10.

[9] Yushan Han, Hui Zhang, Huifang Li, Yi Jin, Congyan Lang, and Yidong Li. 2023. Collaborative perception in autonomous driving: Methods, datasets, and challenges. *IEEE Intelligent Transportation Systems Magazine* 15, 6 (2023), 131–151.

[10] Yushan Han, Hui Zhang, Huifang Li, Yi Jin, Congyan Lang, and Yidong Li. 2023. Collaborative Perception in Autonomous Driving: Methods, Datasets, and Challenges. *IEEE Intelligent Transportation Systems Magazine* 15, 6 (Nov. 2023), 131–151. https://doi.org/10.1109/MITS.2023.3298534

[11] Suozhi Huang, Juexiao Zhang, Yiming Li, and Chen Feng. 2024. ActFormer: Scalable Collaborative Perception via Active Queries. *2025 ICRA* (2024). https://doi.org/10.1109/ICRA57147.2024.10610997

[12] Manzoor Hussain, Nazakat Ali, and Jang-Eui Hong. 2022. Vision beyond the field-of-view: A collaborative perception system to improve safety of intelligent cyber-physical systems. *Sensors* 22, 17 (2022), 6610.

[13] International Organization for Standardization. 2020. *Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles.* Publicly Available Specification 22736. ISO. https://www.iso.org/standard/73766.html ISO/SAE PAS 22736:2020.

[14] Lantao Li, Yujie Cheng, Chen Sun, and Wenqi Zhang. 2024. ICOP: Image-based Cooperative Perception for End-to-End Autonomous Driving. *IEEE Intelligent Vehicles Symposium, Proceedings* (2024), 2367–2374.

[15] Yiming Li, Qi Fang, Jiamu Bai, Siheng Chen, Felix Juefei-Xu, and Chen Feng. 2023. Among Us: Adversarially Robust Collaborative Perception by Consensus. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Paris, France, 186–195. https://doi.org/10.1109/ICCV51070.2023.00024

[16] Yiming Li, Qi Fang, Jiamu Bai, Siheng Chen, Felix Juefei-Xu, and Chen Feng. 2023. Among us: Adversarially robust collaborative perception by consensus. In *ICCV*. 186–195.

[17] Genjia Liu, Yue Hu, Chenxin Xu, Weibo Mao, Junhao Ge, Zhengxiang Huang, Yifan Lu, Yinda Xu, Junkai Xia, Yafei Wang, and Siheng Chen. 2024. Towards Collaborative Autonomous Driving: Simulation Platform and End-to-End System. *arXiv preprint arXiv:2404.09496* (4 2024). https://doi.org/10.48550/arXiv.2404.09496 arXiv:2404.09496 [cs].

[18] Shaoshan Liu, Jie Tang, Zhe Zhang, and Jean-Luc Gaudiot. 2017. Computer architectures for autonomous driving. *Computer* 50, 8 (2017), 18–25.

[19] Muhana Magboul Ali Muslam. 2024. Enhancing Security in Vehicle-to-Vehicle Communication: A Comprehensive Review of Protocols and Techniques. *Vehicles* 6, 1 (2024), 450–467.

[20] Zhenyang Ni, Zixing Lei, Yifan Lu, Dingju Wang, Chen Feng, Yanfeng Wang, and Siheng Chen. 2024. Self-Localized Collaborative Perception. arXiv:2406.12712 (June 2024). https://doi.org/10.48550/arXiv.2406.12712 arXiv:2406.12712 [cs].

[21] Zhenyang Ni, Zixing Lei, Yifan Lu, Dingju Wang, Chen Feng, Yanfeng Wang, and Siheng Chen. 2024. Self-Localized Collaborative Perception. *arXiv* arXiv:2406.12712 (6 2024). https://doi.org/10.48550/arXiv.2406.12712 arXiv:2406.12712 [cs].

[22] Sina Nordhoff, John D Lee, Simeon C Calvert, Siri Berge, Marjan Hagenzieker, and Riender Happee. 2023. (Mis-) use of standard Autopilot and Full Self-Driving (FSD) Beta: Results from interviews with users of Tesla's FSD Beta. *Frontiers in psychology* 14 (2023), 1101520.

[23] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaiane, and Martin Jagersand. 2020. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition* 106 (2020), 107404. https://doi.org/10.1016/j.patcog.2020.107404

[24] Deyuan Qu, Qi Chen, Tianyu Bai, Hongsheng Lu, Heng Fan, Hao Zhang, Song Fu, and Qing Yang. 2024. SiCP: Simultaneous Individual and Cooperative Perception for 3D Object Detection in Connected and Automated Vehicles. arXiv:2312.04822 (Aug. 2024). https://doi.org/10.48550/arXiv.2312.04822 arXiv:2312.04822 [cs].

[25] Rui Song, Chenwei Liang, Hu Cao, Zhiran Yan, Walter Zimmer, Markus Gross, Andreas Festag, and Alois Knoll. 2024. Collaborative Semantic Occupancy Prediction with Hybrid Feature Fusion in Connected Automated Vehicles. , 17996-18006 pages. https://rruisong.github.io/publications/CoHFF

[26] Sanbao Su, Songyang Han, Yiming Li, Zhili Zhang, Chen Feng, Caiwen Ding, and Fei Miao. 2024. Collaborative Multi-Object Tracking with Conformal Uncertainty Propagation. arXiv:2303.14346 (Jan. 2024). https://doi.org/10.48550/arXiv.2303.14346 arXiv:2303.14346 [cs].

[27] Chen Sun, Ruihe Zhang, Yukun Lu, Yaodong Cui, Zejian Deng, Dongpu Cao, and Amir Khajepour. 2023. Toward ensuring safety for autonomous driving perception: Standardization progress, research advances, and perspectives. *IEEE Transactions on Intelligent Transportation Systems* 25, 5 (2023), 3286–3304.

[28] Ardi Tampuu, Tambet Matiisen, Maksym Semikin, Dmytro Fishman, and Naveed Muhammad. 2020. A survey of end-to-end driving: Architectures and training methods. *IEEE Transactions on Neural Networks and Learning Systems* 33, 4 (2020).

[29] Chenyi Wang, Raymond Muller, Ruoyu Song, Jean-Philippe Monteuuis, Jonathan Petit, Yanmao Man, Ryan Gerdes, Z Berkay Celik, and Ming Li. 2025. From Threat to Trust: Exploiting Attention Mechanisms for Attacks and Defenses in Cooperative Perception. In *USENIX Security 25*.

[30] Tianhang Wang, Fan Lu, Zehan Zheng, Guang Chen, and Changjun Jiang. 2024. RCDN: Towards Robust Camera-Insensitivity Collaborative Perception via Dynamic Feature-based 3D Neural Modeling. arXiv:2405.16868 (May 2024). https://doi.org/10.48550/arXiv.2405.16868 arXiv:2405.16868 [cs].

[31] Zhe Wang, Siqi Fan, Xiaoliang Huo, Tongda Xu, Yan Wang, Jingjing Liu, Yilun Chen, and Ya-Qin Zhang. 2023. VIMI: Vehicle-Infrastructure Multi-view Intermediate Fusion for Camera-based 3D Object Detection. arXiv:2303.10975 (March 2023). https://doi.org/10.48550/arXiv.2303.10975 arXiv:2303.10975 [cs].

[32] Zhaobin Wang and Zekun Yang. 2020. Review on image-stitching techniques. *Multimedia Systems* 26, 4 (2020), 413–430.

[33] Hai Wu, Chenglu Wen, Shaoshuai Shi, Xin Li, and Cheng Wang. 2023. Virtual sparse convolution for multimodal 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 21653–21662.

[34] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. 2022. CoBEVT: Cooperative Bird's Eye View Semantic Segmentation with Sparse Transformers. *Proceedings of Machine Learning Research* 205 (7 2022), 989–1000. https://arxiv.org/abs/2207.02202v2

[35] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. 2022. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 ICRA*. IEEE, 2583–2589.

[36] Melih Yazgan, Thomas Graf, Min Liu, Tobias Fleck, and J. Marius Zöllner. 2024. A Survey on Intermediate Fusion Methods for Collaborative Perception Categorized by Real World Challenges. In *2024 IEEE Intelligent Vehicles Symposium (IV)*. 2226–2233. https://doi.org/10.1109/IV55156.2024.10588382

[37] Haibao Yu, Wenxian Yang, Jiaru Zhong, Zhenwei Yang, Siqi Fan, Ping Luo, and Zaiqing Nie. 2025. End-to-End Autonomous Driving Through V2X Cooperation. *Proceedings of the AAAI Conference on Artificial Intelligence* 39 (2025). https://doi.org/10.1609/AAAI.V39I9.33040

[38] Qingzhao Zhang, Shuowei Jin, Ruiyang Zhu, Jiachen Sun, and Xumiao Zhang. [n. d.]. On Data Fabrication in Collaborative Vehicular Perception: Attacks and Countermeasures. ([n. d.]).

[39] Seth Z. Zhao, Hao Xiang, Chenfeng Xu, Xin Xia, Bolei Zhou, and Jiaqi Ma. 2024. CooPre: Cooperative Pretraining for V2X Cooperative Perception. arXiv:2408.11241 (Aug. 2024). https://doi.org/10.48550/arXiv.2408.11241 arXiv:2408.11241 [cs].

[40] Yangheng Zhao, Zhen Xiang, Sheng Yin, Xianghe Pang, Siheng Chen, and Yanfeng Wang. 2023. Malicious agent detection for robust multi-agent collaborative perception. *arXiv preprint arXiv:2310.11901* (2023).

[41] Jiaru Zhong, Haibao Yu, Tianyi Zhu, Jiahui Xu, Wenxian Yang, Zaiqing Nie, and Chao Sun. 2024. Leveraging Temporal Contexts to Enhance Vehicle-Infrastructure Cooperative Perception. arXiv:2408.10531 (Aug. 2024). https://doi.org/10.48550/arXiv.2408.10531 arXiv:2408.10531 [cs].