# CP-Guard: Malicious Agent Detection and Defense in Collaborative Bird's Eye View Perception

**Senkang Hu[1*], Yihang Tao[1*], Guowen Xu[2], Yiqin Deng[1†], Xianhao Chen[3], Yuguang Fang[1], and Sam Kwong[4]**

[1]Department of Computer Science, City University of Hong Kong
[2]School of Computer Science and Engineering, University of Electronic Science and Technology of China
[3]Department of Electrical and Electronic Engineering, The University of Hong Kong
[4]Department of Computing and Decision Sciences, Lingnan University
{senkang.forest, yihang.tommy}@my.cityu.edu.hk, guowen.xu@uestc.edu.cn
xchen@eee.hku.hk, samkwong@ln.edu.hk, {yiqideng, my.Fang}@cityu.edu.hk

## Abstract

Collaborative Perception (CP) has shown a promising technique for autonomous driving, where multiple connected and autonomous vehicles (CAVs) share their perception information to enhance the overall perception performance and expand the perception range. However, in CP, ego CAV needs to receive messages from its collaborators, which makes it easy to be attacked by malicious agents. For example, a malicious agent can send harmful information to the ego CAV to mislead it. To address this critical issue, we propose a novel method, **CP-Guard**, a tailored defense mechanism for CP that can be deployed by each agent to accurately detect and eliminate malicious agents in its collaboration network. Our key idea is to enable CP to reach a consensus rather than a conflict against the ego CAV's perception results. Based on this idea, we first develop a probability-agnostic sample consensus (PASAC) method to effectively sample a subset of the collaborators and verify the consensus without prior probabilities of malicious agents. Furthermore, we define a collaborative consistency loss (CCLoss) to capture the discrepancy between the ego CAV and its collaborators, which is used as a verification criterion for consensus. Finally, we conduct extensive experiments in collaborative bird's eye view (BEV) tasks and our results demonstrate the effectiveness of our CP-Guard.

## Introduction

Recently, multi-agent collaborative perception has attracted great attention from both academia and industries since it can overcome the limitation of single-agent perception such as occlusion and limitation of sensing range (Han et al. 2023; Hu et al. 2024b). Because the collaborative CAVs send complementary information (e.g., raw sensor data, intermediate features, and perception results) to ego CAV, the ego CAV can leverage this complementary information to extend its perception range and tackle the blind spot problem in its view, which is crucial for the safety of the autonomous driving systems. The operational flow of CP is as follows. Each CAV independently encodes local sensor inputs into intermediate feature maps. Then, these CAVs share their feature maps with their ego CAV by vehicle-to-vehicle (V2V) communication. Finally, the ego CAV fuses the received feature maps with its own feature maps and decodes them to acquire the final perception results.

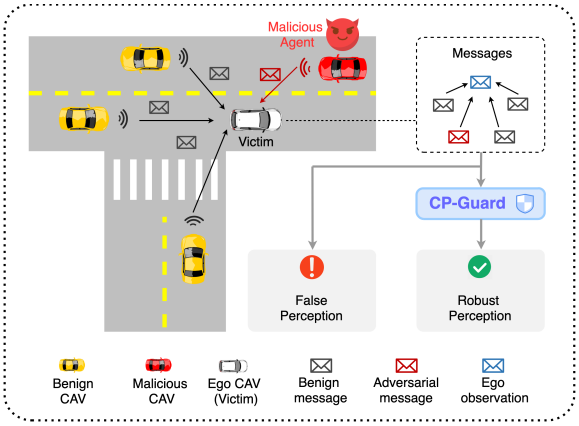


Figure 1: **Illustration of the threats of malicious agent in collaborative perception and our defense framework, CP-Guard**. When there is no defense, malicious CAVs could easily send intricately crafted adversarial messages to the ego CAV, consequently misleading the CP system and resulting in false perception outputs. To counter this vulnerability, we propose CP-Guard, a tailored defense mechanism for CP that can effectively detect and neutralize malicious agents, thereby ensuring robust perception outcomes.

However, compared with single-agent perception, multi-agent CP is more vulnerable to security threats and easy to attack, since it incorporates the information from multiple agents which makes the attack surface larger. An attack could be executed by a man-in-the-middle who alters the feature maps sent to the victim agent, or by a malicious agent that directly transmits manipulated feature maps to the victim agent. For example, Tu *et al.* (Tu et al. 2021) generated adversarial perturbations on the feature maps and attacked the ego CAV, resulting in the wrong perception results. Ad-

*These authors contributed equally.
†Corresponding author.

ditionally, as the encoded feature maps are not visually interpretable by humans, moderate modifications to these maps will go unnoticed, rendering the attack quite stealthy.

This issue raises significant risks to CP if the ego CAV cannot accurately detect and eliminate the malicious agents in its collaboration network and the perception results are corrupted, which may result to catastrophic consequences. For example, the ego CAV may misclassify the traffic light status or fail to detect the front objects, leading to serious traffic accidents or even loss of life. Therefore, it is essential to develop a defense mechanism for CP that is robust to attack from malicious agents and can remove the malicious agents in its collaboration network.

In order to address this issue, several works have explored this problem. For example, Li *et al.* (Li et al. 2023) proposed robust collaborative sampling consensus (ROBOSAC) to randomly sample a subset of the collaborators and verify the consensus, but it requires the prior probabilities of malicious agents, which are usually unknown in practice. In addition, Zhao *et al.* (Zhao et al. 2023) and Zhang *et al.* (Zhang et al. 2023) also developed defense methods against malicious agents, while these methods need to check the collaborators one by one, which is inefficient and computation-consuming. Moreover, other works (Tu et al. 2021; Raghunathan et al. 2020; Zhang and Li 2020) use adversarial training to enhance the robustness of the model. However, adversarial training introduces additional overhead during training and lacks generalization for unseen attacks. Additionally, it may result in a reduction in accuracy and is non-trivial to achieve computationally efficient and generalizable adversarial defense in CP.

In order to fill in the research gap and overcome the aforementioned limitations, we design a novel defense mechanism, CP-Guard. It can be deployed by each agent to accurately detect and eliminate malicious agents in the local collaboration network. The key idea is to enable CP to achieve a consensus rather than a conflict against the ego CAV's perception results. Following this idea, we first design a *probability-agnostic sample consensus* (PASAC) method to effectively sample a subset of the collaborators and verify the consensus without prior probabilities of malicious agents. In addition, the consensus is verified by our carefully designed *collaborative consistency loss* (CCLoss), which is used to calculate the discrepancy between the ego CAV and its collaborators. If a collaborator's collaborative consistency loss exceeds a predefined threshold, the collaborator is considered a benign agent, otherwise, it is considered a malicious agent. The main contributions of this paper are summarized as follows.

- We analyze the vulnerabilities of CP against malicious agents and develop a novel framework for robust collaborative BEV perception, CP-Guard, which can defend against attacks and eliminate malicious agents from the local collaboration network.
- We establish a probability-agnostic sample consensus (PASAC) method to effectively sample a subset of the collaborators and verify the consensus without prior probabilities of malicious agents. In addition, we design a collaborative consistency loss (CCLoss) as a verification criterion for consensus, which can calculate the discrepancy between the ego CAV and its collaborators.
- We conduct extensive experiments on collaborative BEV tasks and the results demonstrate the effectiveness of our CP-Guard and its generalization to different attacks.

## Background and Related Work

### Collaborative Perception

Collaborative perception has been investigated as a means to mitigate the limitations inherent to the field-of-view (FoV) in single-agent perception systems, enhancing the accuracy, robustness, and resilience of these systems (Fang et al. 2024). In this collaborative context, agents may opt for one of three predominant data fusion strategies: (1) early-stage raw data fusion, (2) intermediate-stage feature fusion, and (3) late-stage output fusion. Early-stage fusion, while increasing the data communication load, typically yields more precise collaboration outcomes. In contrast, late-stage fusion consumes less bandwidth but introduces greater uncertainty to the results. Intermediate-stage fusion, favored in much of the current literature, strikes an optimal balance between communication overhead and perceptual accuracy. Research aimed at enhancing collaborative perception performance is multifaceted, addressing aspects such as communication overhead (Su et al. 2024), robustness (Lu et al. 2023), system heterogeneity (Lu et al. 2024), and domain generalization (Hu et al. 2024a). Among these, robustness has emerged as a particularly critical focus within the field of collaborative perception. Despite extensive studies on system intrinsic robustness, addressing challenges such as communication disruptions (Ren et al. 2024), pose noise correction (Lu et al. 2023), and communication latency (Lei et al. 2022), most existing research works have not accounted for the presence of malicious attackers within the collaborative framework. Only a selected few studies examine the implications of robustness in scenarios compromised by malicious nodes, highlighting a significant gap in current research methodologies.

### Adversarial Perception

Adversarial attacks targeting at single-vehicle perception systems predominantly employ techniques such as GPS spoofing (Li et al. 2021), LiDAR spoofing (Hallyburton et al. 2022), and the deployment of physically realizable adversarial objects (Tu et al. 2020). In the context of multi-vehicle collaborative perception, the nature of adversarial strategies can vary significantly depending on the stage of collaboration. For early-stage collaborative perception, Zhang *et al.* (Zhang et al. 2023) have developed sophisticated attacks involving object spoofing and removal. These attacks exploit vulnerabilities by simulating the presence or absence of objects and reconstructing LiDAR point clouds using advanced ray-casting techniques. In contrast, late-stage collaboration typically involves the sharing of object locations (Schiegg et al. 2020), which provides adversaries with opportunities to manipulate these shared locations easily. Intermediate-stage attacks are particularly nuanced, often requiring that an

attacker possesses white-box access to the perception models. This knowledge enables more precise manipulations of the system, though such systems are generally resistant to simplistic black-box strategies, such as ray-casting attacks, due to the protective effect of benign feature maps which significantly reduce the efficacy of such attacks. Tu *et al.* (Tu et al. 2021) were among the pioneers in articulating an untargeted adversarial attack aimed at maximizing the generation of inaccurate detection bounding boxes by manipulating feature maps in intermediate-fusion systems. Building on this foundational work, Zhang *et al.*(Zhang et al. 2023) have advanced the methodology by integrating perturbation initialization and feature map masking techniques to facilitate realistic, real-time targeted attacks. Our work is dedicated to exploring and mitigating adversarial threats specifically under intermediate-level collaborative perception framework, aiming to enhance system resilience against sophisticated attacks.

### Defensive Perception

To fortify intermediate-level collaborative perception systems against adversarial attacks, Li *et al.* (Li et al. 2023) proposed the Robust Collaborative Sampling Consensus (ROBOSAC) method. This approach entails a random selection of a subset of collaborators for consensus verification. Despite its potential, the efficacy of ROBOSAC hinges on the availability of prior probabilities of malicious intent among agents, which are often unknown in real-world scenarios. Moreover, Zhao *et al.* (Zhao et al. 2023) and Zhang *et al.* (Zhang et al. 2023) have formulated defensive strategies targeting at identifying malicious agents. These techniques, however, involve scrutinizing each collaborator individually, rendering them both computationally intensive and inefficient. Adversarial training has also been explored as a mechanism to bolster system robustness, as demonstrated in the studies by Tu *et al.* (Tu et al. 2021), Raghunathan *et al.* (Raghunathan et al. 2020), and Zhang *et al.* (Zhang and Li 2020). While this approach enhances system security, it substantially increases the computational load during training and may not effectively generalize to novel, unseen attacks. Additionally, adversarial training often results in diminished model accuracy and poses significant challenges in developing a computationally efficient and scalable adversarial defense that can be broadly applied across collaborative perception platforms. In contrast, our research introduces a methodology that can be autonomously implemented by each agent to accurately detect and neutralize malicious entities within the local collaborative network, aiming to enhance both the efficiency and effectiveness of defense mechanisms in collaborative perception systems.

## Problem Setup

### Collaborative BEV Segmentation

BEV segmentation is essential for autonomous driving since it enables multi-modal sensor data (e.g. LiDAR point cloud, multi-view camera images) to be transformed into a unified BEV space for information aggregation or fusion. This approach offers significant advantages in accurately maintaining the spatial and temporal locations and scales of road elements. In this paper, we focus on the LiDAR-based collaborative BEV segmentation task with an intermediate feature fusion paradigm.

Specifically, consider a set of $N+1$ CAVs, including the ego CAV, each CAV is installed with a feature encoder $f_{\mathrm{E}}$ and a feature aggregator $f_{\mathrm{A}}$ as well as a BEV segmentation decoder $f_{\mathrm{D}}$. For the $i$-th CAV, the input is a set of voxelized LiDAR point cloud $\mathbf{X}_i \in \mathbb{R}^{W\times H\times n}$, where $W$ and $H$ are the width and height of the voxelized point cloud, respectively, and $n$ is the dimension of the voxelized point cloud. The collaborative BEV segmentation pipeline can be described as follows.

1. Firstly, the feature encoder $f_{\mathrm{E}}$ is used to extract the intermediate feature maps $\mathbf{F}_i = f_{\mathrm{E}}(\mathbf{X}_i) \in \mathbb{R}^{\frac{W}{K}\times\frac{H}{K}\times C}$, where $K$ indicates the down-sample rate and $C$ is the number of channels.
2. Then, the other CAVs transmit their intermediate feature maps $\mathbf{F}_i$ to the ego CAV. The ego CAV leverages the aggregator $f_{\mathrm{A}}$ to fuse the feature maps from all CAVs including its own feature map, which can be formulated as $\mathbf{F} = f_{\mathrm{A}}(\mathbf{F}_0, \mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_N)$, where $\mathbf{F}_0$ is the ego CAV's feature map.
3. Finally, the ego CAV decodes the aggregated feature map $\mathbf{F}$ into the final BEV segmentation map $\mathbf{Y} = f_{\mathrm{D}}(\mathbf{F})$.

During training, given the ground truth BEV segmentation map $\mathbf{Y}^*$, the loss function is defined as $\mathcal{L}_{\text{seg}}(\mathbf{Y}, \mathbf{Y}^*)$. The goal is to minimize the loss function $\mathcal{L}_{\text{seg}}$ by optimizing the parameters of the feature encoder $f_{\mathrm{E}}$, feature aggregator $f_{\mathrm{A}}$, and BEV segmentation decoder $f_{\mathrm{D}}$.

### Adversarial Threat Model in CP

In order to defend against malicious agents in CP, we first need to figure out the attack scenarios and the attacker's abilities. Specifically, we consider an attacker to have full access to malicious CAVs. In addition, since the BEV segmentation model is deployed on each CAV, the attacker has full access to the model architecture, parameters, and intermediate feature maps, enabling the attacker to launch a white-box attack. Based on this, the attacker aims to manipulate the intermediate feature maps by adding adversarial perturbations to maximize the ego CAV's BEV segmentation loss. Then, these adversarial messages are transmitted to the ego CAV to fool its perception fusion. The attacker's goal can be formulated as follows.

$$\begin{aligned} &\max_{\|\delta\|\leq\Delta} \mathcal{L}_{\text{seg}}(\mathbf{Y}^{\delta}, \mathbf{Y}^*), \\ &\text{s.t.} \quad \mathbf{Y}^{\delta} = f_{\mathrm{D}}\left(f_{\mathrm{A}}(\mathbf{F}_0, \mathbf{F}_1, \mathbf{F}_m + \delta, \cdots, \mathbf{F}_N)\right), \end{aligned} \tag{1}$$

where $\mathbf{Y}^{\delta}$ is the adversarial collaborative BEV segmentation map, $\mathbf{F}_m$ is the malicious agent's feature map, and $\delta$ is the optimization perturbation which is constrained by $\|\delta\| \leq \Delta$ to ensure its stealth to avoid being detected. Its size is the same as the size of intermediate feature map $\mathbf{F}_m$, which is $\mathbb{R}^{\frac{W}{K}\times\frac{H}{K}\times C}$.

Algorithm 1: PASAC

**Input**:

- $\{\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_N\}$, intermediate feature maps from collaborators. $\mathbf{F}_0$, the intermediate feature of ego CAV.
- $f_{\text{A}}$, $f_{\text{D}}$, the aggregator and BEV segmentation decoder.
- $N_{\text{upper}}$, maximum number of selected collaborators for ego CAV, and $N_{\text{upper}} \leq N$.
- $\varepsilon$, the threshold of $\mathcal{L}_{\text{CCLoss}}$.

**Output**: $\{\mathbf{B}_i\}$, the set of benign collaborators

$\mathbf{Y}_0 = f_{\text{D}}(\mathbf{F}_0)$.
**procedure** PASAC($\{\mathbf{F}_i\}_{i=1,\ldots,N}$)
  **if** len($\{\mathbf{B}_i\}$) $\geq N_{\text{upper}}$ **then**
    **return** $\{\mathbf{B}_i\}$
  **end if**
  **if** len($\{\mathbf{F}_i\}$) $= 1$ **then**
    $\mathbf{Y}_k = f_{\text{D}}(f_{\text{A}}(\mathbf{F}_0, \mathbf{F}_k))$.
    **if** $\mathcal{L}_{\text{CCLoss}}(\mathbf{Y}_0, \mathbf{Y}_k) \leq \varepsilon$ **then**
      $\{\mathbf{B}_i\} \leftarrow \{\mathbf{B}_i\} \cup \mathbf{F}_k$
    **end if**
    **return** $\{\mathbf{B}_i\}$
  **end if**
  $\{\mathbf{F}_i\}_{i=1,\ldots,N} \rightarrow \{\mathbf{F}_i\}_{i=1,\ldots,\frac{N}{2}}, \{\mathbf{F}_i\}_{i=\frac{N}{2},\ldots,N}$
  $\mathbf{Y}_{\text{G1}} = f_{\text{D}}(f_{\text{A}}(\mathbf{F}_0, \{\mathbf{F}_i\}_{i=1,\ldots,\frac{N}{2}}))$
  $\mathbf{Y}_{\text{G2}} = f_{\text{D}}(f_{\text{A}}(\mathbf{F}_0, \{\mathbf{F}_i\}_{i=\frac{N}{2},\ldots,N}))$
  **if** $\mathcal{L}_{\text{CCLoss}}(\mathbf{Y}_0, \mathbf{Y}_{\text{G1}}) \leq \varepsilon$ **then**
    $\{\mathbf{B}_i\}_{\text{sublist}}$ = PASAC($\{\mathbf{F}_i\}_{i=1,\ldots,\frac{N}{2}}$)
    $\{\mathbf{B}_i\} \leftarrow \{\mathbf{B}_i\} \cup \{\mathbf{B}_i\}_{\text{sublist}}$
  **else**
    $\{\mathbf{B}_i\} \leftarrow \{\mathbf{B}_i\} \cup \{\mathbf{F}_i\}_{i=1,\ldots,\frac{N}{2}}$
  **end if**
  **if** $\mathcal{L}_{\text{CCLoss}}(\mathbf{Y}_0, \mathbf{Y}_{\text{G2}}) \leq \varepsilon$ **then**
    $\{\mathbf{B}_i\}_{\text{sublist}}$ = PASAC($\{\mathbf{F}_i\}_{i=\frac{N}{2},\ldots,N}$)
    $\{\mathbf{B}_i\} \leftarrow \{\mathbf{B}_i\} \cup \{\mathbf{B}_i\}_{\text{sublist}}$
  **else**
    $\{\mathbf{B}_i\} \leftarrow \{\mathbf{B}_i\} \cup \{\mathbf{F}_i\}_{i=\frac{N}{2},\ldots,N}$
  **end if**
  **return** $\{\mathbf{B}_i\}$
**end procedure**

## Method

In this section, we present our CP-Guard in detail. It consists of two main components: (1) *Probability-Agnostic Sample Consensus* (PASAC) and (2) *Collaborative Consistency Loss Verification* (CCLoss). PASAC is designed to effectively sample a subset of collaborators for consensus verification without relying on prior probabilities of malicious intent. CCLoss is proposed to verify the consensus between the ego CAV and the collaborative CAVs. These two components work collaboratively to detect and neutralize malicious agents in the collaborative perception network. We elaborate on these two components in the following subsections.

### Probability-Agnostic Sample Consensus

To achieve the consensus of collaborators, the most straightforward method is to check the collaborators one by one. However, this method is time-consuming and computation-intensive, especially when the number of collaborators is large. A better method is to randomly sample a subset of collaborators for consensus verification at each time, such as ROBOSAC (Li et al. 2023). However, ROBOSAC requires the prior probabilities of malicious intent among agents, which are often unknown in real-world scenarios. To fill in this research gap, we propose PASAC.

More specifically, given a set of $N$ collaborators, the ego CAV will generate the collaborative BEV segmentation map $\mathbf{Y}$ based on its observation and the received messages for feature fusion from the $N$ collaborators. Firstly, the ego CAV generates its BEV segmentation map $\mathbf{Y}_0$ based on its own observation. Then, it randomly split the collaborators into two groups of equal size. After receiving all the messages, the ego CAV fuses the features and generates the BEV segmentation map $\mathbf{Y}_{\text{G1}}$ based on the messages $\{\mathbf{F}_i\}_{i=1,\ldots,\frac{N}{2}}$ from the first group. Similarly, it generates the BEV segmentation map $\mathbf{Y}_{\text{G2}}$ based on the messages $\{\mathbf{F}_i\}_{i=\frac{N}{2},\ldots,N}$ from the second group.

Then, the ego CAV verifies the consensus and checks if there is any malicious CAV in the two groups. The consensus is verified by CCLoss to be introduced in the next subsection. Specifically, the CCLoss is calculated between the ego CAV and each group, that is, $\mathcal{L}_{\text{CCLoss}}(\mathbf{Y}_0, \mathbf{Y}_{\text{G1}})$ and $\mathcal{L}_{\text{CCLoss}}(\mathbf{Y}_0, \mathbf{Y}_{\text{G2}})$. If the CCLoss exceeds a predefined threshold, the group is considered benign, otherwise, it is considered to contain malicious CAVs.

Suppose the first group is benign and the second group is verified to have malicious CAVs, all CAVs in the first group are marked as benign and incorporated in the following collaboration. For the second group, the ego CAV continues to split the second group into two subgroups and repeats the consensus verification process. This process will continue until finding all the malicious CAVs or obtain enough benign CAVs. The detailed procedures of PASAC can be summarized as follows.

1. Generate the BEV segmentation map $\mathbf{Y}_0$ based on the ego CAV's observation.
2. Split the collaborators into two groups.
3. Generate the BEV segmentation maps $\mathbf{Y}_{\text{G1}}$ and $\mathbf{Y}_{\text{G2}}$ based on the messages from the two groups, respectively.
4. Calculate the CCLoss between the ego CAV and two groups, respectively, and verify if there are malicious CAVs in the two groups.
5. If the group has no malicious CAVs, mark all CAVs in the group as benign. Otherwise, repeat 2-5 until all malicious CAVs are found or enough benign CAVs are obtained.

The pseudo-code of PASAC is shown in Algorithm 1.

### Collaborative Consistency Loss Verification

To verify the consensus between the ego CAV and the collaborators, we design a novel loss function, *Collaborative Consistency Loss* (CCLoss). CCLoss is used to calculate the discrepancy between the ego CAV and the collaborators.

Given the intermediate feature maps $\mathbf{F}_0$ of the ego CAV and a set of intermediate feature maps $\{\mathbf{F}_1, \ldots, \mathbf{F}_i\}$ from

| **Method** | Vehicle | Sidewalk | Terrain | Road | Buildings | Pedestrian | Vegetation | **mIoU** |
|---|---|---|---|---|---|---|---|---|
| Upper-bound | 55.58 | 48.20 | 47.33 | 69.60 | 29.34 | 21.67 | 41.02 | 40.45 |
| CP-Guard (against FGSM attack) | 52.76 | 46.35 | 46.67 | 68.32 | 28.98 | 20.51 | 40.15 | 39.30 |
| CP-Guard (against C&W attack) | 49.22 | 44.08 | 44.76 | 65.58 | 30.12 | 20.83 | 39.10 | 37.95 |
| CP-Guard (against PGD attack) | 52.84 | 46.41 | 46.73 | 68.41 | 29.01 | 20.48 | 40.16 | 39.34 |
| Lower-bound | 47.06 | 42.46 | 43.78 | 64.07 | 30.51 | 21.21 | 37.32 | 37.09 |
| No Defense (FGSM attack) | 26.80 | 27.21 | 29.05 | 36.41 | 16.44 | 12.05 | 22.99 | 21.57 |
| No Defense (C&W attack) | 34.53 | 35.66 | 35.54 | 56.59 | 24.27 | 13.37 | 34.10 | 29.80 |
| No Defense (PGD attack) | 22.50 | 19.63 | 15.42 | 15.33 | 9.18 | 8.29 | 22.72 | 14.34 |

Table 1: **Quantitative results of CP-Guard** with CCLoss threshold $\varepsilon = 0.08$, PGD iterations = 15, C&W iteration = 15, and FGSM noise variance = 10.

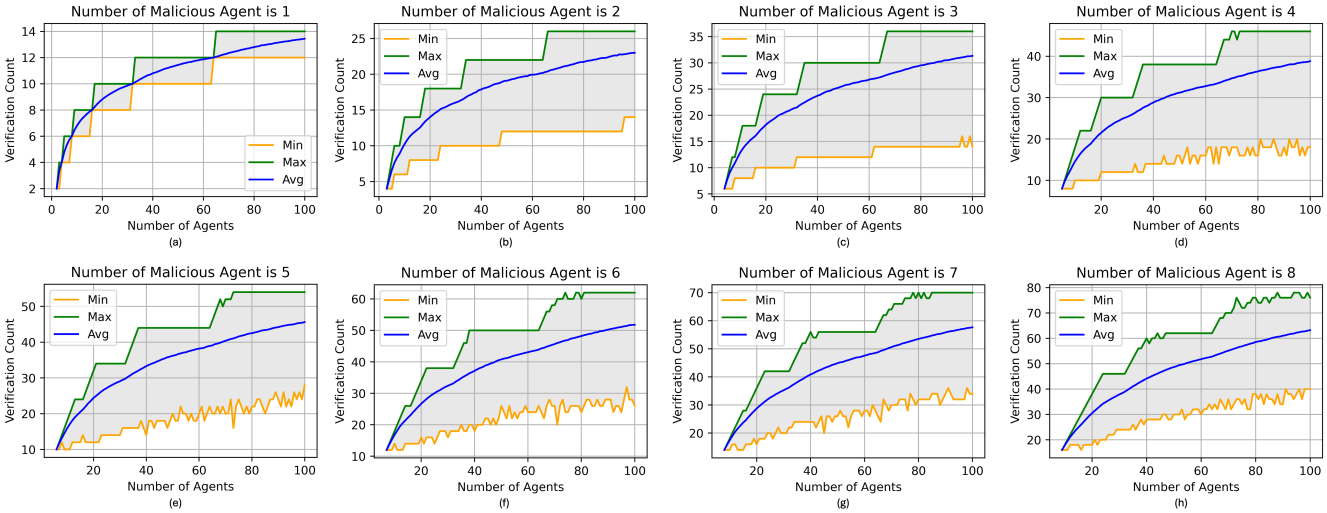


Figure 2: **Quantitative results of PASAC:** Number of Agents vs Verification Count.

the collaborators. The ego CAV will generate two BEV segmentation maps:

$$\mathbf{Y}_0 = f_{\text{D}}(\mathbf{F}_0), \tag{2}$$

$$\mathbf{Y}_{\text{fuse}} = f_{\text{D}}(f_{\text{A}}(\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_i)), \tag{3}$$

where $\mathbf{Y}_0$ and $\mathbf{Y}_{\text{fuse}}$ are 3D matrices and their sizes are in $\mathbb{R}^{W_D \times H_D \times C}$ with $W_D$, $H_D$, and $C$ being the width, height, and the number of classes of the BEV segmentation map, respectively.

As stated before, our key idea is to enable CP to achieve consensus rather than conflict against the ego CAV's perception result. Following this idea, we carefully design the CCLoss to measure the discrepancy between the ego CAV and the collaborators, which is formulated as:

$$\mathcal{L}_{\text{CCLoss}}(\mathbf{Y}_0, \mathbf{Y}_{\text{fuse}}) = \frac{\sum_{j=1}^{C} w_j \sum_{i=1}^{W_D \cdot H_D} p_{i,j}^0 p_{i,j}^{\text{fuse}}}{\sum_{j=1}^{C} w_j \left( \sum_{i=1}^{W_D \cdot H_D} p_{i,j}^0 + \sum_{i=1}^{W_D \cdot H_D} p_{i,j}^{\text{fuse}} \right)} \tag{4}$$

where $C$ is the number of classes, $p_{i,j}^0$ and $p_{i,j}^{\text{fuse}}$ are the probabilities of the $j$-th class at the $i$-th pixel in the BEV segmentation map $\mathbf{Y}_0$ and $\mathbf{Y}_{\text{fuse}}$, respectively, and $w_j$ is the weight of the $j$-th class, defined as the inverse frequency of the class $w_j = 1/(\sum_{j=1}^{C}(p_{i,j}^0 + p_{i,j}^{\text{fuse}}))^2$. For the numerator of $\mathcal{L}_{\text{CCLoss}}$, it calculates the weighted sum of the product of the probabilities for each pixel and each class, which essentially measures the overlap between the two distributions. The weight $w_j$ ensures that the contribution of each class is adjusted according to its importance or frequency. The denominator sums up the weighted sums of the probabilities from both the ego CAV's prediction map and the fused segmentation maps for each class. It represents the total probability mass for each class, adjusted by the weights. Finally, the fraction measures the similarity between the two distributions. If these two distributions are similar, the value of $\mathcal{L}_{\text{CCLoss}}$ will be close to 1. On the contrary, if these two distributions are different, the value $\mathcal{L}_{\text{CCLoss}}$ will be close to 0.

In addition, we need to set a threshold $\varepsilon$ to determine whether the set contains a malicious agent. Thus, we have the following verification rule if there is a malicious agent in the set:

$$\mathcal{L}_{\text{CCLoss}}(\mathbf{Y}_0, \mathbf{Y}_{\text{fuse}}) \leq \varepsilon. \tag{5}$$

The choice of the threshold $\varepsilon$ is crucial. A large $\varepsilon$ will lead to a high false-positive rate, while a small $\varepsilon$ will lead to a high false-negative rate.

| **Threshold** $\varepsilon$ | Vehicle | Sidewalk | Terrain | Road | Buildings | Pedestrian | Vegetation | **mIoU** |
|---|---|---|---|---|---|---|---|---|
| 0.02 | 34.94 | 34.98 | 28.55 | 36.20 | 22.62 | 14.23 | 32.62 | 25.97 |
| 0.05 | 50.70 | 45.25 | 44.76 | 64.50 | 28.34 | 19.84 | 39.45 | 37.79 |
| 0.08 | **52.84** | **46.41** | **46.73** | **68.41** | **29.01** | **20.48** | **40.16** | **39.34** |
| 0.10 | 52.48 | 46.24 | 46.36 | 67.97 | 28.95 | 20.70 | 39.97 | 39.17 |
| 0.12 | 49.61 | 43.80 | 44.74 | 65.46 | 30.21 | 21.26 | 38.56 | 37.99 |
| 0.15 | 48.62 | 42.89 | 44.19 | 64.91 | 30.25 | 21.19 | 37.75 | 37.52 |

Table 2: **Quantitative results of ablation studies** on $\mathcal{L}_{\text{CCLoss}}$ threshold $\varepsilon$ against PGD attack (iterations = 15).

| Attack Ratio | ROBOSAC | | | PASAC (Ours) | | |
|---|---|---|---|---|---|---|
| | Verification Count | | | Verification Count | | |
| | Min | Max | Avg | Min | Max | Avg |
| 0.8 | 1 | 17 | 4.73 | 8 | **8** | 8.00 |
| 0.6 | 1 | 46 | 8.29 | 6 | **8** | **7.59** |
| 0.4 | 1 | 39 | 10.36 | 4 | **8** | **6.60** |
| 0.2 | 1 | 19 | 4.89 | 4 | **6** | **4.79** |
| Average | 1.00 | 30.25 | 7.06 | 5.50 | **7.50** | **6.74** |

Table 3: **Comparison results** between ROBOSAC and PASAC

# Experiments

## Experimental Setup

**Datasets and Evaluation Metrics.** In our experiments, we leverage V2X-Sim (Li et al. 2022) as our dataset. It is the first synthetic dataset for CP generated by CARLA-SUMO co-simulator. In addition, to evaluate the performance of the segmentation, we adopt mean Intersection over Union (mIoU) as the evaluation metric. We also use Verification Count to evaluate the performance of PASAC, which is the total number of times that malicious agents are checked.

**Implementation Details.** We build the collaborative BEV segmentation model by PyTorch, using U-Net (Ronneberger, Fischer, and Brox 2015) as the backbone, V2VNet (Wang et al. 2020) as the fusion method. Our experiment is deployed on a computer consisting of 2 Intel(R) Xeon(R) Silver 4410Y CPUs (2.0GHz), four NVIDIA RTX A5000 GPUs, and 512GB DDR4 RAM. As for the implementation of adversarial attacks, we employ three kinds of attacks: fast gradient sign method (FGSM) (Goodfellow, Shlens, and Szegedy 2015), Carlini & Wagner (C&W) (Carlini and Wagner 2017), and the projected gradient descent (PGD) (Madry et al. 2018). For each attack, we set the maximum perturbation $\delta_{\text{max}} = 0.1$, iterations steps $T = 15$, and the step size $\alpha = 0.01$.

## Quantitative Evaluation

**Evaluation of CP-Guard.** We evaluate the efficacy of our CP-Guard scheme against a variety of adversarial attacks. The outcomes of these evaluations are detailed in Table **??**. In scenarios where the collaborative perception system lacks defensive mechanism, the mIoU across all three attack modalities significantly falls below the established lower bound, registering at 37.09. This substantial degradation in performance underscores the effectiveness of the adversarial attacks implemented. Conversely, our CP-Guard framework demonstrates robust defensive capabilities, effectively countering all evaluated attacks and achieving an mIoU that closely approaches the upper bound of 40.45. Specifically, for the FGSM and PGD attacks, setting the CCLoss threshold to $\varepsilon = 0.08$ proves optimal experimentally. This configuration allows CP-Guard to maximally leverage its defensive mechanisms, yielding mIoU scores of 39.30 and 39.34, respectively. The methodology for determining the optimal CCLoss threshold is further explored in the Ablation Studies section of this paper. In addition, it is noteworthy that while CP-Guard maintains performance above the lower bound when defending against the C&W attack, the mIoU observed is relatively lower at 37.95. This reduced efficacy can be attributed to the sophisticated fine-grained optimization process inherent to the C&W attack, which complicates the detection and mitigation efforts.

**Evaluation of PASAC.** To investigate the performance of PASAC, we conduct extensive experiments to study the relationship between the verification count and the number of benign agents and malicious agents. As shown in Fig. 2, the x-axis represents the number of benign agents and the y-axis represents the verification count. There are three lines in each subfigure, which represent the minimum, average, and maximum verification count, respectively. We can observe that the verification count increases with the number of collaborative agents and the growth trend is fast at the beginning and then becomes slow. In addition, the verification count is far less than the total number of agents, which indicates that PASAC is efficient in sampling collaborators. For example, when the number of collaborative agents is 100, the number of malicious agents is $m$, the average verification count is usually around $100 \times m \times 0.1$, and the mean verification count is below $100 \times m \times 0.05$.

**Comparison Results.** We conduct a comparative experiment with the previous state-of-the-art method, ROBOSAC (Li et al. 2023). Following the experiment setup from ROBOSAC, we evaluate the performance of ROBOSAC and PASAC under different attack ratios. The results are shown in Table 3. We observe that PASAC outperforms ROBOSAC in terms of the verification count. Specifically, PASAC achieves a lower verification count than ROBOSAC under different attack ratios. For example, when the attack ratio is 0.6, the average verification count of PASAC is 7.59, which is lower than the average verification count of ROBOSAC (8.29), and when the attack ratio is 0.4, the average verifica-

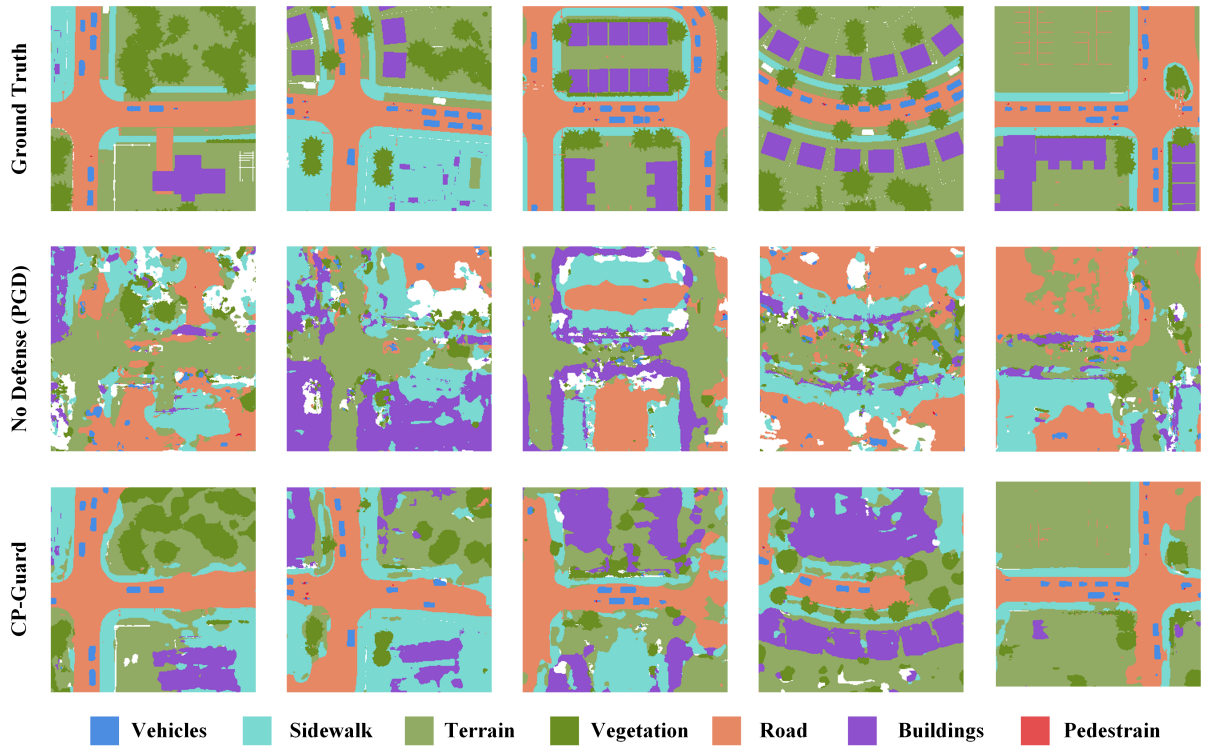


Figure 3: **Visualization of no defense and defensive CP-Guard results** on V2X-Sim datasets.

tion count of PASAC is 6.60, which is much lower than the average verification count of ROBOSAC (10.36). In addition, the results of PASAC are more stable than ROBOSAC, for example, the maximum verification count of PASAC is 8, while the maximum verification count of ROBOSAC is 46, which is much higher than the average verification count. These results indicate that PASAC can effectively sample collaborators and outperforms the state-of-the-art method, ROBOSAC. Furthermore, ROBOSAC needs to know the prior probabilities of malicious agents, while PASAC is a probability-agnostic sample method, so it does not require this information, which makes PASAC more practical in real-world scenarios.

### Qualitative Evaluation

As depicted in Fig. 3, we present the visualization results on the V2X-Sim dataset. Without CP-Guard, attackers can significantly disrupt collaborative perception, leading to a marked degradation in the performance of BEV segmentation tasks. However, our introduced CP-Guard framework can intelligently identify benign collaborators and eliminate malicious collaborators, thereby facilitating robust CP.

### Ablation Studies

We have further undertaken ablation studies to ascertain the optimal CCLoss threshold for our CP-Guard framework in defending against PGD attacks. The results of these studies are summarized in Table **??**. On the one hand, when the CCLoss threshold $\varepsilon$ is set below 0.08, the ability of the ego CAV to distinguish malicious agents is progressively impaired, resulting in a decline in the mIoU. Notably, at a CCLoss threshold of $\varepsilon = 0.02$, the ego-agent fails to identify the malicious agent, culminating in an mIoU of 25.97, which is substantially below the established lower-bound. On the other hand, as the CCLoss threshold $\varepsilon$ is increased beyond 0.08, there is a noticeable deterioration in the mIoU, approaching the lower-bound. This suggests that the ego CAV begins to mistrust benign collaborators, increasingly relying on its own inputs. Based on these ablation results, we determine that the optimal CCLoss threshold for CP-Guard against PGD attacks is 0.08. This setting optimally balances the trade-off between excluding malicious inputs and maintaining trust in benign collaborative data, thereby enhancing the overall robustness of the system.

## Conclusion

In this paper, we have designed a novel defense framework for CP named CP-Guard. It consists of two parts, the first is PASAC which can effectively sample the collaborators without the prior probabilities of malicious agents. The second is collaborative consistency loss verification which calculates the discrepancy between the ego CAV and the collaborators, which is used as a verification criterion for consensus. The extensive experiments show that our CP-Guard can defend against different types of attacks and can adaptively adjust the trade-off between the performance and computational overhead.

## Acknowledgements

This work was supported in part by the Hong Kong Innovation and Technology Commission under InnoHK Project CIMDA, in part by the Hong Kong SAR Government under the Global STEM Professorship and Research Talent Hub, and in part by the Hong Kong Jockey Club under the Hong Kong JC STEM Lab of Smart City (Ref.: 2023-0108). The work of Yiqin Deng was supported in part by the National Natural Science Foundation of China under Grant No. 62301300. The work of Xianhao Chen was supported in part by the Research Grants Council of Hong Kong under Grant 27213824.

## References

Carlini, N.; and Wagner, D. 2017. Towards Evaluating the Robustness of Neural Networks. arXiv:1608.04644.

Fang, Z.; Hu, S.; An, H.; Zhang, Y.; Wang, J.; Cao, H.; Chen, X.; and Fang, Y. 2024. PACP: Priority-Aware Collaborative Perception for Connected and Autonomous Vehicles. arXiv:2404.06891.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572.

Hallyburton, R. S.; Liu, Y.; Cao, Y.; Mao, Z. M.; and Pajic, M. 2022. Security Analysis of Camera-LiDAR Fusion Against Black-Box Attacks on Autonomous Vehicles. In *31st USENIX Security Symposium (USENIX Security 22)*, 1903–1920. Boston, MA: USENIX Association. ISBN 978-1-939133-31-1.

Han, Y.; Zhang, H.; Li, H.; Jin, Y.; Lang, C.; and Li, Y. 2023. Collaborative Perception in Autonomous Driving: Methods, Datasets and Challenges. *IEEE Intelligent Transportation Systems Magazine*, 15(6): 131–151. ArXiv:2301.06262 [cs].

Hu, S.; Fang, Z.; Chen, X.; Fang, Y.; and Kwong, S. 2024a. Towards Full-scene Domain Generalization in Multi-agent Collaborative Bird's Eye View Segmentation for Connected and Autonomous Driving. arXiv:2311.16754.

Hu, S.; Fang, Z.; Deng, Y.; Chen, X.; and Fang, Y. 2024b. Collaborative Perception for Connected and Autonomous Driving: Challenges, Possible Solutions and Opportunities. ArXiv:2401.01544 [cs, eess].

Lei, Z.; Ren, S.; Hu, Y.; Zhang, W.; and Chen, S. 2022. Latency-Aware Collaborative Perception. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, 316–332. Berlin, Heidelberg: Springer-Verlag.

Li, Y.; Fang, Q.; Bai, J.; Chen, S.; Juefei-Xu, F.; and Feng, C. 2023. Among Us: Adversarially Robust Collaborative Perception by Consensus. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 186–195. Paris, France: IEEE. ISBN 9798350307184.

Li, Y.; Ma, D.; An, Z.; Wang, Z.; Zhong, Y.; Chen, S.; and Feng, C. 2022. V2X-Sim: Multi-Agent Collaborative Perception Dataset and Benchmark for Autonomous Driving. *IEEE Robotics and Automation Letters*, 7(4): 10914–10921.

Li, Y.; Wen, C.; Juefei-Xu, F.; and Feng, C. 2021. Fooling LiDAR Perception via Adversarial Trajectory Perturbation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Lu, Y.; Hu, Y.; Zhong, Y.; Wang, D.; Wang, Y.; and Chen, S. 2024. An Extensible Framework for Open Heterogeneous Collaborative Perception. In *The Twelfth International Conference on Learning Representations*.

Lu, Y.; Li, Q.; Liu, B.; Dianati, M.; Feng, C.; Chen, S.; and Wang, Y. 2023. Robust Collaborative 3D Object Detection in Presence of Pose Errors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 4812–4818.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Raghunathan, A.; Xie, S. M.; Yang, F.; Duchi, J. C.; and Liang, P. 2020. Understanding and mitigating the tradeoff between robustness and accuracy. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, 7909–7919. JMLR.org.

Ren, S.; Lei, Z.; Wang, Z.; Dianati, M.; Wang, Y.; Chen, S.; and Zhang, W. 2024. Interruption-Aware Cooperative Perception for V2X Communication-Aided Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 9(4): 4698–4714.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. ArXiv:1505.04597 [cs].

Schiegg, F. A.; Bischoff, D.; Krost, J. R.; and Llatser, I. 2020. Analytical Performance Evaluation of the Collective Perception Service in IEEE 802.11p Networks. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 1–6.

Su, W.; Chen, L.; Bai, Y.; Lin, X.; Li, G.; Qu, Z.; and Zhou, P. 2024. What Makes Good Collaborative Views? Contrastive Mutual Information Maximization for Multi-Agent Perception. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16): 17550–17558.

Tu, J.; Ren, M.; Manivasagam, S.; Liang, M.; Yang, B.; Du, R.; Cheng, F.; and Urtasun, R. 2020. Physically Realizable Adversarial Examples for LiDAR Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tu, J.; Wang, T.; Wang, J.; Manivasagam, S.; Ren, M.; and Urtasun, R. 2021. Adversarial Attacks On Multi-Agent Communication. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 7748–7757. ISSN: 2380-7504.

Wang, T.-H.; Manivasagam, S.; Liang, M.; Yang, B.; Zeng, W.; and Urtasun, R. 2020. V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction. In *Computer Vision - ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, 605–621. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-030-58535-8.

Zhang, J.; and Li, C. 2020. Adversarial Examples: Opportunities and Challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7): 2578–2593. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Zhang, Q.; Jin, S.; Zhu, R.; Sun, J.; Zhang, X.; Chen, Q. A.; and Mao, Z. M. 2023. On Data Fabrication in Collaborative Vehicular Perception: Attacks and Countermeasures. ArXiv:2309.12955 [cs].

Zhao, Y.; Xiang, Z.; Yin, S.; Pang, X.; Chen, S.; and Wang, Y. 2023. Malicious Agent Detection for Robust Multi-Agent Collaborative Perception. ArXiv:2310.11901 [cs].