FROM REJECTION TO REPAIR: RECOVERING COLLABORATIVE PERCEPTION UNDER ADVERSARIAL ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Collaborative perception (CP) improves autonomous driving by enabling connected and autonomous vehicles (CAVs) to share complementary observations, but compromised CAVs can manipulate shared features and corrupt the ego vehicle's perception. Existing defenses largely follow a *detect-and-reject* paradigm, discarding suspicious CAVs or regions once corruption is detected. Such rejection indiscriminately removes benign, non-redundant information along with the adversarial content. We propose `Gecko`, a repair-oriented defense that selectively restores corrupted collaborative features. Our key insight is that feature-space references can guide where and how corruption should be repaired, while the resulting predictions verify the downstream effects of the repair decisions. `Gecko` first constructs learned and collaborative references to localize corruption and selectively restore affected regions according to cross-vehicle support. It then verifies the repaired features through their effects on the detector predictions and maps abnormal responses back to the corresponding repair decisions for correction. Extensive experiments demonstrate that `Gecko` consistently outperforms rejection-based defenses across diverse attacks and CP models. Compared with directly rejecting malicious CAVs, `Gecko` achieves relative improvements of 7.63% in AP@0.5 and up to 14.05% in AP@0.7 under MOR and TOR attacks.

1 INTRODUCTION

Collaborative perception (CP) enables connected and autonomous vehicles (CAVs) to overcome the limitations of individual sensing by sharing complementary observations across viewpoints (Han et al., 2023). In particular, intermediate-feature collaboration allows vehicles to exchange learned feature representations and jointly perceive objects that are distant, occluded, or outside the ego vehicle's field of view (Xu et al., 2022b;a; Hu et al., 2022; Xu et al., 2023). Recent work has further advanced CP through heterogeneous collaboration, communication-efficient feature exchange, and unified spatiotemporal aggregation (Gao et al., 2025; Xu et al., 2025), strengthening its role in reliable autonomous driving (Huang et al., 2025; Bai et al., 2024; Dao et al., 2024).

However, the reliance on information from remote collaborators also exposes CP to a critical attack surface (Zhang et al., 2026a). Once a legitimate collaborator is compromised, an adversary can manipulate its shared intermediate features before fusion, causing the corrupted information to propagate into the ego vehicle's perception results. Prior studies have demonstrated that such attacks can fabricate nonexistent objects, suppress real objects, or substantially degrade collaborative detection (Tu et al., 2021; Zhang et al., 2024; Wang et al., 2025; He et al., 2026; Lin et al., 2025). More importantly, the information-sharing mechanism that enables CP to benefit from complementary observations also creates an inherent security risk, since the ego vehicle cannot directly verify whether the received features faithfully reflect a collaborator's actual observations. Robustly handling potentially corrupted collaborative features is therefore essential for secure CP.

Existing defenses largely address this threat through a *detect-and-reject* paradigm. They typically identify malicious collaborators or suspicious information through consensus verification, anomaly detection, and trust estimation, and subsequently exclude, suppress, mask, or downweight the detected information (Li et al., 2023b; Zhao et al., 2024; Hu et al., 2025; Wang et al., 2025; Yu et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2026; Zhang et al., 2026b; Tao et al., 2026). Recent studies have further incorporated finer-grained spatial consistency to identify unreliable regions (Zhang et al., 2024; Liu et al., 2026). Despite increasingly sophisticated detection mechanisms, existing studies still predominantly adopt rejection-oriented defenses, excluding suspicious CAVs or regions from collaborative perception.

The key limitation of the *detect-and-reject* paradigm is that an attacked CAV is not necessarily unreliable as a whole. Adversarial manipulation may corrupt only part of its shared representation, while substantial benign and non-redundant information can remain useful to the ego vehicle. This distinction is particularly important in CP, where perception gains rely on complementary observations that may not be available from other CAVs. This motivates moving beyond the *detect-and-reject* paradigm toward selectively restoring corrupted features while preserving unaffected collaborative information. Realizing this shift requires first determining ***where*** corruption occurs, then deciding ***how*** the affected features should be restored, and finally ***verifying*** whether the repaired features still adversely affect perception.

To this end, we propose `Gecko`, a repair-oriented defense framework for recovering reliable CP under adversarial attacks. To determine *where* corruption occurs and *how* to restore the affected information, `Gecko` first performs *Feature-Space Corruption Localization and Repair*. It constructs a learned benign prior and a leave-one-out collaborative consensus as complementary recovery references, and uses cross-vehicle support to localize suspicious regions and select appropriate repair sources. To *verify* whether the repaired features still adversely affect perception, `Gecko` then performs *Prediction-Space Verification and Correction*, which compares the resulting detector responses with consensus-based reference responses and maps residual positive amplification back to the corresponding feature regions for further correction. Experiments on OPV2V and V2V4Real across diverse attacks and CP models show that `Gecko` consistently outperforms rejection-based defenses. Compared with directly rejecting malicious CAVs, `Gecko` achieves relative improvements of 7.63% in AP@0.5 and up to 14.05% in AP@0.7, while remaining effective under varying attacker populations and adaptive attacks.

2 Related Work

Collaborative Perception. Collaborative perception (CP) enables connected and autonomous vehicles to overcome limited sensing range and occlusion by sharing complementary observations. Existing methods generally adopt early, intermediate, or late fusion, among which intermediate fusion offers a favorable balance between information richness and communication efficiency. Representative approaches improve cross-vehicle feature aggregation through transformer-based interaction and communication-aware spatial selection (Xu et al., 2022a; Hu et al., 2022). Large-scale benchmarks such as OPV2V and V2V4Real have further supported systematic evaluation of CP (Xu et al., 2022b; 2023). Recent methods extend CP toward heterogeneous and communication-efficient collaboration (Gao et al., 2025; Xu et al., 2025), while the reliance on remotely transmitted representations also introduces security risks from potentially compromised CAVs.

Attacks and Defenses in Collaborative Perception. Compromised CAVs can manipulate shared intermediate features to degrade collaborative detection, fabricate objects, or remove safety-critical targets (Tu et al., 2021; Zhang et al., 2024; Wang et al., 2025). Existing defenses largely follow a *detect-and-reject* paradigm, identifying malicious CAVs through consensus verification, anomaly detection, or trust estimation and then excluding or downweighting their contributions (Li et al., 2023b; Zhao et al., 2024; Hu et al., 2025; Wang et al., 2025; Zhang et al., 2026b; Hallyburton & Pajic, 2025). Some methods further localize conflicting or uncertain regions, but still mitigate them through masking or suppression (Zhang et al., 2024; Liu et al., 2026). In contrast, our work considers that attacked CAVs may retain substantial valid information and proposes to selectively restore corrupted intermediate features while preserving unaffected collaborative information.

3 System and Threat Model

3.1 System Model

We consider an intermediate-fusion CP system consisting of an ego CAV and multiple neighboring CAVs. At each time step, each CAV extracts an intermediate feature representation from its local

sensor observation and transmits it to the ego CAV. After spatial alignment, the ego CAV aggregates its own feature with the received features through a fusion module and produces the final perception output using a detection decoder. `Gecko` is deployed after feature alignment and before collaborative fusion. It processes the feature representation received from each non-ego CAV, selectively repairs suspicious spatial cells, and passes the resulting features to the original fusion and detection modules without modifying the underlying CP architecture.

3.2 Threat Model

Following prior studies on adversarial CP (Tu et al., 2021; Zhang et al., 2024; Wang et al., 2025), we assume that one or more legitimate non-ego CAVs may be compromised and manipulate their shared intermediate features before fusion. The adversary may induce untargeted degradation or targeted perception errors, such as suppressing safety-critical objects, while maintaining valid CAV identities and communication formats. The defender has no prior knowledge of the identities or number of compromised CAVs, the corrupted spatial regions, or the attack strategy. We assume that the ego CAV and its local perception pipeline remain trustworthy, although its observations may be incomplete because of occlusion and limited sensing range. `Gecko` aims to restore corrupted feature components while preserving unaffected collaborative information. Attacks against the ego CAV and communication-level threats, including jamming, denial-of-service, replay, packet dropping, and Sybil attacks, are outside the scope of this work.

4 Defense Framework

4.1 Defense Challenges

Developing a restoration-oriented defense for CP involves three key challenges. **C1: Precise localization of corrupted features.** Cross-CAV feature differences may naturally arise from viewpoint changes, occlusion, sensor noise, and spatial misalignment, making adversarial corruption difficult to distinguish from benign variations. **C2: Faithful recovery without clean references.** Once suspicious cells are identified, the defense must reconstruct task-relevant features without access to their ground-truth benign counterparts, while the ego CAV and other CAVs may provide only incomplete observations of the affected regions. **C3: Preserving benign information during recovery.** The defense must sufficiently mitigate adversarial corruption without unnecessarily modifying unaffected features, since overly aggressive restoration can destroy useful collaborative information, whereas insufficient restoration may leave residual corruption in the fusion process.

4.2 Overview

Effective defense in CP requires suppressing adversarial corruption while retaining benign information that contributes to CP. However, existing defenses largely follow a *detect-and-reject* paradigm, in which suspicious CAVs or regions are excluded before fusion. Such rejection can discard benign and non-redundant observations together with corrupted information, weakening the very complementary perception that CP is designed to provide. To overcome this limitation, we propose `Gecko`, a repair-oriented defense framework that recovers CP by selectively restoring corrupted collaborative information rather than discarding it. `Gecko` first determines where corruption occurs and how the affected information should be restored in the feature space, and then verifies the repaired features in the prediction space to identify any remaining harmful effects. The complete procedure is summarized in Appendix A.1.

As illustrated in Figure 1, `Gecko` realizes this recovery process through two stages. In the first stage, support-aware corruption localization combines learned and collaborative references with cross-vehicle support to distinguish suspicious regions from benign cross-CAV variations, addressing **C1**. Multi-source reference construction and source selection then provide appropriate recovery targets when clean feature counterparts are unavailable, addressing **C2**. The feature repair further addresses **C3** by adapting the repair strength and source at each location, avoiding unnecessary modification of unaffected information. In the second stage, prediction-space verification checks for residual abnormal effects after repair and strengthens the repair only at the corresponding locations, further addressing **C3** while leaving the remaining repair decisions unchanged.

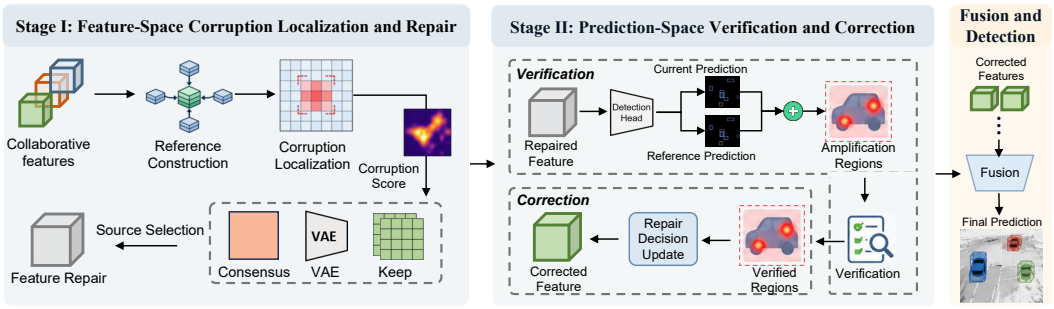

Figure 1: Overview of Gecko. Stage I localizes and repairs corrupted collaborative features using complementary references and support-aware source selection. Stage II evaluates the repaired features in the prediction space and maps verified abnormal responses back to the repair decision before final fusion and detection.

4.3 Feature-Space Corruption Localization and Repair

In the first stage, Gecko localizes and repairs corrupted collaborative features in three steps. Multi-source reference construction builds a VAE-based benign prior, a leave-one-out collaborative consensus, and a support map indicating cross-vehicle support. Support-aware corruption localization uses these references and the support map to estimate spatial corruption, after which feature repair uses the corruption score to determine the repair strength and select an appropriate repair source at each location.

Multi-Source Reference Construction. Reliable feature recovery requires a benign reference, yet no single reference is universally available in CP. Overlapping regions can be inferred from other CAVs, whereas unique-view regions may only be observed by CAV$_i$. To address this issue, we construct two complementary references, including a learned feature prior and a leave-one-out collaborative consensus.

To obtain a learned benign prior, we train a conditional VAE to recover clean collaborative features from perturbed inputs. For CAV$_i$, we perturb its clean feature as

$$F_i^{\text{pert}} = F_i^{\text{clean}} + \delta_i,$$

where δ_i is sampled from a diverse set of training-time perturbations. The reconstruction is conditioned on the ego feature F_e,

$$F_i^{\text{vae}} = \mathcal{V}_\phi\left(F_i^{\text{pert}}, F_e\right). \tag{1}$$

We train the VAE with

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{task}} + \lambda_{\text{rec}} \left\| F_i^{\text{vae}} - F_i^{\text{clean}} \right\|_1 + \beta \mathcal{L}_{\text{KL}}, \tag{2}$$

where $\mathcal{L}_{\text{task}}$ encourages the reconstructed feature to preserve task-relevant information, λ_{rec} weights the reconstruction loss, and β controls the KL regularization. The detector is fixed during training, and the VAE is frozen afterward. At inference time, the reconstruction error is used to derive the spatial anomaly score $e_i^{\text{anom}}(x)$, while F_i^{vae} provides a recovery reference when reliable cross-vehicle observations are unavailable. Detailed training procedures are provided in Appendix A.4.

We further construct a leave-one-out collaborative consensus for each target CAV$_i$ by excluding CAV$_i$ and aggregating the ego feature with the remaining eligible CAVs. Let $\mathcal{E}_i$ denote the eligible non-ego CAVs. For each CAV $j \in \mathcal{E}_i$, we define its reconstruction confidence as $q_j = 1 - \widetilde{E}_j$, where $\widetilde{E}_j \in [0,1]$ denotes the normalized mean VAE reconstruction error of CAV$_j$. The consensus feature at spatial location x is then computed as

$$F_i^{\text{cons}}(x) = \frac{F_0(x) + \sum_{j \in \mathcal{E}_i} q_j F_j(x)}{1 + \sum_{j \in \mathcal{E}_i} q_j}. \tag{3}$$

Since cross-vehicle observations may be unavailable at some locations, we define

$$G_i(x) = \mathbb{I}\left[\max_{j \in \mathcal{E}_i} a_j(x) \geq \tau_{\text{sup}}\right], \tag{4}$$

where $a_j(x)$ denotes the feature activity of CAV_j. Here, F_i^{cons} provides the collaborative reference, while $G_i(x)$ indicates whether it is supported at location x.

Support-Aware Corruption Localization. We next combine complementary evidence to estimate a spatial corruption score. The VAE anomaly $e_i^{\text{anom}}(x)$ measures deviation from the learned benign prior. Where cross-vehicle support exists, we additionally compare the received feature with the leave-one-out consensus,

$$e_i^{\text{dev}}(x) = \mathcal{D}\left(F_i(x), F_i^{\text{cons}}(x)\right), \tag{5}$$

where $\mathcal{D}$ denotes the normalized feature deviation.

To capture perturbations that suppress valid detector responses, we introduce a CAV-level suppression score $s_i \in [0,1]$, which measures the extent to which CAV_i suppresses detector responses that can be recovered after replacing its feature with the leave-one-out consensus. Let $P^{\text{raw}}(x)$ denote the detector response at location x using the received features, and let $P_i^{\text{rep}}(x)$ denote the response after replacing F_i with F_i^{cons}. We compute

$$s_i = \sigma\left(\frac{\max_x\left(\max(P_i^{\text{rep}}(x) - P^{\text{raw}}(x),\, 0)\right) - \tau_s}{T_s}\right), \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid function, τ_s is the suppression threshold, and T_s controls the transition sharpness. A larger s_i indicates that replacing CAV_i restores a stronger detector response, providing stronger evidence that its original feature suppresses valid detections.

To reduce the influence of this suppression evidence on globally anomalous frames, we introduce a frame-level reliability factor $w_f \in [0,1]$. Formally, $w_f = (1 - r_f)^{\gamma}$, where

$$r_f = \max_{i \in \mathcal{E}} \frac{1}{|\Omega|} \sum_{x \in \Omega} e_i^{\text{anom}}(x), \tag{7}$$

$\mathcal{E}$ denotes the set of collaborating vehicles, Ω denotes the set of spatial locations in the feature map, and $e_i^{\text{anom}}(x)$ represents the anomaly score of vehicle i at location x. Thus, r_f captures the maximum spatially averaged anomaly level among all collaborators in the current frame, while γ controls the attenuation strength. A larger r_f yields a smaller w_f, thereby reducing the contribution of s_i in highly anomalous frames.

We then compute a spatial corruption score according to the available cross-vehicle support. Let $p_i^{\text{sup}}(x) \in [0,1]$ denote the corruption score for a supported location, where anomaly, cross-vehicle deviation, and suppression evidence are all available. In contrast, $p_i^{\text{uni}}(x) \in [0,1]$ denotes the corruption score for a unique-view location, where cross-vehicle deviation is excluded because no independent observation is available. Accordingly, the corruption score at supported locations is computed by combining the anomaly, deviation, and suppression evidence as

$$p_i^{\text{sup}}(x) = 1 - \left(1 - e_i^{\text{anom}}(x)\right)\left(1 - e_i^{\text{dev}}(x)\right)\left(1 - w_f s_i\right). \tag{8}$$

For unique-view locations, where independent cross-vehicle evidence is unavailable, the deviation term is excluded, yielding

$$p_i^{\text{uni}}(x) = 1 - \left(1 - e_i^{\text{anom}}(x)\right)\left(1 - w_f s_i\right). \tag{9}$$

The final corruption score is then selected according to the support map $G_i(x)$:

$$p_i(x) = G_i(x) p_i^{\text{sup}}(x) + \left(1 - G_i(x)\right) p_i^{\text{uni}}(x). \tag{10}$$

Feature Repair. After localization, we select a repair source $S_i(x)$ for each location x according to the available cross-vehicle support. Supported regions use the leave-one-out consensus, whereas unique-view regions use the VAE reconstruction only when the anomaly evidence exceeds a threshold τ_{vae}. Otherwise, the original feature is preserved. Formally,

$$S_i(x) = \begin{cases} F_i^{\text{cons}}(x), & G_i(x) = 1, \\ F_i^{\text{vae}}(x), & G_i(x) = 0 \wedge e_i^{\text{anom}}(x) \geq \tau_{\text{vae}}, \\ F_i(x), & \text{otherwise.} \end{cases} \tag{11}$$

The repaired feature $\hat{F}_i(x)$ is then obtained as

$$\hat{F}_i(x) = \left(1 - p_i(x)\right) F_i(x) + p_i(x) S_i(x). \tag{12}$$

Here, $p_i(x)$ controls the repair strength, while $S_i(x)$ determines the repair source. This selective repair preserves unique-view information when no reliable cross-vehicle reference is available. However, even a repair that appears reasonable in the feature space may not fully eliminate its effect on the detector. We next examine this through the downstream prediction changes induced by the repaired features.

4.4 Prediction-Space Verification and Correction

Feature-space analysis helps localize corruption and select appropriate repair references, while the prediction space reveals how the repaired features affect the perception task. Our intuition is that if a repaired CAV still induces abnormal positive responses, replacing its feature with the leave-one-out consensus should reduce these responses. We therefore use this prediction change to verify the feature-space repair and map the identified locations back to the same repair decision.

Prediction-Space Verification. To assess the task-level effect of each repaired CAV, we examine how its presence changes the detector response after feature-space repair. Specifically, if a repaired CAV still induces abnormal positive responses, replacing its feature with the corresponding leave-one-out consensus should attenuate these responses. This comparison therefore provides prediction-space evidence for identifying the locations that require correction. Formally, let

$$\hat{\mathcal{F}} = \left\{ F_e, \hat{F}_1, \ldots, \hat{F}_N \right\} \tag{13}$$

denote the repaired collaborative feature set, where N is the number of non-ego CAVs. Let $\mathcal{P}(\cdot)$ denote the spatial response map produced by the frozen detection head. The response obtained from all repaired features is

$$P^{\text{full}} = \mathcal{P}\left(\hat{\mathcal{F}}\right). \tag{14}$$

For CAV$_i$, we construct a reference feature set $\hat{\mathcal{F}}_i^{\text{ref}}$ by replacing only $\hat{F}_i$ with its leave-one-out consensus F_i^{cons}, while keeping all other repaired features unchanged. Applying the same prediction operator yields the reference response

$$P_i^{\text{ref}} = \mathcal{P}\left(\hat{\mathcal{F}}_i^{\text{ref}}\right). \tag{15}$$

We then quantify the positive prediction change using an amplification map $A_i(x)$,

$$A_i(x) = \mathcal{R}\left(\max\left(P^{\text{full}} - P_i^{\text{ref}}, 0\right)\right)(x), \tag{16}$$

where $\mathcal{R}(\cdot)$ denotes spatial interpolation to the feature-map resolution. A larger $A_i(x)$ indicates a stronger positive response attributable to CAV$_i$. Suppression effects are already incorporated into feature-space localization through s_i, so this verification focuses on positive amplification.

A few isolated positive responses may arise from benign prediction fluctuations and should not trigger correction. We therefore examine whether the amplification forms a sufficiently widespread spatial pattern. Let Ω denote the spatial feature grid. Accordingly, the amplification ratio ρ_i is computed as the fraction of locations whose amplification exceeds a local threshold τ_{cell},

$$\rho_i = \frac{1}{|\Omega|} \sum_{x \in \Omega} \mathbb{I}\left[A_i(x) > \tau_{\text{cell}}\right]. \tag{17}$$

Then, we use a binary indicator $g_i \in \{0, 1\}$ to determine whether this CAV requires prediction-space correction,

$$g_i = \mathbb{I}\left[\rho_i \geq \tau_{\text{ratio}}\right], \tag{18}$$

where τ_{ratio} specifies the minimum spatial coverage required to activate correction. In this way, τ_{cell} filters weak local responses, while τ_{ratio} prevents sparse benign fluctuations from triggering correction.

Repair Correction. Once a CAV is verified by the prediction-space criterion, the amplification map further indicates where its feature-space repair should be strengthened. We construct a correction mask $M_i(x)$ that selects locations with positive amplification,

$$M_i(x) = g_i \, \mathbb{I}\left[A_i(x) > 0\right]. \tag{19}$$

Table 1: Defense performance comparison on OPV2V and V2V4Real under different attacks. We report ORR/ASR (%)↓, AP@0.5↑, and AP@0.7↑. Upper Bound denotes the clean CP performance without attacks or defenses.

Method	MOR			TOR (In)			TOR (Out)			TOR (Random)			BIM		PGD	
	ORR/ASR	AP@0.5	AP@0.7	ASR	AP@0.5	AP@0.7	ASR	AP@0.5	AP@0.7	ASR	AP@0.5	AP@0.7	AP@0.5	AP@0.7	AP@0.5	AP@0.7
OPV2V																
Upper Bound	–	0.91	0.83	–	0.91	0.83	–	0.91	0.83	–	0.91	0.83	0.91	0.83	0.91	0.83
No Defense	100.00/99.95	0.00	0.00	99.61	0.89	0.79	98.79	0.90	0.79	98.09	0.90	0.79	0.00	0.00	0.00	0.00
ROBOSAC	10.52/0.20	0.71	0.55	96.34	0.89	0.78	96.78	0.89	0.78	95.90	0.89	0.78	0.71	0.55	0.71	0.55
LUCIA	7.52/0.15	0.84	0.72	6.05	0.84	0.72	47.25	0.84	0.72	9.02	0.84	0.72	0.84	0.72	0.84	0.72
CP-Guard	7.82/0.20	0.84	0.72	97.42	0.89	0.79	97.32	0.89	0.79	95.32	0.89	0.79	0.83	0.71	0.83	0.71
PRBI	11.00/0.20	0.79	0.66	89.00	0.88	0.84	91.00	0.88	0.84	90.00	0.88	0.84	0.79	0.66	0.79	0.66
Cerberus	7.54/0.15	0.84	0.72	4.19	0.84	0.72	32.62	0.84	0.72	8.92	0.84	0.72	0.84	0.72	0.84	0.72
Gecko	**2.34/0.09**	**0.90**	**0.82**	**3.12**	0.90	0.82	**2.88**	0.90	0.82	**2.58**	0.90	0.82	**0.90**	**0.81**	**0.90**	**0.82**
V2V4Real																
Upper Bound	–	0.61	0.32	–	0.61	0.32	–	0.61	0.32	–	0.61	0.32	0.61	0.32	0.61	0.32
No Defense	86.29/39.22	0.09	0.05	27.79	0.58	0.40	36.94	0.58	0.40	30.61	0.58	0.40	0.02	0.01	0.02	0.01
ROBOSAC	33.78/9.70	0.42	0.25	20.48	0.54	0.37	37.11	0.55	0.38	29.14	0.55	0.38	0.45	0.27	0.45	0.27
LUCIA	29.04/9.05	0.45	0.27	7.64	0.45	0.27	33.53	0.45	0.27	18.58	0.45	0.27	0.45	0.27	0.45	0.27
CP-Guard	33.57/9.43	0.42	0.25	19.66	0.55	0.37	37.05	0.55	0.38	27.25	0.55	0.38	0.45	0.27	0.45	0.27
PRBI	32.00/5.00	0.44	0.26	25.00	0.56	0.38	35.00	0.56	0.39	28.00	0.56	0.39	0.45	0.27	0.45	0.27
Cerberus	57.76/26.76	0.27	0.16	17.06	0.49	0.31	36.51	0.49	0.31	25.41	0.49	0.31	0.45	0.27	0.26	0.16
Gecko	**14.52/2.17**	**0.55**	**0.32**	**2.33**	0.57	0.33	**11.48**	0.57	0.33	**8.72**	0.58	0.33	**0.56**	**0.32**	**0.56**	**0.32**

The correction follows a simple rule. At locations selected by $M_i(x)$, the previous repair decision is replaced by the most conservative supported choice: the repair strength is set to one and the leave-one-out consensus is used as the repair source. All other locations retain the repair strength $p_i(x)$ and repair source $S_i(x)$ obtained in the feature space. We denote the resulting repair strength and repair source by $p'_i(x)$ and $S'_i(x)$, respectively. Accordingly, the final repaired feature F_i^* is obtained by applying the same repair operator in equation 12 with $p'_i(x)$ and $S'_i(x)$. When $g_i = 0$, no location is selected and the feature-space repair remains unchanged. Otherwise, prediction-space evidence strengthens the repair only at locations exhibiting verified positive amplification before collaborative fusion.

5 Experiments

5.1 Experimental Setups

Following prior works (Zhang et al., 2026b; Zhao et al., 2024), we evaluate our Gecko primarily on the OPV2V (Xu et al., 2022b) and further assess its generalization on V2V4Real (Xu et al., 2023). We consider four representative attacks, including PGD, BIM, MOR (Wang et al., 2025), and TOR (Wang et al., 2025), together with a white-box adaptive attack that explicitly optimizes against our Gecko. To assess generalization across fusion architectures, we conduct experiments on AttentiveFusion (Xu et al., 2022b), CoAlign (Lu et al., 2023), V2VAM (Li et al., 2023a), and Where2comm (Hu et al., 2022). We compare our method with representative defenses, including ROBOSAC (Li et al., 2023b), CP-Guard (Hu et al., 2025), LUCIA (Wang et al., 2025), PRBI (Yu et al., 2026), and Cerberus (Zhang et al., 2026b). We evaluate defense performance using Attack Success Rate (ASR), Object Removal Rate (ORR), and AP@0.5/0.7. Detailed configurations and hyperparameters are provided in Appendix A.2.

5.2 Defense Effectiveness

Comparison with Existing Defenses. Table 1 compares Gecko with representative CP defenses on both OPV2V and V2V4Real, where clean CP performance without attacks and defenses serves as the upper bound. Across both benchmarks, Gecko consistently restores perception performance close to the clean upper bound while substantially reducing attack-specific ORR and ASR. This trend holds across MOR, TOR, BIM, and PGD, where Gecko generally preserves higher AP than existing defenses. The results highlight the advantage of selective feature restoration over rejection-oriented defenses. Rather than discarding an entire suspicious CAV, Gecko repairs corrupted features while retaining useful collaborative information, allowing the system to recover toward the performance of clean full collaboration.

Robustness to the Number of Attackers. Figure 2 evaluates the robustness of our method as the number of attackers increases from one to four. Under both MOR and TOR, our method consistently maintains low ORR/ASR, while the undefended system remains highly vulnerable. Meanwhile, the defended AP@0.5 stays largely stable as more malicious CAVs participate. A moderate drop is observed under MOR with four attackers, as more collaborative information becomes corrupted. Nevertheless, our method still preserves high perception accuracy, demonstrating strong robustness to varying numbers of malicious CAVs.

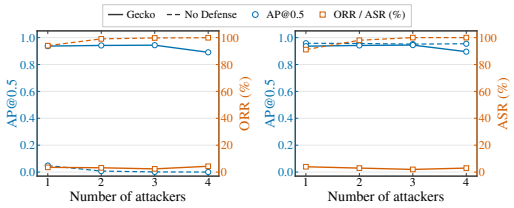

Figure 2: Impact of the number of attackers on defense performance. The left and right panels show MOR and TOR results, respectively.

5.3 Robustness Analysis

Cross-Model Adaptability. Table 2 evaluates our method across four representative CP models. Overall, our method maintains effective defense performance under both MOR and TOR despite substantial differences in fusion architectures. The ORR remains between 2.34% and 7.99% under MOR, while the ASR ranges from 2.58% to 9.41% under TOR. Meanwhile, the defended AP@0.5 remains at 0.79–0.90 across all models. These results show that our method is not limited to a specific fusion architecture and can effectively restore corrupted collaborative features across different CP models, demonstrating strong cross-model adaptability.

Table 2: Defense performance across different CP models.

Model	MOR		TOR	
	ORR	AP@0.5	ASR	AP@0.5
AttentiveFusion	2.34	0.90	2.58	0.90
CoAlign	4.64	0.79	5.17	0.79
Where2comm	4.74	0.84	5.17	0.85
V2VAM	7.99	0.85	9.41	0.86

Robustness to Adaptive Attacks. To evaluate Gecko under adaptive attacks, we construct a strong white-box attacker with full knowledge of the defense mechanism. The attack augments the standard detection objective with a defense-evasion objective that suppresses the evidence used by Gecko for corruption localization and repair, where λ controls the trade-off between the two objectives and $\lambda = 0$ corresponds to the standard attack. As shown in Figure 3, the defended AP gradually decreases as λ increases, indicating that the adaptive objective can effectively challenge the defense. Nevertheless, Gecko maintains relatively high perception accuracy across all evaluated settings and remains substantially more robust than the undefended model. These results demonstrate that Gecko remains effective even when the attacker has complete knowledge of the defense.

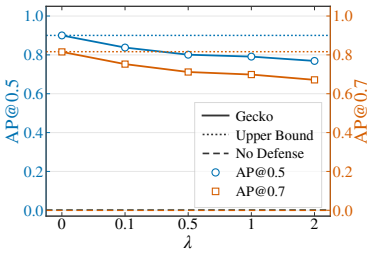

Figure 3: Defense performance against adaptive attacks with varying evasion weights λ. Our method maintains high AP@0.5 and AP@0.7 across different λ.

Performance–Latency Trade-off. Figure 4 compares the performance–latency trade-off of different defense methods. Gecko achieves the highest AP@0.5 and AP@0.7 among all evaluated methods, while introducing moderately higher latency than lightweight defenses such as LUCIA and PRBI. This additional cost is accompanied by a clear improvement in perception accuracy, and Gecko remains faster than ROBOSAC. Overall, these results show that Gecko provides a favorable trade-off between defense effectiveness and computational efficiency, despite the additional feature restoration and verification operations.



Figure 4: Performance–latency trade-off of different defense methods. The left and right panels show AP@0.5 and AP@0.7 versus defense latency, respectively.

Table 3: Ablation study under MOR, TOR, PGD, and BIM attacks, where "w/o" denotes "without". The complete Gecko with all components achieves the most consistent defense performance across different attacks.

Attack	MOR			TOR			PGD		BIM	
Method Variant	ORR↓	AP@0.5↑	AP@0.7↑	ASR↓	AP@0.5↑	AP@0.7↑	AP@0.5↑	AP@0.7↑	AP@0.5↑	AP@0.7↑
w/o anomaly	65.83%	0.31	0.24	66.21%	0.93	0.91	0.09	0.07	0.02	0.01
w/o deviation	2.61%	0.91	0.82	2.78%	0.91	0.83	0.90	0.82	0.90	0.81
w/o suppression	2.34%	0.90	0.82	2.58%	0.90	0.82	0.90	0.81	0.90	0.81
w/o correction	2.34%	0.90	0.82	2.58%	0.90	0.82	0.90	0.81	0.90	0.81
Our Gecko (Full)	2.34%	0.90	0.82	2.58%	0.90	0.82	0.90	0.82	0.90	0.81

5.4 Ablation Study

Impact of Each Component on Defense Performance. Table 3 examines the contribution of each component across four attacks. Removing anomaly evidence leads to the most pronounced degradation. Under MOR, the ORR increases from 2.34% to 65.83%, while AP@0.5/0.7 drops from 0.90/0.82 to 0.31/0.24. A similar trend is observed under TOR, where the ASR rises from 2.58% to 66.21%. Its removal also severely degrades performance under PGD and BIM, reducing AP@0.5 to 0.09 and 0.02, respectively. Removing deviation evidence slightly increases the ORR and ASR under MOR and TOR, while the overall AP remains comparable. In contrast, removing suppression evidence or prediction-space correction results in only marginal changes under these attacks. Overall, the results identify anomaly evidence as the dominant component for reliable corruption localization, while the remaining components provide complementary support to the full defense.

Repair vs. Rejection. Figure 5 compares Gecko with oracle reject, an idealized strategy that perfectly identifies and removes the malicious CAV before fusion. Across MOR and different TOR settings, Gecko consistently achieves higher AP@0.5 while maintaining lower ORR/ASR. The difference is particularly pronounced under TOR (Out), where removing the malicious CAV leads to a substantial increase in ASR. This result suggests that even perfect rejection may discard useful observations provided by the affected CAV. In contrast, Gecko repairs corrupted features while preserving useful collaborative information, allowing the system to better recover the perception capability of full collaboration.

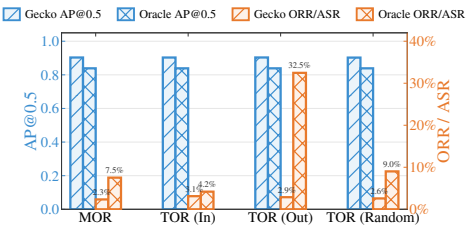

Figure 5: Comparison between Gecko and Oracle Reject under MOR and TOR attacks.

Additional Results. Additional experiments in Appendix A.5 further evaluate Gecko through qualitative visualizations, clean-scene performance, repair under limited collaboration, sensitivity to the collaborative support and reconstruction anomaly thresholds, and robustness to colluding attackers. These results provide complementary analyses of how Gecko preserves benign perception, behaves under different collaboration conditions and parameter settings, and remains effective when multiple malicious CAVs coordinate their attacks.

6 Conclusion

In this paper, we revisit the prevailing *detect-and-reject* paradigm for securing CP and show that corrupted shared features need not be treated as entirely unusable. We propose Gecko, a repair-oriented defense that selectively restores corrupted collaborative features by localizing and repairing corruption in the feature space and verifying the resulting repair in the prediction space. Experiments on OPV2V and V2V4Real across multiple attacks and CP architectures show that Gecko consistently recovers perception performance, remains robust under adaptive attacks, and provides particularly clear advantages when collaborative redundancy is limited. These results demonstrate that Gecko can repair corrupted collaborative features to recover perception performance without discarding useful collaborative information.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing code for implementing the proposed method and with translation. We did not use generative AI tools for other tasks requiring disclosure, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to edit portions of the manuscript to improve readability and language clarity. All AI-assisted code was manually reviewed, verified, and tested by the authors for correctness. All AI-assisted translations and language edits were also carefully reviewed by the authors to ensure that the technical meaning, terminology, and scientific claims were accurately preserved. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the security of collaborative perception systems and proposes a defense against malicious manipulation of shared intermediate features. Our experiments are conducted using established collaborative perception research datasets and do not involve human-subject studies or the collection of new personal data. The adversarial attacks considered in this work are used solely to evaluate the robustness of collaborative perception and the effectiveness of the proposed defense in controlled research settings. While such attack techniques may have potential dual-use implications, our goal is to identify vulnerabilities and improve the robustness and security of collaborative perception systems. We encourage the use of the methods and findings presented in this work for defensive research, robustness evaluation, and responsible security testing.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, experimental settings, attack configurations, and evaluation metrics in the main paper and appendix. Additional implementation details, hyperparameter settings, and supplementary experimental results are also included to facilitate reproduction of our results.

REFERENCES

Zhengwei Bai, Guoyuan Wu, Matthew J. Barth, Yongkang Liu, Emrah Akin Sisbot, Kentaro Oguchi, and Zhitong Huang. A survey and framework of cooperative perception: From heterogeneous singleton to hierarchical cooperation. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):15191–15209, 2024.

Minh-Quan Dao, Julie Stephany Berrio, Vincent Frémont, Mao Shan, Elwan Héry, and Stewart Worrall. Practical collaborative perception: A framework for asynchronous and multi-agent 3d object detection. *IEEE Transactions on Intelligent Transportation Systems*, 25(9):12163–12175, 2024.

Xiangbo Gao, Runsheng Xu, Jiachen Li, Ziran Wang, Zhiwen Fan, and Zhengzhong Tu. Stamp: Scalable task and model-agnostic collaborative perception. *arXiv preprint arXiv:2501.18616*, 2025.

R Spencer Hallyburton and Miroslav Pajic. Security-aware sensor fusion with mate: the multi-agent trust estimator. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2009–2023, 2025.

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

Ji He, Ying Wang, Lijie Zheng, Xinghui Zhu, Yulong Shen, and Xiaohong Jiang. GLST: Defending confidence-driven v2x collaborative perception against stealthy multi-attacker feature injection. *arXiv preprint arXiv:2607.23059*, 2026.

Senkang Hu, Yihang Tao, Guowen Xu, Yiqin Deng, Xianhao Chen, Yuguang Fang, and Sam Kwong. CP-Guard: Malicious agent detection and defense in collaborative bird's eye view perception. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 23203–23211. AAAI Press, 2025.

Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, 35:4874–4886, 2022.

Tao Huang, Jianan Liu, Xi Zhou, Dinh C. Nguyen, Mostafa Rahimi Azghadi, Yuxuan Xia, Qing-Long Han, and Sumei Sun. Vehicle-to-everything cooperative perception for autonomous driving. *Proceedings of the IEEE*, 113(5):443–477, 2025.

Jinlong Li, Runsheng Xu, Xinyu Liu, Jin Ma, Zicheng Chi, Jiaqi Ma, and Hongkai Yu. Learning for vehicle-to-vehicle cooperative perception under lossy communication. *IEEE Transactions on Intelligent Vehicles*, 8(4):2650–2660, 2023a.

Yiming Li, Qi Fang, Jiamu Bai, Siheng Chen, Felix Juefei-Xu, and Chen Feng. Among US: Adversarially robust collaborative perception by consensus. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 186–195, 2023b.

Hongwei Lin, Dongyu Pan, Qiming Xia, Hai Wu, Cheng Wang, Siqi Shen, and Chenglu Wen. Pretend benign: A stealthy adversarial attack by exploiting vulnerabilities in cooperative perception. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19947–19956. IEEE, 2025.

Yutong Liu, Chenyi Wang, Ming F Li, and Qingzhao Zhang. Adversarial trust poisoning in vehicular collaborative perception. *arXiv preprint arXiv:2605.22122*, 2026.

Yifan Lu, Quanhao Li, Baoan Liu, Mehrdad Dianati, Chen Feng, Siheng Chen, and Yanfeng Wang. Robust collaborative 3d object detection in presence of pose errors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4812–4818. IEEE, 2023.

Yihang Tao, Senkang Hu, Haonan An, Zhengru Fang, Hangcheng Cao, and Yuguang Fang. Learning mutual view information graph for adaptive adversarial collaborative perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 42321–42330, June 2026.

James Tu, Tsunhsuan Wang, Jingkang Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Adversarial attacks on multi-agent communication. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7768–7777, 2021.

Chenyi Wang, Raymond Muller, Ruoyu Song, Jean-Philippe Monteuuis, Jonathan Petit, Yanmao Man, Ryan Gerdes, Z Berkay Celik, and Ming Li. From Threat to Trust: Exploiting attention mechanisms for attacks and defenses in cooperative perception. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 7387–7406, 2025.

Junhao Xu, Yanan Zhang, Zhi Cai, and Di Huang. Cosdh: communication-efficient collaborative perception via supply-demand awareness and intermediate-late hybridization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6834–6843, 2025.

Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision*, pp. 107–124. Springer, 2022a.

Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2583–2589. IEEE, 2022b.

Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, et al. V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13712–13722. IEEE, 2023.

Yi Yu, Libing Wu, Zhuangzhuang Zhang, Jing Qiu, Lijuan Huo, and Jiaqi Feng. All vehicles can lie: Efficient adversarial defense in fully untrusted-vehicle collaborative perception via pseudo-random bayesian inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6549–6558, June 2026.

Qingzhao Zhang, Shuowei Jin, Ruiyang Zhu, Jiachen Sun, Xumiao Zhang, Qi Alfred Chen, and Z Morley Mao. On data fabrication in collaborative vehicular perception: Attacks and countermeasures. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 6309–6326, 2024.

Qingzhao Zhang, Runting Zhang, and Z Morley Mao. From stealthy data fabrication to unsafe driving: Realistic scenario attacks on collaborative perception. *arXiv preprint arXiv:2605.01301*, pp. 1–16, 2026a.

Zhuangzhuang Zhang, Mingxin Li, Libing Wu, Wei-Bin Lee, and Jianping Wang. The latent guardian: Defending collaborative perception via feature-level consistency verification. In *Forty-third International Conference on Machine Learning*, 2026b.

Yangheng Zhao, Zhen Xiang, Sheng Yin, Xianghe Pang, Yanfeng Wang, and Siheng Chen. MADE: Malicious agent detection for robust multi-agent collaborative perception. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13817–13823, 2024.

A Appendix

A.1 Algorithmic Description of Gecko

Algorithm 1 The main workflow of Gecko.

Input: Ego feature F_e, collaborator features $\{F_i\}_{i=1}^{N}$, trained VAE $\mathcal{V}_\phi$, frozen prediction operator $\mathcal{P}$
Output: Corrected features $\{F_i^*\}_{i=1}^{N}$
Stage I: Feature-Space Corruption Localization and Repair
for $i = 1$ to N **do**
  Obtain F_i^{vae} ▷ Learned benign prior
  Construct F_i^{cons} and G_i using equation 3 and equation 4 ▷ Collaborative consensus and support
  Compute p_i using equations 5–10 ▷ Localize suspicious regions
  Determine S_i using equation 11 ▷ Select repair source
  Compute $\hat{F}_i$ using equation 12 ▷ Feature repair
end for
Stage II: Prediction-Space Verification and Correction
Compute P^{full} using equation 14 ▷ Response after repair
for $i = 1$ to N **do**
  Compute P_i^{ref} using equation 15 ▷ Reference response
  Compute A_i, ρ_i, and g_i using equations 16–18 ▷ Verify abnormal positive responses
  $(p_i', S_i') \leftarrow (p_i, S_i)$ ▷ Retain feature-space repair decisions
  if $g_i = 1$ **then**
    Compute M_i using equation 19 ▷ Identify regions for correction
    $p_i'(x) \leftarrow 1$ for $M_i(x) = 1$
    $S_i'(x) \leftarrow F_i^{\text{cons}}(x)$ for $M_i(x) = 1$ ▷ Strengthen feature repair
  end if
  Obtain F_i^* using equation 12 with p_i' and S_i' ▷ Corrected feature
end for
return $\{F_i^*\}_{i=1}^{N}$

A.2 Implementation Details

Datasets and CP Models. We follow the official data splits of OPV2V and V2V4Real. For OPV2V, the test split contains 2,170 frames, from which we use all 2,051 multi-CAV frames in the main

experiments. These frames contain 2, 3, 4, and 5 CAVs with 970, 869, 111, and 101 frames, respectively. For V2V4Real, we use all 1,846 multi-CAV frames from its 1,993-frame test split. The ego CAV is determined by the first CAV ID after sorting the vehicles in each scene. Following the current evaluation protocol, an OPV2V object is retained only when at least eight of its bounding-box corners fall within the perception range, while V2V4Real follows its original two-corner filtering rule.

AttentiveFusion is used as the default CP model. We additionally evaluate `Gecko` on CoAlign, V2VAM, and Where2comm to study its cross-model generalization. All CP models are pretrained on clean data and remain fixed during attack generation and defense evaluation. The same detector checkpoint and evaluation manifest are used when comparing different defense methods under the same experimental setting. Since the feature distributions of the two datasets differ substantially, `Gecko` uses separately trained VAE checkpoints for OPV2V and V2V4Real. The detailed VAE architecture and training procedure are provided in Appendix A.4.

Attack Settings. All feature-space attacks are applied to the intermediate representation after the PointPillar scatter module and before the BEV backbone and CP fusion. For PGD, we use an L_∞ perturbation budget of $\epsilon = 0.3$, where ϵ denotes the maximum element-wise perturbation magnitude. The perturbation is initialized from $\mathcal{U}(-\epsilon, \epsilon)$ and optimized for $K = 10$ iterations with a step size of $\alpha = 0.1$, followed by projection onto the L_∞ ball after each update. BIM uses the same step size $\alpha = 0.1$ and $K = 10$ iterations, but starts from zero and does not apply an explicit perturbation-budget projection. Both attacks maximize the same detection-oriented objective, which combines a focal classification loss over positive ground-truth anchors with a regression term defined on these anchors. For object-removal attacks, MOR removes all valid ground-truth objects in the current frame, whereas TOR selects a single ground-truth object as the removal target. Both attacks use 10 signed-gradient updates with a step size of 0.1. Their optimization objectives combine the corresponding object-removal loss with the model-specific fusion-attention objective. The same attack hyperparameters are used across the compared defense methods.

Evaluation Metrics. We evaluate attack effectiveness and defense performance using Attack Success Rate (ASR), Object Removal Rate (ORR), and Average Precision (AP). For Targeted Object Removal (TOR), ASR measures the fraction of frames in which the designated target is successfully removed. Let $\mathcal{Q}$ denote the set of evaluated frames and q index a frame. A TOR attack is considered successful when the designated target has no positive IoU overlap with any predicted box and the number of false positives at IoU $= 0.5$ is smaller than two. Accordingly,

$$\mathrm{ASR} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbb{I}\left[d_q^{\mathrm{tar}} = 0 \wedge \mathrm{FP}_q < 2 \right], \tag{20}$$

where d_q^{tar} indicates whether the designated target in frame q has a positive IoU overlap with at least one predicted box, and FP_q denotes the number of false-positive predictions in that frame.

For Mass Object Removal (MOR), we use ORR to measure the reduction in the number of predicted objects relative to the number of ground-truth objects. Let $\mathcal{Q}_+ \subseteq \mathcal{Q}$ denote the subset of frames containing at least one ground-truth object. ORR is computed as

$$\mathrm{ORR} = \frac{1}{|\mathcal{Q}_+|} \sum_{q \in \mathcal{Q}_+} \left(1 - \frac{\min\left(N_q^{\mathrm{pred}}, N_q^{\mathrm{gt}}\right)}{N_q^{\mathrm{gt}}} \right), \tag{21}$$

where N_q^{pred} and N_q^{gt} denote the numbers of predicted and ground-truth objects in frame q, respectively. Higher ASR and ORR indicate stronger attacks, while lower values indicate more effective defense.

We further report AP@0.5 and AP@0.7 to evaluate overall detection performance. These metrics correspond to the area under the precision–recall curve using bird's-eye-view bounding-box IoU thresholds of 0.5 and 0.7, respectively.

`Gecko` Configuration. Unless otherwise specified, the same `Gecko` configuration is used across all attacks. The cross-CAV support threshold τ_{sup} and the VAE-source activation threshold τ_{vae} are both set to 0.5. For prediction-space verification, we set the cell-level amplification threshold to $\tau_{\mathrm{cell}} = 0.02$ and the CAV-level amplification-ratio threshold to $\tau_{\mathrm{ratio}} = 0.036$. Here, τ_{cell} determines whether an individual spatial location exhibits meaningful prediction amplification, while τ_{ratio} determines whether the fraction of amplified locations is sufficiently large to trigger the second repair

stage. Once triggered, all locations with positive amplification are reconsidered, corresponding to $\tau_{\text{veto}} = 0$.

The sigmoid temperature used for suppression evidence is 0.02. For the VAE anomaly score, the reconstruction-error center and sigmoid temperature are set to 0.10 and 0.02, respectively. The deviation and anomaly evidence scales are set to 0.80 and 1.0. When constructing cross-CAV consensus, we require a confidence of at least 0.75 for an additional CAV to participate, and use an activity ratio of 0.20 with temperature 0.05. Temporal-history-based repair is disabled in the reported experiments, and all retained CAVs use unit trust during CP fusion.

Baselines. We compare `Gecko` against ROBOSAC, CP-Guard, LUCIA, and Cerberus using their corresponding implementations in our evaluation framework. ROBOSAC uses a sampling budget of 10 and a consistency threshold of 0.3. For CP-Guard, the recursive consistency threshold is set to 0.3 with a maximum recursion depth of 20. LUCIA uses a feature average-pooling ratio of 32. For Cerberus, the three component weights are set to $(0.3, 0.3, 0.4)$, with SSIM, MDAG, and UA-AT enabled. The UA-AT module uses a base threshold of 0.30, $\gamma = 2.5$, and $\beta = 0.15$ under the entropy-based configuration. These defenses do not require additional training checkpoints and are evaluated using the same CP detector checkpoints and frame manifests as `Gecko`.

A.3 Adaptive White-Box Attack

We also test `Gecko` against a strong adaptive attacker who has full knowledge of the defense and its parameters. Rather than treating detection and defense as separate problems, this attacker optimizes the compromised collaborator's intermediate feature to jointly degrade perception and evade `Gecko`.

The attack is built on standard L_∞ PGD. We initialize $\delta_0 \sim \mathcal{U}(-\epsilon, \epsilon)$ with $\epsilon = 0.3$ and run $T = 10$ iterations at step size $\alpha = 0.1$, updating the perturbation as

$$\delta_{t+1} = \Pi_{\|\delta\|_\infty \le \epsilon} \left(\delta_t + \alpha \operatorname{sign} \left(\nabla_{\delta_t} \mathcal{J}_\lambda \right) \right), \tag{22}$$

with Π the projection onto the L_∞ ball.

The objective combines a detection term and an evasion term,

$$\mathcal{J}_\lambda = \mathcal{L}_{\text{det}} - \lambda \mathcal{L}_{\text{eva}}, \tag{23}$$

with λ setting the balance between the two. The detection term is the same one used in the standard PGD attack,

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{cls}}^{\text{GT}} - \mathcal{L}_{\text{reg}}^{0}, \tag{24}$$

the focal classification loss on the ground-truth positive anchors minus the weighted Smooth-L_1 regression loss on those same anchors toward a zero target.

The evasion term is where the attack becomes defense-aware. It targets the two pieces of evidence `Gecko` uses to localize residual corruption,

$$\mathcal{L}_{\text{eva}} = \overline{p}_{\text{att}} + \mathbb{I}[g = 1]\, \overline{a}, \tag{25}$$

where $\overline{p}_{\text{att}}$ is the compromised collaborator's mean corruption score and $\overline{a}$ is the mean residual amplification left after the first-stage repair, gated by the second-stage amplification indicator g. The residual amplification itself is

$$\overline{a} = \text{mean} \left[\text{ReLU} \left(P_{\text{repaired}} - P_{\text{cf}} \right) \right], \tag{26}$$

comparing the detector's response on the repaired collaborative features against its response under the counterfactual where the compromised collaborator is swapped out for the collaborative consensus.

We sweep $\lambda \in \{0, 0.1, 0.5, 1, 2\}$, with $\lambda = 0$ recovering the plain PGD objective in equation 23. As λ grows, the attacker is pushed to degrade perception while also suppressing the corruption and amplification signals `Gecko` relies on, rather than optimizing either goal alone. Non-differentiable defense operations are handled with BPDA-compatible surrogate gradients. Throughout, both the detector and `Gecko` stay fixed, and gradients are taken only with respect to δ.

A.4 VAE TRAINING PROTOCOL

The VAE is trained to recover benign collaborative representations from corrupted intermediate features. Directly training it as a conventional autoencoder is undesirable in our setting because the feature received from a CAV may already contain adversarial perturbations, in which case self-reconstruction can preserve the corruption. We therefore adopt a corruption-to-clean training strategy. Given benign features extracted by a pretrained CP model, we deliberately corrupt the features of non-ego CAVs and train the VAE to recover their clean counterparts conditioned on the ego feature. Moreover, since feature-level similarity does not necessarily imply equivalent downstream perception behavior, the reconstructed features are additionally constrained to reproduce the detector responses of the corresponding clean CP scene.

Let F_e denote the feature of the ego CAV, where the subscript e denotes the ego vehicle, and let F_i denote the clean feature of CAV_i. For a frame containing M non-ego CAVs, $i \in \{1, \ldots, M\}$. These features are extracted after the PointPillar scatter module and before the BEV backbone and CP fusion. For each CAV_i, we construct a corrupted feature

$$\widetilde{F}_i = \mathrm{sg}\left[\mathrm{clip}\left(F_i + \delta_i, -10^4, 10^4\right)\right], \tag{27}$$

where $\widetilde{F}_i$ denotes the corrupted feature used as the VAE input, δ_i is a synthetic feature-space perturbation applied to CAV_i, and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. Rather than reproducing a particular evaluation-time attack, we sample δ_i from a diverse corruption distribution so that the VAE is exposed to perturbations with different spatial structures, magnitudes, and downstream effects.

Perturbation Generation. We generate either stochastic feature noise or detector-guided projected-gradient perturbations. With probability 0.1, we generate a noise perturbation whose magnitude a is sampled from

$$a \sim \mathcal{U}(0.05, 0.40), \tag{28}$$

where $\mathcal{U}(l, u)$ denotes a uniform distribution over the interval $[l, u]$. The noise type is uniformly selected from high-frequency Gaussian noise, two spatially smoothed Gaussian variants, and uniform noise. The sampled noise tensor is normalized by its maximum absolute magnitude and then rescaled to a.

Otherwise, we construct an iterative L_∞-bounded perturbation. Conditioned on entering this gradient-based branch, we sample the standard, fast, and strong regimes with probabilities 0.65, 0.25, and 0.10, respectively. Their perturbation budgets, denoted by ϵ, are sampled as

$$\epsilon_{\mathrm{std}} \sim \mathcal{U}(0.22, 0.55), \qquad \epsilon_{\mathrm{fast}} \sim \mathcal{U}(0.20, 0.55), \tag{29}$$

and

$$\epsilon_{\mathrm{strong}} \sim \mathcal{U}(0.20, 0.50). \tag{30}$$

Here, ϵ_{std}, ϵ_{fast}, and $\epsilon_{\mathrm{strong}}$ denote the L_∞ budgets of the three perturbation regimes. The standard and fast regimes use 8–15 and 3–8 optimization steps, respectively, whereas the strong regime uses a fixed 15 steps.

For each gradient-based perturbation trajectory, we independently sample a detector output $q \in \{\mathrm{psm}, \mathrm{rm}\}$, where psm and rm denote the raw classification and regression outputs of the CP model, respectively. We also sample an update operator $h(\cdot)$ and keep both q and $h(\cdot)$ fixed throughout the corresponding perturbation trajectory:

$$h(g) = \begin{cases} \mathrm{sign}(g), & \text{signed-gradient update,} \\ g, & \text{raw-gradient update,} \end{cases} \tag{31}$$

where g denotes the gradient of the selected perturbation objective.

Let $D_q(\cdot)$ denote the output q of the frozen CP model D. For CAV_i, we maximize the following objective:

$$\mathcal{J}_q(\delta_i) = \mathrm{mean}\left[\sigma\left(D_q\left(F_e, F_1, \ldots, F_i + \delta_i, \ldots, F_M\right)\right)\right], \tag{32}$$

where $\mathcal{J}_q$ denotes the perturbation-generation objective and $\sigma(\cdot)$ is the sigmoid function. Only the feature of CAV_i is perturbed when constructing δ_i, while all remaining CAVs retain their clean features.

Starting from $\delta_i^{(0)} = 0$, where the superscript k denotes the k-th optimization step, we perform projected gradient ascent:

$$\delta_i^{(k+1)} = \Pi_{\|\delta\|_\infty \le \epsilon} \left[\delta_i^{(k)} + \alpha h \left(\nabla_{\delta_i^{(k)}} \mathcal{J}_q \right) \right], \tag{33}$$

where $\Pi_{\|\delta\|_\infty \le \epsilon}(\cdot)$ denotes projection onto the L_∞ ball with radius ϵ, and α denotes the update step size. We use $\alpha = 1$ in our implementation. The positive update direction explicitly performs gradient ascent on $\mathcal{J}_q$. No ground-truth boxes or detection labels are required during this procedure. By combining stochastic noise with detector-guided perturbations of different strengths and optimization trajectories, the resulting training distribution exposes the VAE to a broad range of feature corruptions rather than a single predefined attack.

Conditional Feature Reconstruction. Given the corrupted feature $\widetilde{F}_i$ of CAV_i, we concatenate it with the ego feature F_e along the channel dimension and feed the resulting conditional input into the VAE encoder. Let E_ϕ denote the encoder parameterized by ϕ. The encoder predicts

$$(\mu_i, \log v_i) = E_\phi \left([\widetilde{F}_i; F_e] \right), \tag{34}$$

where $[\cdot;\cdot]$ denotes channel-wise concatenation, μ_i is the mean of the latent distribution for CAV_i, and $\log v_i$ denotes its log-variance.

During training, the latent representation z_i is sampled using the reparameterization trick:

$$z_i = \mu_i + \exp\left(\frac{1}{2} \log v_i \right) \odot \xi_i, \qquad \xi_i \sim \mathcal{N}(0, I), \tag{35}$$

where ξ_i is standard Gaussian noise, $\mathcal{N}(0, I)$ denotes a standard normal distribution, and $\odot$ denotes element-wise multiplication.

Let G_θ denote the decoder parameterized by θ. The reconstructed feature of CAV_i is

$$\widehat{F}_i = G_\theta(z_i), \tag{36}$$

where $\widehat{F}_i$ denotes the reconstruction of the clean feature F_i. The encoder and decoder are connected through multi-scale skip connections to preserve spatial information that may otherwise be lost through the latent bottleneck. The reconstructed feature $\widehat{F}_i$ has the same spatial resolution and number of channels as F_i.

We supervise the reconstruction using the corresponding clean feature:

$$\mathcal{L}_{\mathrm{rec}} = \frac{1}{M} \sum_{i=1}^{M} \mathrm{mean} \left| \widehat{F}_i - F_i \right|, \tag{37}$$

where $\mathcal{L}_{\mathrm{rec}}$ denotes the average L_1 reconstruction loss over all non-ego CAVs in the frame. Importantly, the reconstruction target is the clean feature F_i, rather than the corrupted input $\widetilde{F}_i$, which explicitly encourages the VAE to remove the injected corruption instead of reproducing it.

Task-Aware Training Objective. Feature-level reconstruction alone does not guarantee that the recovered representation induces the same perception behavior after CP fusion. We therefore further constrain the reconstruction using the frozen CP model.

Let

$$\mathcal{F}_{\mathrm{clean}} = \{F_e, F_1, \ldots, F_M\} \tag{38}$$

denote the complete clean feature set of all CAVs in a frame, and let

$$\mathcal{F}_{\mathrm{recon}} = \{F_e, \widehat{F}_1, \ldots, \widehat{F}_M\} \tag{39}$$

denote the corresponding feature set in which the features of all non-ego CAVs are replaced by their VAE reconstructions.

Passing the two feature sets through the same frozen CP model D gives

$$(P_c, R_c) = D(\mathcal{F}_{\mathrm{clean}}), \qquad (P_r, R_r) = D(\mathcal{F}_{\mathrm{recon}}), \tag{40}$$

where P_c and R_c denote the raw classification and regression outputs obtained from the clean feature set, while P_r and R_r denote those obtained from the reconstructed feature set.

We define the task-consistency loss as

$$\mathcal{L}_{\text{task}} = \text{mean}\,|P_r - \text{sg}(P_c)| + \text{mean}\,|R_r - \text{sg}(R_c)|\,, \tag{41}$$

where $\mathcal{L}_{\text{task}}$ encourages the reconstructed feature set to preserve both the classification and regression responses of the corresponding clean CP scene.

The parameters of the CP model D remain fixed throughout VAE training. However, the reconstructed features $\widehat{F}_i$ are not detached before being passed through D. Therefore, gradients from $\mathcal{L}_{\text{task}}$ propagate through the frozen BEV backbone, CP fusion module, and detection heads to the VAE, allowing the CP model to serve as a fixed task-level supervisor.

We additionally regularize the latent distribution using the KL divergence

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2M}\sum_{i=1}^{M}\text{mean}\left[1 + \log v_i - \mu_i^2 - \exp(\log v_i)\right], \tag{42}$$

where $\mathcal{L}_{\text{KL}}$ encourages the approximate latent posterior to remain close to a standard Gaussian distribution.

The overall VAE training objective is

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{task}} + w_{\text{rec}}\mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{KL}}, \tag{43}$$

where $\mathcal{L}_{\text{VAE}}$ denotes the complete training objective, w_{rec} controls the contribution of the feature reconstruction loss, and β controls the strength of the KL regularization. We set $\beta = 0.005$ and linearly decay w_{rec} from 1.0 to 0.1 over the training epochs. The larger reconstruction weight during early training encourages the VAE to first establish a stable corruption-to-clean mapping, while its gradual reduction increases the relative influence of the task-consistency constraint during later training.

Training Details. We optimize only the VAE parameters while keeping the CP model fixed. The VAE is trained for eight epochs using AdamW with an initial learning rate of 10^{-3} and weight decay 10^{-5}. A cosine learning-rate scheduler decreases the learning rate to 10^{-6}, and the gradient norm of the VAE parameters is clipped to 1.0. Each frame produces one optimization step after the features of all non-ego CAVs have been reconstructed.

During training, the latent representation is sampled according to equation 35. At inference time, we remove the sampling operation and directly use the posterior mean,

$$z_i = \mu_i, \tag{44}$$

which makes the reconstruction deterministic for a given feature of CAV_i and ego context. The trained VAE is subsequently fixed and used by the defense without online adaptation.

A.5 Additional Results

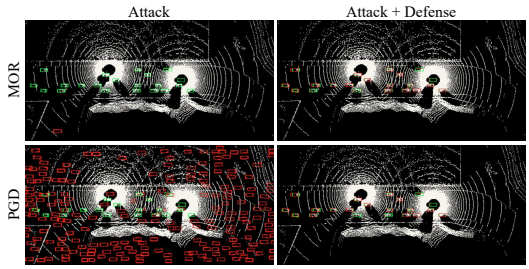

Figure 6: Visualization of perception results under MOR and PGD attacks. The left column shows the results under attack, while the right column shows the results after applying our defense. Red boxes denote model predictions, and green boxes denote ground-truth objects.

Visualization Analysis. Figure 6 shows the perception results before and after applying Gecko under MOR and PGD. MOR suppresses object detections and leads to missed objects, whereas PGD introduces a large number of false positives across the scene. After applying Gecko, the missed detections under MOR are largely recovered and the spurious predictions caused by PGD are substantially removed. Figure 7 further illustrates the underlying feature repair process. The repair-source maps show that Gecko selects different repair sources according to the available collaborative evidence, while the repair-magnitude maps indicate where corrupted features are modified. The repaired features subsequently recover suppressed detections under MOR and eliminate dense false positives under PGD. These results show that Gecko can effectively repair corrupted collaborative features under different attack patterns while preserving useful collaborative information.

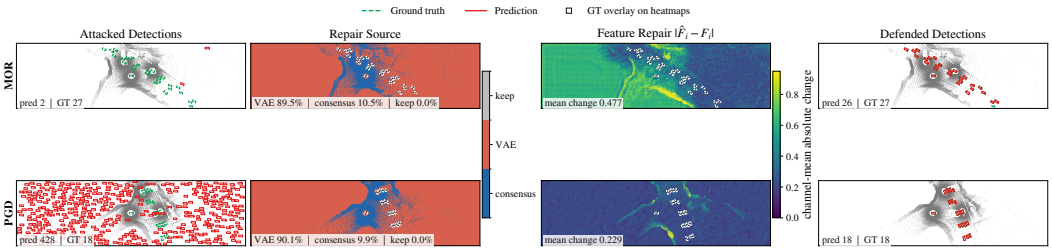

Figure 7: Qualitative results of Gecko under MOR and PGD attacks. From left to right, each row shows the attacked detections, repair source, magnitude of feature repair, and defended detections. Green dashed and red solid boxes denote ground truth and predictions, respectively, while white dashed boxes mark ground-truth locations on the repair maps. Gecko restores detections suppressed by MOR and removes the dense false positives introduced by PGD.

Impact on Clean Perception. Table 4 evaluates whether Gecko affects normal CP in the absence of attacks. Compared with the clean upper bound, Gecko introduces only a marginal decrease in AP, from 0.91/0.83 to 0.90/0.82 at AP@0.5/0.7. This small gap indicates that the defense largely preserves benign collaborative information and does not noticeably interfere with normal perception. Overall, Gecko provides effective protection while maintaining nearly the same perception performance in attack-free scenarios.

Table 4: Clean-scene performance of Gecko on the full OPV2V test set. Upper Bound denotes clean CP without defense.

Setting	AP@0.5	AP@0.7
Upper Bound	0.91	0.83
Gecko (Ours)	0.90	0.82

Effect of the Number of Collaborators. Table 5 investigates whether feature repair provides an advantage over rejecting the compromised CAV as collaborative redundancy varies. When only one collaborator is available, Oracle Reject discards the entire contribution of that CAV and therefore suffers a substantial performance loss, whereas Gecko recovers performance close to the clean upper bound by repairing corrupted features and retaining useful observations. As additional collaborators become available, this advantage gradually diminishes because other benign CAVs can increasingly compensate for the information lost by rejection. With three collaborators, Oracle Reject becomes comparable to Gecko, indicating that discarding one CAV is much less costly when sufficient redundant observations are available. These results show that feature repair is particularly beneficial in collaboration-limited scenarios, where each CAV may provide useful information that cannot be readily replaced by other participants.

Effect of the Collaborative Support Threshold τ_{sup}. Table 6 evaluates the effect of the collaborative support threshold τ_{sup}, which determines whether a region receives sufficient support from other CAVs and is therefore treated as a supported region rather than a unique-view region. Gecko remains highly stable as τ_{sup} varies from 0.50 to 0.90, with almost unchanged AP under both No Attack and MOR and an MOR ORR of approximately 2.2–2.3%. When τ_{sup} increases to 1.00, AP slightly improves, but the ORR rises to 4.0%. This indicates that an overly strict support threshold

Table 5: Performance under different numbers of collaborating CAVs. The number of collaborators includes the malicious CAV. Upper Bound denotes clean CP with no attack and no defense, while Oracle Reject denotes perfect rejection of the malicious CAV.

# Collaborators	Method	AP@0.5↑	AP@0.7↑
1	Upper Bound	0.80	0.69
	No Defense	0.00	0.00
	Oracle Reject	0.68	0.51
	Gecko	0.80	0.67
2	Upper Bound	0.85	0.76
	No Defense	0.01	0.01
	Oracle Reject	0.83	0.71
	Gecko	0.84	0.74
3	Upper Bound	0.87	0.79
	No Defense	0.03	0.02
	Oracle Reject	0.86	0.77
	Gecko	0.86	0.77

reduces the availability of collaborative evidence for feature repair, thereby weakening the defense against object removal. Overall, Gecko is robust to a broad range of τ_{sup}, while moderate values provide a better balance between preserving perception performance and enabling effective repair.

Table 6: Effect of τ_{sup} on no attack and MOR performance.

Attack	No Attack	MOR	
τ_{sup}	AP@0.5↑	AP@0.5↑	ORR↓
0.50	0.900	0.903	0.023
0.70	0.900	0.903	0.023
0.90	0.901	0.904	0.022
1.00	0.908	0.908	0.040

Effect of the Reconstruction Anomaly Threshold τ_{anom}. Table 7 evaluates the effect of the reconstruction anomaly threshold τ_{anom}, which determines how strongly reconstruction errors contribute to the anomaly evidence used for corrupted-feature localization and repair. The results remain stable when τ_{anom} varies from 0.05 to 0.15, with nearly unchanged performance under No Attack and consistently low ORR under MOR. When τ_{anom} increases to 0.20, however, MOR AP@0.5 decreases to 0.894 and the ORR rises to 3.7%. This suggests that an overly large threshold makes the anomaly evidence less responsive to corrupted features, leaving more harmful regions insufficiently repaired. Overall, Gecko is robust to moderate variations in τ_{anom}, while excessively large values can weaken its repair effectiveness.

Table 7: Effect of τ_{anom} on defense performance.

Attack	No Attack	MOR	
τ_{anom}	AP@0.5↑	AP@0.5↑	ORR↓
0.05	0.901	0.903	0.023
0.10	0.900	0.903	0.023
0.15	0.901	0.906	0.025
0.20	0.901	0.894	0.037

Robustness to Colluding Attackers. Table 8 evaluates Gecko against multi-attacker TOR (Random), where two or three malicious CAVs either optimize their perturbations independently or jointly toward the same target. Without defense, both attack strategies achieve high ASR, and joint optimization with three attackers further increases the ASR to 75.25%. In contrast, Gecko consistently reduces the ASR to below 4% across all settings while maintaining high perception accuracy. Notably, its performance remains stable even when multiple malicious CAVs jointly op-

Table 8: Defense performance against multi-attacker TOR (Random) on five-CAV OPV2V frames.

Number of Attackers	Optimization	Defense	ASR↓	AP@0.5↑	AP@0.7↑
2	Independent	No Defense	66.34%	0.9258	0.9099
		Gecko	**3.96%**	**0.9523**	**0.9100**
	Joint	No Defense	58.42%	0.9235	**0.9113**
		Gecko	**3.96%**	**0.9508**	0.9073
3	Independent	No Defense	68.32%	0.9314	0.9054
		Gecko	**1.98%**	**0.9600**	**0.9140**
	Joint	No Defense	75.25%	0.9433	**0.9315**
		Gecko	**1.98%**	**0.9596**	0.9108

Table 9: OPV2V detection performance under benign pose misalignment. Noise is independently applied to the position and yaw of non-ego CAVs. Here, σ_{xy} denotes the standard deviation of translational noise in meters, and σ_ψ denotes the standard deviation of yaw noise in degrees. Non-zero noise results are averaged over three seeds. Δ denotes Gecko minus Benign CP.

σ_{xy} / σ_ψ	Benign CP		Gecko		Δ (95% CI)	
	AP@0.5	AP@0.7	AP@0.5	AP@0.7	AP@0.5	AP@0.7
0.0 m / 0.0°	**0.91**	**0.83**	0.90	0.82	−0.012 [−0.014, −0.010]	−0.017 [−0.020, −0.014]
0.2 m / 0.2°	**0.90**	0.73	0.89	0.73	−0.012 [−0.014, −0.010]	0.000 [−0.004, +0.003]
0.4 m / 0.4°	0.83	0.52	0.83	**0.54**	+0.002 [−0.002, +0.005]	+0.025 [+0.020, +0.029]
0.6 m / 0.6°	0.71	0.40	**0.73**	**0.44**	+0.020 [+0.015, +0.024]	+0.035 [+0.031, +0.040]
1.0 m / 1.0°	0.55	0.32	**0.60**	**0.36**	+0.044 [+0.040, +0.048]	+0.042 [+0.038, +0.046]

The zero-noise condition is evaluated once; all other conditions are averaged over three seeds. Confidence intervals are obtained using 2,000 paired bootstrap replicates. Bold indicates a significantly higher result when the corresponding confidence interval excludes zero.

timize against the same target. These results show that Gecko remains effective against collusive attacks in which multiple malicious CAVs coordinate their perturbations toward a shared objective.

Robustness to Benign Pose Misalignment. Table 9 evaluates Gecko under benign pose misalignment, where σ_{xy} and σ_ψ denote the standard deviations of the translational and yaw noise applied to non-ego CAVs, respectively. Under clean and mild-noise conditions, Gecko incurs only a small performance decrease compared with benign CP. As the pose noise increases, this gap gradually disappears, and Gecko begins to outperform benign CP, with increasingly clear gains under moderate and large misalignment. This suggests that the feature repair mechanism can partially mitigate feature inconsistencies introduced by benign spatial misalignment rather than treating them solely as adversarial corruption. Nevertheless, the absolute AP of both settings decreases as the noise becomes more severe, indicating that Gecko improves robustness to pose misalignment but does not eliminate its impact.

Table 10: OPV2V detection performance under benign communication delays. The delay-compatible evaluation set contains 1,971 frames. The realized delay of every non-ego CAV exactly matches the target delay. Δ denotes Gecko minus Benign CP.

Delay (ms)	Benign CP		Gecko		Δ (95% CI)	
	AP@0.5	AP@0.7	AP@0.5	AP@0.7	AP@0.5	AP@0.7
0	**0.91**	**0.83**	0.90	0.82	−0.012 [−0.015, −0.009]	−0.016 [−0.022, −0.011]
100	0.86	0.50	0.86	**0.52**	0.000 [−0.005, +0.004]	+0.024 [+0.018, +0.030]
200	0.59	0.38	**0.63**	**0.39**	+0.036 [+0.029, +0.043]	+0.015 [+0.010, +0.020]
300	0.48	0.35	**0.53**	**0.38**	+0.053 [+0.046, +0.060]	+0.022 [+0.017, +0.028]
500	0.43	0.34	**0.48**	**0.36**	+0.042 [+0.038, +0.047]	+0.020 [+0.016, +0.025]

Values in parentheses are frame-level paired-bootstrap 95% confidence intervals computed using 2,000 replicates. Bold indicates a significantly higher result when the corresponding confidence interval excludes zero.

Robustness to Benign Communication Delays. Table 10 evaluates Gecko under benign communication delays. With no delay, Gecko incurs a small performance cost compared with Benign CP. At 100 ms, the two methods achieve comparable AP@0.5, while Gecko provides a clear improvement in AP@0.7. As the delay further increases, Gecko consistently outperforms Benign CP at both IoU thresholds, with the largest AP@0.5 improvement observed at 300 ms. These results indicate that Gecko does not introduce additional degradation under temporal misalignment and can partially mitigate the perception loss caused by larger communication delays. Nevertheless, the absolute AP of both methods decreases as the delay increases, showing that Gecko improves robustness to benign communication delays rather than eliminating their impact.