SIPA:STEALTHY INTENT PLANNING ATTACKS ON V2X END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

In vehicle-to-everything (V2X) end-to-end driving, the ego vehicle fuses bird's-eye-view (BEV) features shared by collaborators and uses the fused representation for both cooperative perception and planning. Existing V2X attacks and defenses, however, largely focus on the perception side. In particular, perception-level defenses can flag collaborators whose transmitted features visibly corrupt detection, motivating a subtler threat: can a malicious collaborator disrupt downstream planning while keeping cooperative perception close to its clean behavior? We study this *perception-stealthy planning attack* and propose **SIPA** (**S**tealthy **I**ntent **P**lanning **A**ttack). A key challenge is that the attacker cannot access or query the victim planner and does not observe its private route intent. Our key observation is that this intent is not entirely hidden. Planning-aware V2X protocols broadcast a request map generated from provisional waypoints, whose spatial structure retains a coarse proxy of the ego vehicle's route intent. SIPA exploits this protocol-visible side channel to condition an independently trained surrogate planner, infer a counter-intent steering direction, and localize a sparse request-guided perturbation region. Rather than maximizing generic waypoint displacement, SIPA optimizes a *control-aware planning objective* over trajectory-derived quantities used by the downstream controller, while suppressing perception drift. In closed-loop CARLA evaluation, SIPA reduces the Driving Score by **19.91 points** while keeping cooperative perception close to the same-frame clean counterfactual ($\Delta\mathrm{mAP@0.5} = -0.002$). As a result, a perception-level defense (CP-Guard) flags the SIPA collaborator on 18.4% of frames, at or below its 19.2% false-alarm rate on an honest one. These results reveal that perception-level consistency alone does not guarantee downstream planning safety.

1 INTRODUCTION

End-to-end (E2E) autonomous driving couples scene understanding with downstream planning (Hu et al., 2023; Jiang et al., 2023), and vehicle-to-everything (V2X) cooperation extends this capability beyond the ego vehicle's sensing range (Wang et al., 2020; Xu et al., 2022). In cooperative E2E systems, the ego vehicle fuses bird's-eye-view (BEV) features shared by neighboring agents and uses the fused representation for both perception and waypoint-based driving decisions (Liu et al., 2025). This improves driving under occlusion and limited sensing, but also creates an attack surface beyond conventional cooperative perception: a compromised collaborator can manipulate the BEV feature it shares before the feature influences downstream perception and planning (Tu et al., 2021; Liu et al., 2025). Because the two tasks consume the same fused representation but need not respond identically to the same perturbation, a malicious feature may affect perception and planning very differently (Guo et al., 2024; Sadeghi et al., 2023).

Existing V2X attacks (Tu et al., 2021; Zhang et al., 2024) mainly target cooperative perception by fabricating objects, suppressing detections, or perturbing shared features, while corresponding defenses (Li et al., 2023; Zhao et al., 2024; Hu et al., 2025) verify collaborators primarily through perception-level consistency. Such defenses are effective when malicious behavior surfaces as corrupted perception outputs, but they do not directly verify downstream planning. Meanwhile, planner-focused attacks study direct manipulation of planner inputs or sensor inputs, including query-based and white-box optimization (Sadeghi et al., 2023; Bouzidi et al., 2026). This raises a critical question: *can a malicious collaborator substantially disrupt downstream planning while leaving per-*

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.

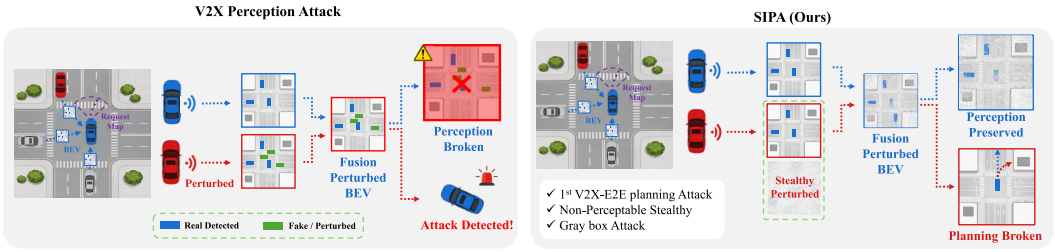


Figure 1: Overview of SIPA, which enables stealthy perturbations that preserve perception while disrupting downstream planning.

ception nearly unchanged, without access to the victim planner or its route intent? We refer to this threat as a *perception-stealthy planning attack*.

Realizing such an attack under V2X access is challenging. We consider a gray-box attacker that can reproduce the deployed perception-side computation needed to evaluate its transmitted feature, but has no access to the victim planner's weights, gradients, outputs, or private route state, and must instead rely on an independently trained surrogate planner. Without the victim's route intent, the surrogate may optimize toward a different trajectory, and the attacker does not know which regions of its shared BEV feature are relevant to the current plan. Moreover, generic waypoint displacement may translate poorly into actual driving changes because the deployed controller uses only specific steering- and speed-related quantities derived from the predicted trajectory (Liu et al., 2025). Finally, because perception and planning share the same fused feature, a perturbation strong enough to redirect the plan may also disturb detection, making selective and stealthy planning disruption non-trivial.

Our key observation is that planning-aware V2X communication unintentionally reveals part of this missing information through the normal protocol. Before the second collaboration round, the ego broadcasts a *request map*, a spatial map generated from its provisional waypoints that indicates where cooperative information is requested (Liu et al., 2025). Although the underlying waypoints and private route target are not transmitted, the request map retains the spatial footprint and directional geometry of the route. It therefore forms a protocol-visible planning side channel that reveals both where the ego requests cooperative information and the directional structure of its route intent.

Based on this observation, we propose **SIPA**, a **S**tealthy **I**ntent **P**lanning **A**ttack. SIPA uses the request map to condition an independent surrogate with a route-intent proxy, infer a counter-intent steering direction, and confine perturbations to a sparse request-guided region. Instead of maximizing generic waypoint displacement, SIPA optimizes a *control-aware planning objective* over steering- and desired-speed quantities that influence downstream vehicle control. It further filters updates that increase spurious low-confidence detection responses, limiting perception drift during attack optimization. In closed-loop CARLA (Dosovitskiy et al., 2017) evaluation, SIPA substantially degrades closed-loop driving while keeping cooperative perception close to the clean baseline, exposing a structural gap between perception-level consistency and downstream planning safety. Our contributions are summarized below:

- **Perception-stealthy attack surface and intent leakage.** We identify a V2X E2E attack surface in which a single malicious collaborator can substantially disrupt downstream planning while cooperative perception remains close to clean behavior, and we trace it to a specific cause: planning-aware request maps, introduced for communication efficiency, leak a coarse proxy of the ego vehicle's route intent through a protocol-visible side channel.
- **Intent-guided gray-box feature attack.** We propose SIPA, which recovers this route intent without any access to the victim planner or its private route target, and uses it to condition an independently trained surrogate planner, localize a sparse request-guided perturbation region, and optimize a control-aware objective over the trajectory-derived quantities the deployed controller actually reads.
- **Stealthy closed-loop disruption.** In closed-loop CARLA, SIPA substantially degrades downstream planning and closed-loop driving while empirically preserving cooperative

perception, and remains flagged by a representative perception-level defense no more often than an honest collaborator.

2 Related Work

2.1 End-to-End and Cooperative Driving

End-to-end driving maps sensor observations directly to planning or control outputs (Pomerleau, 1988; Bojarski et al., 2016). Recent full-stack systems further couple intermediate scene representations with downstream prediction and planning (Prakash et al., 2021; Chen & Krähenbühl, 2022; Shao et al., 2023; Wu et al., 2022; Hu et al., 2023; Jiang et al., 2023). In parallel, V2X collaboration shares raw observations or intermediate bird's-eye-view (BEV) features to overcome occlusion and limited sensing range (Wang et al., 2020; Li et al., 2021; Xu et al., 2022; 2023; Lu et al., 2024). More recent cooperative driving frameworks integrate these paradigms by feeding shared representations into downstream driving decisions, with some jointly optimizing perception and planning (Cui et al., 2022; Yu et al., 2025; Liu et al., 2025; Huang et al., 2025). This coupling allows externally supplied features to directly influence the ego planner, creating an attack surface that extends beyond conventional cooperative perception.

2.2 Request-Aware Communication

Communication-efficient V2X systems have evolved from deciding *who* and *when* to communicate to selectively determining *where* and *what* to transmit (Liu et al., 2020b;a; Hu et al., 2022; Yang et al., 2023; Hu et al., 2026; 2024). The semantics of these spatial requests depend on the system design. Perception-aware methods select regions based on perceptual confidence (Hu et al., 2022), whereas planning-aware systems derive request maps from predicted waypoints, as instantiated by CoDriving (Liu et al., 2025). Although these maps do not explicitly disclose the exact future trajectory, their spatial footprints reveal regions aligned with the ego vehicle's planned route. Prior work (Liu et al., 2025; Huang et al., 2025) treats this protocol signal primarily as a mechanism for improving communication efficiency. We instead identify it as a protocol-visible intent side channel whose spatial structure can reveal coarse planning intent and guide an adversary toward planning-critical regions.

2.3 Attacks on Driving and Cooperative Perception

Adversarial attacks on autonomous driving have explored multiple attack surfaces, including perturbations to ego-side sensor inputs (Deng et al., 2020; Wu et al., 2023; Sato et al., 2021; Boloor et al., 2020; Sahu et al., 2026), adversarial trajectories of surrounding vehicles (Zhang et al., 2022), and perturbations to internal module representations (Wang et al., 2024).

Several of these studies demonstrate that perturbations to visual inputs or motion prediction can lead to unsafe downstream driving behavior (Sato et al., 2021; Boloor et al., 2020; Zhang et al., 2022). A smaller body of work targets planning more explicitly, either by searching for perception outputs that cause a black-box planner to fail or by manipulating trajectory-scoring mechanisms to favor unsafe plans (Sadeghi et al., 2023; Bouzidi et al., 2026). These methods, however, assume repeated query access to the victim planner or direct control over its inputs, capabilities that are not available to an ordinary V2X collaborator (Sadeghi et al., 2023; Bouzidi et al., 2026). In V2X systems, prior attacks instead operate through the collaborative communication channel, perturbing shared features, spoofing agent poses, or fabricating objects (Tu et al., 2021; Li et al., 2024; Zhang et al., 2024; Zhang & Mao, 2024; Wang et al., 2025; Lin et al., 2025). These attacks manipulate cooperative perception through fabricated objects, pose changes, or altered shared representations. Some explicitly pursue unsafe downstream behavior, including through subtle object-pose perturbations that propagate through tracking and prediction (Zhang et al., 2024; Zhang & Mao, 2024; Zhang et al., 2026). Existing defenses assess collaborators through perception-level consistency and verification mechanisms (Li et al., 2023; Zhao et al., 2024; Hu et al., 2025). Selective degradation of one task while preserving other tasks has also been studied in multi-task adversarial attacks (Guo et al., 2024).

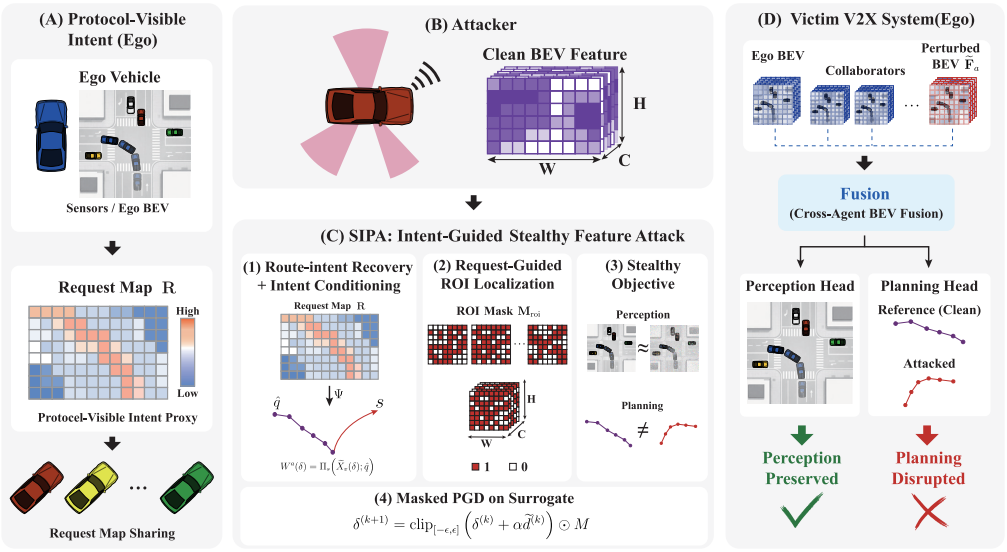


Figure 2: **SIPA pipeline. (A) Protocol-visible intent:** the ego vehicle renders its provisional trajectory into a request map R and broadcasts it to collaborators, exposing a coarse spatial proxy of its route intent. **(B) Attacker:** a single compromised non-ego collaborator modifies only its transmitted BEV feature. **(C) SIPA attack generation: (C-1)** route-intent recovery, intent conditioning, and counter-intent estimation; **(C-2)** request-guided ROI localization; **(C-3)** control-aware counter-intent optimization; and **(C-4)** perception-aware masked PGD. **(D) Victim V2X system:** the victim fuses the perturbed collaborator feature with benign V2X features, resulting in downstream planning disruption while cooperative perception remains close to its clean behavior.

3 METHOD

SIPA addresses a different attack surface than previous attacks on cooperative driving. Rather than corrupting cooperative perception to indirectly affect driving, SIPA perturbs the BEV feature shared by a malicious collaborator to selectively disrupt downstream planning while preserving cooperative perception. It further operates without querying or directly manipulating the victim planner, using a route-intent proxy recovered from the protocol-visible request map under the perception-model and runtime-input access assumptions specified in Appendix A.7.

Figure 2 summarizes the full SIPA pipeline. Panel (A) shows the protocol-visible request map that exposes coarse route intent, while Panel (B) identifies the compromised collaborator and the feature-level attack interface. Panel (C) presents the four components of SIPA attack generation: route-intent recovery and intent conditioning (C-1), request-guided spatial localization (C-2), control-aware counter-intent optimization (C-3), and perception-aware masked optimization (C-4). Panel (D) shows how the perturbed collaborator feature enters the victim V2X pipeline and affects its downstream perception and planning. The following subsections describe these components in detail.

3.1 THREAT MODEL

System. Consider a V2X system with N agents, where agent 0 is the ego vehicle and agents $1, \ldots, N-1$ are collaborators. Each agent produces a BEV feature $X_i \in \mathbb{R}^{C \times H \times W}$. After spatial alignment, the ego vehicle fuses its own feature with the received collaborator features,

$$\widetilde{X}_v = F\big(X_0, \{X_j\}_{j=1}^{N-1}\big)\,, \tag{1}$$

and uses the fused representation for both perception and planning,

$$Y = D(\widetilde{X}_v), \qquad W = \Pi(\widetilde{X}_v) = [w_1, \ldots, w_T]. \tag{2}$$

Here, D and Π denote the victim perception and planning models, respectively. For simplicity, $\Pi(\widetilde{X}_v)$ suppresses additional temporal and map-related planning inputs to the planner, which are detailed in Appendix A.7.

Attacker. The attacker compromises a single non-ego collaborator a and modifies only its transmitted feature,

$$\widetilde{X}_v(\delta) = F(X_0, \{X_j\}_{j\neq 0,a}, X_a + M \odot \delta), \qquad \|\delta\|_\infty \leq \epsilon, \tag{3}$$

where M is a spatial attack mask and ϵ is the perturbation budget.

We consider a gray-box setting with white-box access to the deployed perception stack but no access to the victim planner's parameters, gradients, predicted waypoints, or private route target. This setting is motivated by homogeneous cooperative-perception deployments in which participating agents use a common perception implementation. SIPA instead obtains planning gradients from an independently trained and frozen surrogate planner Π_s, conditioned on the broadcast request map. Our evaluation additionally assumes access to the actual ego-side perception inputs and the contextual tensors required to run the surrogate planner and the perception filter, not the victim planner, which remains inaccessible throughout. Detailed information is given at Appendix A.7.

The desired attack behavior is

$$\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{J}_{\text{plan}}(\delta), \qquad D(\widetilde{X}_v(\delta)) \approx D(\widetilde{X}_v(0)). \tag{4}$$

i.e., disrupting downstream planning while preserving perception. Because the victim planner is inaccessible, this objective cannot be optimized directly. SIPA therefore approximates it using an independently trained surrogate planner and protocol-visible information. The exact feature-injection point, planner inputs suppressed by the notation above, and the attacker access boundary are detailed in Appendix A.7.

3.2 Request Map as a Protocol-Visible Intent Side Channel

As shown in Figure 2(A), the ego vehicle predicts a provisional waypoint trajectory before the second collaboration round and renders it into a spatial request map R. The map specifies where cooperative information is requested and is broadcast to legitimate collaborators through the normal V2X protocol. The underlying waypoints and private route target are not transmitted.

Although introduced for communication efficiency, R retains the spatial structure of the ego vehicle's intended route. SIPA exploits this signal in three ways:

$$R \longrightarrow \underbrace{\hat{q}}_{\text{intent conditioning}}, \qquad \underbrace{s}_{\text{counter-intent direction}}, \qquad \underbrace{M}_{\text{request-guided region}}. \tag{5}$$

Thus, the request map provides both a protocol-visible route-intent cue and a spatial prior for the attack. The underlying two-round request-aware communication protocol and request-map usage are detailed in Appendix A.1.

3.3 Intent Recovery and Conditioning

Route-intent recovery. Figure 2(C-1) shows the intent-recovery stage. Because R is rendered from provisional waypoints, its spatial structure provides a geometric trace of the route. SIPA applies a recovery operator Ψ,

$$W^f = \Psi(R) = [w_1^f, \ldots, w_T^f], \qquad \hat{q} = w_T^f, \tag{6}$$

where W^f is a geometric route proxy and $\hat{q}$ is its endpoint.

Importantly, W^f does not reconstruct the original temporal waypoint sequence or its speed profile. The exact recovery procedure and failure handling are provided in Appendix A.3.

Intent conditioning. The recovered endpoint conditions the independently trained surrogate planner:

$$W^c = \Pi_s(\widetilde{X}_v(0); \hat{q}), \qquad W^a(\delta) = \Pi_s(\widetilde{X}_v(\delta); \hat{q}). \tag{7}$$

Using the same $\hat{q}$ for both predictions isolates the planning change induced by the malicious feature from changes in the surrogate's route condition.

Counter-intent direction. SIPA also extracts the directional sign of the recovered route and defines

$$s = -\operatorname{sign}\big(\operatorname{angle}(W^f)\big), \tag{8}$$

so that the attack favors a steering-related trajectory change opposite to the ego vehicle's inferred route intent. If route recovery fails or the recovered intent is approximately straight, SIPA falls back to an untargeted objective. The exact direction estimator and fallback rules are given in Appendix A.3.

3.4 Intent-Guided Perception-Preserving Feature Attack

Request-guided localization. Figure 2(C-2) shows the spatial attack region. Let M_R denote the binary request support mapped to the attack feature grid. SIPA restricts this support to locations where the compromised collaborator has valid nonzero feature support:

$$M = M_R \odot \mathbf{1}\left[\sum_c |X_{a,c}| > 0\right]. \tag{9}$$

The exact normalization, thresholding, resizing, and dilation used to construct M_R are detailed in Appendix A.2.

Control-aware counter-intent objective. Figure 2(C-3) illustrates the planning objective. Generic waypoint-distance losses may emphasize trajectory changes that have little effect on downstream driving. SIPA instead operates on functions of the predicted trajectory that are used by the deployed waypoint-to-control mapping.

Let

$$a(W) = f_{\text{steer}}(W), \qquad v(W) = f_{\text{speed}}(W) \tag{10}$$

denote the steering-related and desired-speed quantities derived from a predicted waypoint sequence. In the deployed system, these functions depend only on a small subset of the predicted waypoints; their exact form and derivation are given in Appendix A.4.

For clean and attacked surrogate predictions, define

$$\Delta a = a(W^a) - a(W^c), \qquad \Delta v = v(W^a) - v(W^c). \tag{11}$$

Without a directional cue, the corresponding untargeted objective is

$$\mathcal{L}_{\text{ctrl}} = |\Delta a| + \lambda \frac{|\Delta v|}{v_{\max}}. \tag{12}$$

The final SIPA objective incorporates the counter-intent direction recovered from the request map:

$$\mathcal{L}_{\text{SIPA}} = s\,\Delta a - \lambda \frac{\Delta v}{v_{\max}}. \tag{13}$$

The first term rewards a steering-related trajectory change opposite to the recovered route intent, while the second rewards a reduction in the trajectory-derived desired-speed quantity. SIPA therefore optimizes control-relevant functions of the predicted trajectory without modifying the downstream controller itself.

Perception-preserving masked optimization. Figure 2(C-4) shows the optimization stage. Let p^c and p^a denote clean and attacked classification probabilities. SIPA measures spurious confidence increases at locations that are low-confidence under the clean prediction using

$$\mathcal{L}_{\text{FP}} = \left\langle \left(\operatorname{ReLU}(p^a - p^c) \odot \mathbf{1}[p^c \le \tau_p]\right)^2 \right\rangle. \tag{14}$$

Let $d^{(k)}$ denote the planning-ascent direction at iteration k. SIPA filters coordinates whose first-order effect would increase $\mathcal{L}_{\text{FP}}$,

$$\widetilde{d}^{(k)} = d^{(k)} \odot \mathbf{1}\left[d^{(k)} \odot \nabla_\delta \mathcal{L}_{\text{FP}} \le 0\right], \tag{15}$$

and projects the resulting update onto both the perturbation budget and the request-guided spatial support,

$$\delta^{(k+1)} = \mathrm{clip}_{[-\epsilon,\epsilon]}\left(\delta^{(k)} + \alpha\widetilde{d}^{(k)}\right) \odot M. \tag{16}$$

This procedure favors counter-intent planning disruption while suppressing spurious perception responses and restricting all feature modification to the request-guided region. The construction of $d^{(k)}$, momentum, initialization, warm starting, and optimization hyperparameters are deferred to Appendix A.6.

4 Experimental Setup

4.1 Environment and Models

We evaluate SIPA in closed-loop CARLA (Dosovitskiy et al., 2017) using a CoDriving-based V2X-E2E stack (Liu et al., 2025) on eight fixed Town05 routes. All methods use the same victim–surrogate pair, compromised collaborator, route set, and perturbation budget. Following the gray-box threat model in Section 3, the attacker has access to the deployed perception-side computation but not to the victim planner, and instead uses an independently trained frozen surrogate planner. Unless stated otherwise, we use $\epsilon = 6$ and $K = 60$ attack iterations.

We compare against PGD*, SOMBRA (Wang et al., 2025), and SOMBRA*. A superscript $*$ denotes a baseline whose attack objective is replaced by our control-aware planning objective while retaining its original spatial mechanism. Clean denotes the unperturbed reference.

The closed-loop evaluation protocol and shared evaluation settings are detailed in Appendix B.1, and baseline implementations are specified in Appendix B.2. A compact summary of the evaluation configuration is provided in Appendix B.6.

4.2 Evaluation Metrics

Perception is evaluated using $\Delta\mathrm{mAP@0.5}$ and detection precision, while closed-loop driving is evaluated using the change in Driving Score ($\Delta\mathrm{DS}$) and collisions per kilometer (Coll./km).

We additionally evaluate all methods against CP-Guard (Hu et al., 2025), a perception-level V2X defense that verifies collaborators through consistency with the ego vehicle's perception, and report the collaborator rejection rate as *Detection*.

Metric definitions and aggregation rules are detailed in Appendix B.3, and the CP-Guard implementation and threshold calibration are provided in Appendix B.4.

4.3 Ablation Studies

We evaluate three components of SIPA. For spatial targeting, we compare the request-guided ROI with a matched-size random ROI. We further remove intent conditioning or perception-aware filtering to evaluate their individual contributions. The detailed explanation of each ablation variant is provided in Appendix B.5.

5 Results

5.1 Main Attack Comparison

Table 1 compares SIPA with feature-level V2X attack baselines under the same gray-box evaluation setting. The results reveal a clear distinction between maximizing raw closed-loop degradation and achieving planning disruption while preserving perception.

The planning-oriented baselines PGD* and SOMBRA* produce the largest reductions in Driving Score, with $\Delta\mathrm{DS} = -41.39$ and -45.98, respectively. However, their stronger driving disruption is accompanied by severe perception degradation. PGD* decreases mAP@0.5 by 0.294 and reduces detection precision to 15.91, while SOMBRA* decreases mAP@0.5 by 0.282 with a precision of

Table 1: Main attack comparison on the CoDriving-based V2X-E2E model under gray-box checkpoint transfer. Perception is evaluated by ΔmAP@0.5 and detection precision, and closed-loop driving by ΔDS and collisions per kilometer. Detection denotes the CP-Guard collaborator rejection rate; for Clean, it is the false-alarm rate. Clean serves as the reference for the relative metrics, so the corresponding entries are omitted.

Method	Perception		Planning		Detection↓
	$\Delta\mathrm{mAP@0.5}\uparrow$	Precision ↑	ΔDS ↓	Coll./km ↑	
Clean	–	–	–	0.94	19.2
PGD*	-0.294	15.91	-41.39	12.10	71.3
SOMBRA	-0.001	97.23	-1.79	1.40	17.8
SOMBRA*	-0.282	28.00	-45.98	16.52	61.4
SIPA (Ours)	-0.002	96.19	-19.91	9.04	18.4

Table 2: Spatial-targeting ablation. The request-map ROI is compared with a matched-size random ROI over the valid collaborator feature support.

Spatial targeting	$\Delta\mathrm{mAP@0.5}$	ΔDS ↓
Random ROI	-0.001	-11.20
Request-map ROI (Ours)	-0.002	-19.91

28.00. Their collision rates also increase to 12.10 and 16.52 collisions/km, respectively, compared with 0.94 collisions/km for Clean.

In contrast, the original SOMBRA largely preserves perception ($\Delta\mathrm{mAP@0.5} = -0.001$, precision $= 97.23$), but its downstream effect is substantially weaker, with $\Delta\mathrm{DS} = -1.79$. This illustrates the limitation of a perception-oriented attack objective for directly disrupting downstream driving.

SIPA achieves a different attack regime. It reduces Driving Score by 19.91 points while keeping perception close to the clean counterfactual, with $\Delta\mathrm{mAP@0.5} = -0.002$ and a precision of 96.19. Its collision rate increases from 0.94 to 9.04 collisions/km. Thus, although unconstrained planning-oriented attacks achieve larger raw closed-loop degradation, they do so together with substantial perception corruption. SIPA instead induces considerable downstream driving degradation while maintaining perception behavior close to the clean baseline, matching the perception-stealthy attack objective of this work.

Detection under a perception-level defense. The last column of Table 1 reports how often CP-Guard rejects the collaborator, and the ordering follows perception degradation rather than driving degradation. PGD* is rejected on 71.3% of verified frames and SOMBRA* on 61.4%, against a clean false-alarm rate of 19.2%: both are flagged more than three times as often as an honest collaborator. This is the expected consequence of their perception damage, since CP-Guard scores a collaborator by the consistency of the fused detections with the ego vehicle's own. SIPA, by contrast, is rejected on 18.4% of verified frames — at or below the clean false-alarm rate — even though it removes 19.91 points of Driving Score. All three attacks optimize control-aware objectives built from the same trajectory-derived quantities and share the same perturbation budget, surrogate planner, and iteration count; they differ in spatial targeting, in the perception-aware filter, in an additional fusion-attention term for SOMBRA*, and in whether the steering term carries a recovered directional sign. The detection gap therefore cannot be attributed to the choice of control quantities, although it is not isolated to the spatial mechanism alone.

5.2 Effect of Request-Guided Spatial Localization

Table 2 evaluates the contribution of request-guided spatial targeting. We compare the proposed request-map ROI with a matched-size random ROI over the valid collaborator feature support.

The random ROI produces only a small perception change ($\Delta\mathrm{mAP@0.5} = -0.001$), but reduces Driving Score by 11.20 points. Under the same per-frame perturbation-support size, the request-guided ROI increases the degradation to 19.91 points while maintaining a similarly small perception

Table 3: Ablation of SIPA components. We remove intent conditioning or perception-aware coordinate filtering to evaluate their contributions to closed-loop disruption and perception preservation.

Method	$\Delta\mathrm{mAP@0.5}$	$\Delta\mathrm{DS}\downarrow$
w/o intent conditioning	-0.010	-5.94
w/o perception filtering	-0.258	-41.41
Ours	-0.002	-19.91

change ($\Delta\mathrm{mAP@0.5} = -0.002$). The 8.71-point difference in Driving Score shows that the effectiveness of SIPA cannot be explained by sparse perturbation alone; the spatial locations derived from the request map are important for downstream disruption.

These results indicate that the request map provides an informative spatial prior that concentrates perturbations on regions relevant to the victim's current driving intent.

5.3 Ablation of SIPA Components

Table 3 evaluates the contributions of intent conditioning and perception-aware coordinate filtering.

Removing intent conditioning substantially reduces attack effectiveness. Without intent conditioning, the Driving Score decreases by only 5.94 points, compared with 19.91 points for full SIPA. The perception deviation also increases from $\Delta\mathrm{mAP@0.5} = -0.002$ to -0.010. This result supports the role of the protocol-visible request map in conditioning the surrogate toward a more effective downstream attack direction.

Removing perception-aware filtering produces the opposite behavior. Without the filter, the attack becomes considerably stronger in terms of closed-loop degradation, reaching $\Delta\mathrm{DS} = -41.41$, but the perception change simultaneously increases to $\Delta\mathrm{mAP@0.5} = -0.258$. The additional driving degradation therefore comes at the cost of losing the perception stealth targeted by SIPA.

The full SIPA configuration combines these complementary effects: intent conditioning provides planning-side effectiveness, while perception-aware filtering limits visible perception degradation. As a result, SIPA achieves substantial closed-loop disruption ($\Delta\mathrm{DS} = -19.91$) while keeping perception close to the clean counterfactual ($\Delta\mathrm{mAP@0.5} = -0.002$).

6 Conclusion

We presented SIPA, a stealthy intent-guided planning attack against V2X end-to-end autonomous driving. Operating in a gray-box setting in which the victim planner is inaccessible, SIPA compromises a single non-ego agent to perturb its transmitted BEV feature, leaving the ego vehicle's sensors untouched. By exploiting the cooperative request map as a protocol-visible intent proxy, SIPA conditions an independently trained surrogate planner on a recovered route intent, infers a counter-intent steering direction, and confines the perturbation to the requested region. It then optimizes a control-aware planning objective over the quantities the deployed controller reads, while a gradient filter suppresses updates that would raise spurious detections. The resulting attack substantially degrades closed-loop driving yet is rejected by a perception-level consensus defense no more often than an honest collaborator. Our findings expose a critical vulnerability: robust cooperative perception does not guarantee planning safety. We expect SIPA to stimulate the development of comprehensive V2X defenses that ensure end-to-end consistency across feature fusion, perception, and closed-loop driving behavior.

AI USE STATEMENT

We used generative AI tools for two purposes only: editing our text for readability and searching for related work. We did not use these tools to develop research ideas, design experiments, implement methods, interpret results, create synthetic datasets, or draft new text for any part of the paper. All sections were written by the authors, and the AI-assisted edits were limited to grammar, word choice, and consistent terminology in our own text. Before incorporating any edited sentence, we checked it against our derivations and experimental results. For every paper found with the help of these tools, we read the paper ourselves and confirmed that it supports the specific claim for which it is cited. The authors take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work identifies a vulnerability in planning-aware V2X end-to-end driving: a malicious collaborator can influence the ego vehicle's planning while keeping its perception close to the clean baseline. Our goal is to inform the design of safer V2X protocols and defenses, not to enable attacks on deployed vehicles. All experiments were conducted in the CARLA simulator. We did not test the attack on real vehicles, public roads, or any deployed V2X system, and the study involved no human subjects. The attack requires the adversary to participate in the V2X network as a collaborator and to reproduce the deployed perception-side computation. We disclose this threat because it shows that perception-level consistency, on which existing V2X defenses largely rely, does not by itself guarantee planning safety. Our findings point to two mitigation directions: (i) limiting the route intent exposed through the request map, and (ii) checking consistency at the planning level rather than only at the perception level. To reduce the risk of misuse, we will release the code to researchers upon request under a research-only license.

REPRODUCIBILITY STATEMENT

We provide detailed implementation and evaluation specifications in the appendices. Appendix A describes the complete SIPA attack pipeline, including its implementation details, optimization procedure, hyperparameters, and attacker access assumptions. Appendix B documents the closed-loop evaluation protocol, baseline implementations, evaluation metrics and aggregation rules, defense configuration, and ablation settings. Together, these details specify the experimental setup used to obtain all results reported in this paper.

REFERENCES

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Adith Boloor, Karthik Garimella, Xin He, Christopher Gill, Yevgeniy Vorobeychik, and Xuan Zhang. Attacking vision-based perception in end-to-end autonomous driving models. *Journal of Systems Architecture*, 110:101766, 2020. doi: 10.1016/j.sysarc.2020.101766.

Halima Bouzidi, Mboutidem Ekemini Mkpong, Haoyu Liu, and Mohammad Abdullah Al Faruque. Off the rails: Hijacking the scoring head in generative end-to-end driving planners with safety-violating adversarial perturbations. *arXiv preprint arXiv:2606.30807*, 2026.

Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17222–17231, 2022.

Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. COOPERNAUT: End-to-end driving with cooperative perception for networked vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17252–17262, 2022.

Yao Deng, Xi Zheng, Tianyi Zhang, Chen Chen, Guannan Lou, and Miryung Kim. An analysis of adversarial attacks and defenses on autonomous driving models. In *IEEE International*

Conference on Pervasive Computing and Communications (PerCom), pp. 1–10, 2020. doi: 10.1109/PerCom45495.2020.9127389.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16. PMLR, 2017. URL https://proceedings.mlr.press/v78/dosovitskiy17a.html.

Jiacheng Guo, Tianyun Zhang, Lei Li, Haochen Yang, Hongkai Yu, and Minghai Qin. Stealthy multi-task adversarial attacks. *arXiv preprint arXiv:2411.17936*, 2024.

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*, volume 39, pp. 23203–23211, 2025. doi: 10.1609/aaai.v39i22.34486.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862, 2023.

Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4874–4886, 2022.

Yue Hu, Juntong Peng, Sifei Liu, Junhao Ge, Si Liu, and Siheng Chen. Communication-efficient collaborative perception via information filling with codebook. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15481–15490, 2024.

Yue Hu, Xianghe Pang, Xiaoqi Qin, Yonina C. Eldar, Siheng Chen, Ping Zhang, and Wenjun Zhang. Pragmatic communication in multi-agent collaborative perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(8):9279–9296, 2026. doi: 10.1109/TPAMI.2026.3680062.

Jiahao Huang, Jianhang Zhu, Rongpeng Li, Zhifeng Zhao, and Honggang Zhang. Select2Drive: Pragmatic communications for real-time collaborative autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 26(12):21939–21953, 2025. doi: 10.1109/TITS.2025.3611377.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8350, 2023.

Jinlong Li, Baolu Li, Xinyu Liu, Jianwu Fang, Felix Juefei-Xu, Qing Guo, and Hongkai Yu. AdvGPS: Adversarial GPS for multi-agent perception attack. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 18421–18427, 2024. doi: 10.1109/ICRA57147.2024.10610012.

Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29541–29552, 2021.

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 (ICCV)*, pp. 186–195, 2023.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19947–19956, October 2025.

Genjia Liu, Yue Hu, Chenxin Xu, Weibo Mao, Junhao Ge, Zhengxiang Huang, Yifan Lu, Yinda Xu, Junkai Xia, Yafei Wang, and Siheng Chen. Toward collaborative autonomous driving: Simulation platform and end-to-end system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6566–6584, 2025. doi: 10.1109/TPAMI.2025.3560327.

Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4106–4115, 2020a.

Yen-Cheng Liu, Junjiao Tian, Chih-Yao Ma, Nathan Glaser, Chia-Wen Kuo, and Zsolt Kira. Who2com: Collaborative perception via learnable handshake communication. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6876–6883, 2020b.

Yifan Lu, Yue Hu, Yiqi Zhong, Dequan Wang, Yanfeng Wang, and Siheng Chen. An extensible framework for open heterogeneous collaborative perception. In *The Twelfth International Conference on Learning Representations*, 2024.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Dean A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, volume 1, pp. 305–313, 1988.

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7077–7087, 2021.

Jonathan Sadeghi, Nicholas A. Lord, John Redford, and Romain Mueller. Attacking motion planners using adversarial perception errors. *arXiv preprint arXiv:2311.12722*, 2023.

Ishan Sahu, Somnath Hazra, Somak Aditya, and Soumyajit Dey. AD^2: Analysis and detection of adversarial threats in visual perception for end-to-end autonomous driving systems. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1695–1704, March 2026.

Takami Sato, Junjie Shen, Ningfei Wang, Yunhan Jia, Xue Lin, and Qi Alfred Chen. Dirty road can attack: Security of deep learning based automated lane centering under physical-world attack. In *USENIX Security Symposium*, pp. 3309–3326, 2021.

Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Proceedings of the Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 726–737. PMLR, 2023.

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 (ICCV)*, 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 *USENIX Security Symposium*, pp. 7387–7406. USENIX Association, August 2025.

Lu Wang, Tianyuan Zhang, Yikai Han, Muyang Fang, Ting Jin, and Jiaqi Kang. Attack end-to-end autonomous driving through module-wise noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 8349–8352, 2024.

Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2VNet: Vehicle-to-vehicle communication for joint perception and prediction. In *European Conference on Computer Vision (ECCV)*, pp. 605–621. Springer, 2020.

Han Wu, Syed Yunas, Sareh Rowlands, Wenjie Ruan, and Johan Wahlström. Adversarial driving: Attacking end-to-end autonomous driving. In *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–7, 2023. doi: 10.1109/IV55152.2023.10186386.

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6119–6132, 2022.

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 (ECCV)*, pp. 107–124. Springer, 2022.

Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. CoBEVT: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Proceedings of the Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 989–1000. PMLR, 2023.

Dingkang Yang, Kun Yang, Yuzheng Wang, Jing Liu, Zhi Xu, Rongbin Yin, Peng Zhai, and Lihua Zhang. How2comm: Communication-efficient and collaboration-pragmatic multi-agent perception. In *Advances in Neural Information Processing Systems*, volume 36, pp. 25151–25164, 2023.

Haibao Yu, Wenxian Yang, Jiaru Zhong, Zhenwei Yang, Siqi Fan, Ping Luo, and Zaiqing Nie. End-to-end autonomous driving through V2X cooperation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9598–9606, 2025. doi: 10.1609/aaai.v39i9.33040.

Qingzhao Zhang and Z. Morley Mao. Stealthy data fabrication in collaborative vehicular perception. In *Proceedings of the Sixth Workshop on CPS&IoT Security and Privacy (CPSIoTSec)*, pp. 142–149, 2024.

Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, and Z. Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15159–15168, 2022.

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 *USENIX Security Symposium*, 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*, 2026.

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

A METHOD DETAILS

This section provides the implementation details deferred from Section 3. We first describe the request-aware two-round protocol (Appendix A.1), followed by the construction of the request-guided attack region (Appendix A.2), the intent-recovery operator and counter-intent rule (Appendix A.3), the waypoint-to-control mapping underlying the control-aware objective (Appendix A.4), the perception-aware gradient filter (Appendix A.5), the optimization procedure (Appendix A.6), and the precise attacker access assumed in our implementation (Appendix A.7).

A.1 REQUEST-AWARE PROTOCOL

The base V2X system executes two communication rounds per driving tick. In the first round, the ego vehicle performs inference before receiving the second-round requested collaborative features and predicts a provisional trajectory. The provisional waypoints are rendered into a request map $R \in [0,1]^{H_R \times W_R}$, which is broadcast to collaborators through the normal communication protocol. The raw waypoint tensor and the ego vehicle's private route target are not transmitted.

The deployed communication module combines a perception-derived confidence map Q with the warped request map R^{warp}. Specifically,

$$q = Q \odot \mathrm{clip}\left(\frac{5R^{\mathrm{warp}}}{\max R^{\mathrm{warp}} + 10^{-7}}, 10^{-4}, 1 - 10^{-4}\right), \tag{17}$$

after which a hard communication mask is obtained as

$$C = \mathbf{1}\left[q \geq \tau_{\mathrm{comm}}\right]. \tag{18}$$

The ego row is always retained in C. In our configuration, $\tau_{\mathrm{comm}} = 0.01$. The normalization in equation 17 is part of the deployed communication protocol and is distinct from the row-wise normalization used by SIPA to construct its attack ROI in Appendix A.2.

In the second round, the ego receives features selected by this communication mask and re-runs the cooperative perception and planning pipeline. SIPA first performs a clean requested forward pass to obtain its same-frame reference and then holds the corresponding request map fixed throughout the attack optimization for that tick. Thus, the perturbation does not alter the request signal from which its own intent and spatial support were obtained.

In the evaluated implementation, the collaborator BEV feature tensor typically has spatial resolution 192×576, whereas the request map has resolution 96×288. The exact tensor dimensions are obtained from the deployed model at run time.

A.2 REQUEST-GUIDED ROI CONSTRUCTION

Section 3 defines the attack mask as $M = M_R \odot \mathbf{1}[\sum_c |X_{a,c}| > 0]$. We now specify the construction of M_R.

Let $R^{(a)}$ denote the request map associated with the compromised collaborator after the protocol's spatial transformation. SIPA first normalizes each collaborator row by its own peak value,

$$\overline{R}^{(a)} = \frac{R^{(a)}}{\max\left(\max_{h,w} R^{(a)}_{h,w}, 10^{-6}\right)}. \tag{19}$$

The request support is thresholded at the *original request-map resolution*,

$$B_R = \mathbf{1}\left[\overline{R}^{(a)} > \tau_R\right]. \tag{20}$$

The binary mask is then resized to the feature resolution using nearest-neighbor interpolation and dilated on the feature grid:

$$M_R = \mathrm{Dilate}_d\left(\mathrm{Resize}_{\mathrm{NN}}(B_R)\right). \tag{21}$$

Thresholding precedes resizing so that interpolation does not implicitly expand the requested region.

Finally, SIPA intersects this support with the spatial support of the collaborator feature and with the selected attacker row,

$$M = M_R \odot \mathbf{1}\left[\sum_c |X_{a,c}| > 0\right] \odot \mathbf{1}[a \in \mathcal{A}], \tag{22}$$

where $\mathcal{A}$ denotes the selected non-ego attacker rows. For the single-attacker setting used in the final evaluation, $\mathcal{A} = \{a\}$, so only one non-ego collaborator row is modified. For the final SIPA configuration,

$$\tau_R = 0.1, \qquad d = 2, \tag{23}$$

where d is the configured dilation extent measured in cells of the attack feature grid. No additional dilation is applied to the feature-support term.

The ordering is therefore

row normalization → threshold → nearest-neighbor resize → dilation → feature support → attacker-row restriction.

The resulting mask represents regions requested by the victim and available through the compromised collaborator; it should not be interpreted as a formal guarantee that every selected cell is planning-critical.

The perturbation budget ϵ remains a cell-wise ℓ_∞ bound. Restricting the ROI reduces the number of perturbed cells but does not redistribute the unused budget to the remaining cells.

A.3 Intent Recovery and Counter-Intent Direction

Request-map geometry. The request map is produced by placing localized Gaussian responses around provisional waypoints and combining their spatial responses. Ignoring discretization, truncation, and warping, an idealized max-composite request map can be written as

$$R(p) \approx \exp\left(-\frac{\min_t \|p - w_t\|_2^2}{2\varsigma^2}\right). \tag{24}$$

Hence its intensity behaves as a distance field to the nearest rendered waypoint,

$$d(p) = \varsigma\sqrt{-2\log \overline{R}(p)}, \tag{25}$$

where $\overline{R}$ is peak-normalized. Equation 24 is a geometric interpretation rather than an exact inverse model of the deployed renderer: the actual map is discrete, spatially truncated, and warped before use.

Recovery operator. The implemented operator Ψ first peak-normalizes the observed request map and extracts low-distance seed regions. It begins with a distance threshold of 0.5 pixels and, when fewer than two separate seed regions are recovered, successively uses

$$\tau_s \in \{0.5,\ 1.0,\ 1.5,\ 2.5,\ 4.0,\ 6.0\}. \tag{26}$$

Connected components in the resulting seed set provide route samples. When at least two components are available, their representative locations are ordered through nearest-neighbor chaining and the resulting polyline is resampled uniformly in arc length to produce

$$W^f = \Psi(R) = [w_1^f, \ldots, w_T^f], \qquad T = 10. \tag{27}$$

If the map still collapses to a single component at the largest threshold, Ψ returns no path.

Because the final resampling is uniform in arc length, W^f recovers the *spatial route geometry*, including its left–right structure, but not the temporal identity, waypoint timing, or speed of the victim trajectory. Consequently, SIPA uses

$$\hat{q} = w_T^f \tag{28}$$

only as a route-consistent goal proxy. If recovery fails, the surrogate target reuses the most recent valid recovered goal when available and otherwise falls back to the zero target. It never substitutes the inaccessible victim target.

Counter-intent sign. To determine the counter-intent direction, SIPA uses the recovered endpoint direction

$$\phi_R = \operatorname{atan2}\left(w^f_{T,0}, -w^f_{T,1} + \epsilon_\phi\right), \tag{29}$$

where $\epsilon_\phi > 0$ is a small numerical stabilizer. SIPA then defines

$$s = -\operatorname{sign}(\phi_R). \tag{30}$$

The endpoint direction is used instead of the final recovered segment because the request-map rendering and inversion can flatten adjacent lateral coordinates, making a local last-segment direction unstable even when the overall route retains a clear left–right orientation.

We apply the counter-intent direction only when

$$|\phi_R| \geq \tau_{\text{intent}}, \qquad \tau_{\text{intent}} = 0.01. \tag{31}$$

If Ψ fails or $|\phi_R| < \tau_{\text{intent}}$, SIPA drops the directional sign and uses the untargeted control-aware objective in equation 12. The request map provides no reliable speed information, so counter-intent is used only for the steering-related term; the speed term always rewards a reduction in the trajectory-derived desired speed.

A.4 Control-Aware Objective: Controller Mapping and Derivation

The deployed planner predicts $T = 10$ waypoints. A generic waypoint-distance objective assigns attack value to all waypoint coordinates, although the downstream controller does not consume them equally. We therefore derive the functions f_{steer} and f_{speed} used in Section 3 directly from the deployed waypoint-to-control mapping.

Steering-related quantity. Let

$$u(W) = w_T - w_{T-1}. \tag{32}$$

The controller computes

$$\theta_{\text{wp}}(W) = \operatorname{atan2}\left(u_0(W), u_1(W) + 10^{-7}\right) \tag{33}$$

and maps the resulting direction to

$$a(W) = \operatorname{sign}\big(\theta_{\text{wp}}(W)\big)\,\frac{180 - \left|\deg\big(\theta_{\text{wp}}(W)\big)\right|}{90}. \tag{34}$$

For $T = 10$, this quantity depends on the final segment $w_{10} - w_9$.

Desired-speed quantity. Let

$$r_t = \|w_t\|_2. \tag{35}$$

The deployed controller computes desired speed from the first three radial increments,

$$v(W) = \frac{1}{3}\sum_{t=1}^{3}(r_{t+1} - r_t)\, v_{\max} \tag{36}$$

$$= \frac{\|w_4\|_2 - \|w_1\|_2}{3} v_{\max}. \tag{37}$$

Thus, the intermediate terms telescope: w_2 and w_3 cancel from the desired-speed computation. Waypoints $w_5, \ldots, w_8$ do not enter either equation 34 or equation 37.

For clean and attacked surrogate trajectories,

$$\Delta a = a(W^a) - a(W^c), \qquad \Delta v = v(W^a) - v(W^c). \tag{38}$$

The untargeted objective is

$$\mathcal{L}_{\text{ctrl}} = |\Delta a| + \lambda \frac{|\Delta v|}{v_{\max}}, \tag{39}$$

with

$$\lambda = 1, \qquad v_{\max} = 8. \tag{40}$$

The steering term is already normalized relative to the controller's steering range, whereas the speed term is normalized by the configured maximum speed.

When a valid counter-intent sign is available, SIPA replaces the absolute objective with

$$\mathcal{L}_{\mathrm{SIPA}} = s\,\Delta a - \lambda \frac{\Delta v}{v_{\max}}. \tag{41}$$

The first term favors a steering-related trajectory change opposite to the recovered route intent. Because Ψ does not recover velocity, SIPA does not infer a directional speed target; the second term instead consistently rewards reduced desired speed.

Our evaluated controller configuration uses `global_target_weight` $= 0$, under which the final waypoint segment directly contributes to the steering input. With the stock weight of 1, this direct steering pathway is removed, but predicted waypoints continue to affect desired speed and therefore throttle/braking. The control-aware objective should consequently be interpreted relative to the controller configuration used in evaluation rather than as a claim that all deployments consume waypoints identically.

A.5 Perception-Aware Gradient Filter

SIPA does not optimize a weighted sum of planning and perception losses. Instead, perception drift is used to filter coordinates of the planning-ascent direction.

Let

$$p^c = \mathrm{sigmoid}(z^c), \qquad p^a = \mathrm{sigmoid}(z^a) \tag{42}$$

denote clean and attacked classification probabilities. With $\tau_p = 0.2$, we define

$$\mathcal{L}_{\mathrm{FP}} = \mathrm{mean}\left[\left(\mathrm{ReLU}(p^a - p^c) \odot \mathbf{1}[p^c \le \tau_p]\right)^2\right]. \tag{43}$$

The operation order is ReLU, confidence mask, square, and spatial average.

$\mathcal{L}_{\mathrm{FP}}$ suppresses confidence increases at locations that were low-confidence in the clean prediction. This is a confidence-based surrogate rather than a ground-truth false-positive objective, and it does not impose equality of post-NMS boxes. In particular, the final SIPA configuration does not apply the complementary high-confidence suppression filter; the mechanism is therefore intentionally one-sided.

Let $d^{(k)}$ be the sign-based planning update before perception filtering. The FP-preserving coordinate mask is

$$K_{\mathrm{FP}}^{(k)} = \mathbf{1}\left[d^{(k)} \odot \nabla_\delta \mathcal{L}_{\mathrm{FP}} \le 0\right]. \tag{44}$$

The final SIPA configuration uses only $K_{\mathrm{FP}}^{(k)}$:

$$\widetilde{d}^{(k)} = d^{(k)} \odot K_{\mathrm{FP}}^{(k)}. \tag{45}$$

Accordingly, the filtering mechanism is a one-sided surrogate designed to suppress spurious confidence increases rather than a formal constraint that all detections remain identical. Perception preservation is therefore evaluated empirically against the same-frame clean prediction.

A.6 Optimization, Initialization, and Hyperparameters

Iterative optimization. At each iteration, SIPA differentiates the active planning objective with respect to the perturbation applied to the compromised collaborator feature. Let

$$g^{(k)} = \nabla_\delta \mathcal{L}^{(k)}, \tag{46}$$

where $\mathcal{L}^{(k)} = \mathcal{L}_{\mathrm{SIPA}}$ when a valid counter-intent direction is available and $\mathcal{L}^{(k)} = \mathcal{L}_{\mathrm{ctrl}}$ otherwise.

To stabilize the update direction, SIPA uses normalized momentum:

$$m^{(k)} = \mu m^{(k-1)} + \frac{g^{(k)}}{\max\left(\mathrm{mean}\,|g^{(k)}|, 10^{-12}\right)}, \qquad d^{(k)} = \mathrm{sign}(m^{(k)}). \tag{47}$$

Table 4: Principal settings of the final SIPA configuration.

Category	Setting	Value
Planning	Planner gradient source	Independent surrogate planner
	Planning objective	Control-aware objective
	Control direction	Counter-intent
	Intent threshold	0.01
	Speed weight λ	1.0
Spatial targeting	ROI threshold τ_R	0.1
	ROI dilation	2 feature cells
	Feature-support restriction	Enabled
Optimization	Perturbation budget ϵ	6
	Iterations K	60
	Step size α	0.25
	Momentum μ	0.9
Initialization	Random initialization	Enabled
	Temporal warm start	Enabled
	Perception-aware filtering	Enabled

The mean is computed over the complete perturbation tensor, and the momentum state is reset at the beginning of each frame.

The planning-ascent direction is then filtered by the perception-aware coordinate mask defined in Appendix A.5. Using the resulting direction $\widetilde{d}^{(k)}$, SIPA updates

$$\delta^{(k+1)} = \mathrm{clip}\left(\delta^{(k)} + \alpha\widetilde{d}^{(k)}, -\epsilon, \epsilon\right) \odot M. \tag{48}$$

Thus, every iteration remains within the element-wise ℓ_∞ budget and the request-guided spatial support.

Initialization and temporal warm start. SIPA reuses the perturbation from the previous frame when its tensor shape is compatible with the current frame. The carried perturbation is clipped to the current ℓ_∞ budget and re-masked using the current attack support M. If the tensor shape changes, the previous perturbation is discarded. The final configuration does not apply spatial rolling or pose-based motion compensation to the carried perturbation.

Random initialization is used when no usable warm start remains. Specifically, if the re-masked initialization satisfies

$$\|\delta \odot M\|_\infty < 10^{-6}, \tag{49}$$

SIPA samples a new perturbation uniformly from $[-\epsilon, \epsilon]$ and applies the current mask before optimization.

Final configuration. Table 4 summarizes the principal settings used in the final SIPA configuration.

A.7 Attack Interface and Access Assumptions

Attack interface. The perturbation is applied to the selected non-ego agent feature after pillar scatter and before the subsequent backbone, communication, warping, fusion, and downstream prediction stages. Thus, X_a in Section 3 denotes this intermediate per-agent representation, and gradients are taken only with respect to the perturbation δ. In the evaluated two-agent setting, the sole non-ego agent is an RSU, and only this collaborator is perturbed.

Gray-box model access. The attacker reproduces the deployed perception-side computation, including the backbone, fusion module, and detection head, to evaluate how its perturbed feature affects cooperative perception. These components use the same parameters as the victim-side perception stack. This model-knowledge assumption is motivated by homogeneous deployments using

a common perception stack and by the availability of publicly released CoDriving perception checkpoints. The attack optimizer has no access to the victim planner's parameters, gradients, predicted waypoints, or private route target, and obtains planning gradients exclusively from an independently trained and frozen surrogate planner. Accordingly, model independence in our gray-box setting applies to the planner, not to the perception-side computation. The surrogate planner is trained from a random initialization in a separate training run and never loads the victim planner's checkpoint at any stage; it is trained on Towns 1–4 and 6, and Town05 is excluded from its training towns.

Model knowledge and runtime-input access are specified separately. Candidate perturbations are evaluated using the actual ego-side perception inputs and the temporal and contextual tensors supplied by the evaluation pipeline. The ego-side BEV features are already available to the attacker, since a collaborator must reproduce the deployed perception-side computation to evaluate its own transmitted feature. The remaining temporal and contextual tensors are used only to run the attacker's own frozen surrogate planner and the perception-gradient filter — never the victim planner, which stays inaccessible throughout. They are treated as a runtime-input assumption of our evaluation, distinct from knowledge of the victim model, and none of them exposes the victim planner's parameters, gradients, outputs, or private route target.

Route and context inputs. The victim's private route target is never supplied to the surrogate attack objective. The surrogate target is obtained only from the protocol-visible request map: SIPA uses a recovered endpoint from $\Psi(R)$ when available, then the most recent valid recovered endpoint, and finally a zero target if recovery fails. No victim-planner output or ground-truth route annotation is used as a fallback.

B Experimental Details

This section provides the experimental details deferred from Section 4. We describe the closed-loop evaluation protocol, baseline implementations, metric aggregation, defense configuration, and ablation settings used in the main experiments. Method-specific implementation details, including request-map processing, intent recovery, perception-aware filtering, optimization, and attacker access, are provided in Appendix A.

B.1 Evaluation Protocol

We evaluate all methods in closed-loop CARLA Dosovitskiy et al. (2017) using the same CoDriving-based V2X-E2E system (Liu et al., 2025). The evaluation uses a fixed set of eight Town05 cooperative-driving routes, which is held constant across all methods and ablations. Each route is evaluated over three repeated passes. The victim and surrogate checkpoints are also fixed throughout the evaluation.

The evaluated configuration contains the ego vehicle and one non-ego collaborator. In our setting, the non-ego collaborator is an RSU and is the sole compromised agent; the ego feature is never modified. Only the feature transmitted by the compromised collaborator is perturbed, while the ego observations, route, traffic scenario, and remaining system inputs are unchanged.

All methods use the same victim–surrogate pair, route set, compromised-agent selection rule, and perturbation budget. Unless otherwise stated, attacks use an ℓ_∞ perturbation budget of $\epsilon = 6$ and $K = 60$ optimization iterations.

For same-frame comparisons, the clean counterfactual is obtained by re-evaluating the victim on the identical frame with the adversarial perturbation removed. This isolates the effect of the transmitted-feature perturbation from scene-to-scene variation. The precise feature-injection point and attacker access boundary are described in Appendix A.7.

B.2 Baseline Implementations

We compare SIPA with PGD*, SOMBRA (Wang et al., 2025), and SOMBRA*. A superscript $*$ denotes a baseline whose attack objective is replaced by our control-aware planning objective $\mathcal{L}_{\mathrm{ctrl}}$, while retaining the baseline's spatial attack mechanism. The starred variants are therefore used to separate the effect of the planning objective from the spatial and optimization mechanisms of SIPA.

PGD*. PGD* applies projected gradient descent (Madry et al., 2018) directly to the transmitted BEV feature of the compromised collaborator under the same ℓ_∞ budget and using the same surrogate planner as SIPA.

It uses the control-aware planning objective $\mathcal{L}_{\text{ctrl}}$ but disables SIPA-specific mechanisms: request-guided spatial localization, request-derived intent conditioning, and perception-aware coordinate filtering. We use the momentum-based update and temporal warm start employed by SIPA, yielding a strengthened feature-space PGD baseline rather than a textbook PGD implementation.

SOMBRA. SOMBRA (Wang et al., 2025) is a cooperative-perception attack that identifies influential regions of the shared representation through fusion-attention guidance and optimizes its original perception-oriented attack objective. It does not use the protocol-visible request map for either intent conditioning or spatial localization.

SOMBRA*. SOMBRA* retains SOMBRA's fusion-attention-based spatial guidance but replaces its perception objective with $\mathcal{L}_{\text{ctrl}}$. It therefore provides a planning-oriented feature attack whose spatial guidance is determined by fusion influence rather than by the request map.

Clean. Clean denotes the unperturbed setting, $\delta = 0$. It provides the reference closed-loop drive and the same-frame clean counterfactual used for perception comparisons.

B.3 Evaluation Metrics and Aggregation

We evaluate each attack in terms of perception preservation, closed-loop driving disruption, and defense detection.

Perception. Perception preservation is measured using the change in mAP at IoU 0.5 relative to the same-frame clean counterfactual, $\Delta\text{mAP@}0.5$, together with detection precision. Values of $\Delta\text{mAP@}0.5$ closer to zero indicate that the attacked perception output remains closer to its clean counterpart, while higher precision indicates fewer spurious detections.

Closed-loop driving. Closed-loop degradation is measured using the change in Driving Score, ΔDS, relative to the corresponding clean drive and collisions per kilometer (Coll./km). More negative ΔDS and higher Coll./km indicate stronger downstream driving disruption.

Metric aggregation. Perception metrics and Driving Score are first averaged within each route across the three repeated passes, after which the resulting route-level values are averaged with equal weight across the eight routes. This avoids assigning greater weight to routes that contribute more frames because the attack changes how far the ego vehicle progresses.

Collisions per kilometer are distance-pooled. The total number of collisions is summed over all evaluated runs and divided by the corresponding total driven distance. Defense detection is frame-pooled by summing rejected and evaluated frames over all runs.

B.4 Defense Configuration

We additionally evaluate attacks against CP-Guard (Hu et al., 2025), a perception-level V2X defense that verifies collaborators by measuring consistency between collaborative perception and the ego vehicle's own perception output.

CP-Guard employs collaborator-subset verification together with a collaborative consistency score. In our implementation, predicted bounding boxes are rasterized onto a BEV occupancy representation for consistency evaluation rather than using the segmentation representation of the original formulation. Because this modification changes the numerical scale of the consistency score, the original decision threshold does not transfer directly. We therefore normalize the collaborative consistency score using the ego vehicle's self-consistency score and calibrate the rejection threshold on clean frames. The resulting threshold is set to

$$\tau_{\text{def}} = 0.93.$$

Table 5: Principal settings used in the closed-loop evaluation.

Setting	Value
Simulator	CARLA
Environment	Town05
Number of routes	8
Compromised collaborators	1
Compromised-agent type	RSU
Perturbation budget ϵ	6
Attack iterations K	60
CP-Guard threshold τ_{def}	0.93

The evaluated V2X configuration contains only one non-ego collaborator. Consequently, collaborator-subset sampling reduces to a single verification of that collaborator on each evaluated frame. We report *Detection* as the percentage of verified frames in which the collaborator is rejected. For Clean, where no adversarial perturbation is present, the same quantity is interpreted as the false-alarm rate.

B.5 Ablation Configurations

We evaluate the contributions of spatial targeting, intent conditioning, and perception-aware filtering. Within each comparison, the remaining SIPA components are kept unchanged unless explicitly stated otherwise.

Spatial targeting. The full SIPA configuration uses the request-guided ROI described in Appendix A.2. For the *Random ROI* variant, the number of perturbed spatial cells is matched to the request-guided ROI independently on every frame, while their locations are sampled randomly. Thus, the support size is preserved and only the spatial placement of the perturbation changes.

Intent conditioning. The *w/o intent conditioning* variant removes the request-derived route-intent proxy used to condition the surrogate planner. No victim-planner output, victim waypoint prediction, or private route target is supplied as a replacement. The remaining attack objective, spatial targeting, and perception-aware filtering are kept unchanged.

Perception-aware filtering. The *w/o perception filtering* variant disables the perception-aware coordinate filter defined in equation 15. The intent conditioning, request-guided spatial mask, control-aware planning objective, perturbation budget, and optimization procedure are otherwise kept identical to the full SIPA configuration.

B.6 Summary of Evaluation Settings

For completeness, Table 5 summarizes the principal evaluation-level settings. Method-specific SIPA hyperparameters are reported separately in Appendix A.6.

C QUALITATIVE RESULTS

We provide qualitative examples illustrating how SIPA preserves cooperative perception while inducing downstream closed-loop driving failures.

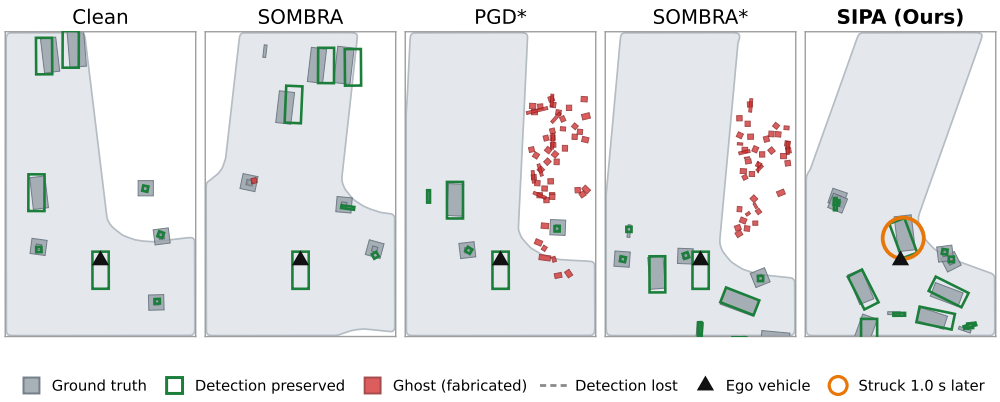


Figure 3: Qualitative comparison of cooperative perception under different attacks. PGD* and SOMBRA* introduce numerous ghost detections, whereas SIPA largely preserves the clean detection outputs without fabricated objects. Despite this perception preservation, the SIPA trajectory leads to a collision shortly afterward, illustrating that substantial downstream driving disruption can occur without visible perception corruption.

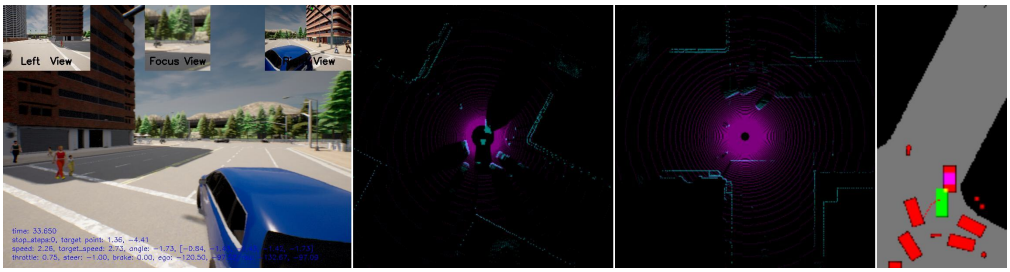


Figure 4: Representative closed-loop scene under SIPA. The camera and BEV views show the ego vehicle approaching the surrounding traffic while cooperative perception remains visually plausible. Nevertheless, the SIPA-induced planning deviation ultimately causes the ego vehicle to collide, demonstrating downstream driving failure despite preserved perception.

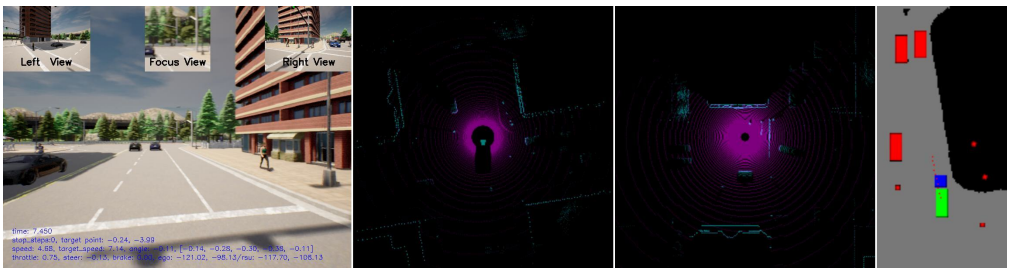


Figure 5: Representative closed-loop scene under the clean setting. Without adversarial perturbation, the ego vehicle maintains normal perception and follows the intended driving trajectory without the collision observed under SIPA.