Neural Reconstruction of LiDAR Point Clouds under Jamming Attacks via Full-Waveform Representation and Simultaneous Laser Sensing

Ryo Yoshida **Takami Sato** **Wenlun Zhang**
Yuki Hayakawa **Shota Nagai** **Kentaro Yoshioka**

Keio University, Yokohama, Japan

{ryo-yoshida,wenlun_zhang,hykwyuk,n-shota,kyoshioka47}@keio.jp
takami.sato@yos.elec.keio.ac.jp

Abstract

LiDAR sensors are critical for autonomous driving perception, yet remain vulnerable to spoofing attacks. Jamming attacks inject high-frequency laser pulses that completely blind LiDAR sensors by overwhelming authentic returns with malicious signals. We discover that while point clouds become randomized, the underlying full-waveform data retains distinguishable signatures between attack and legitimate signals. In this work, we propose PULSAR-Net, capable of reconstructing authentic point clouds under jamming attacks by leveraging previously underutilized intermediate full-waveform representations and simultaneous laser sensing in modern LiDAR systems. PULSAR-Net adopts a novel U-Net architecture with axial spatial attention mechanisms specifically designed to identify attack-induced signals from authentic object returns in the full-waveform representation. To address the lack of full-waveform representations in existing LiDAR datasets under jamming attacks, we introduce a physics-aware dataset generation pipeline that synthesizes realistic full-waveform representations under jamming attacks. Despite being trained exclusively on synthetic data, PULSAR-Net achieves reconstruction rates of 92% and 73% for vehicles obscured by jamming attacks in real-world static and driving scenarios, respectively. The dataset and code are publicly available through the project page: https://keio-csg.github.io/PULSAR_Net_project_page/.

1 Introduction

LiDAR sensors are indispensable for autonomous driving, providing precise 3D perception critical for safe navigation. By measuring the time-of-flight (ToF) of emitted laser pulses, LiDAR generates point clouds that enable vehicle localization and obstacle detection. However, this active sensing introduces a fundamental vulnerability: susceptibility to spoofing attacks that inject malicious infrared lasers at the sensor's operating wavelength [20, 12]. These attacks deceive vehicles by creating phantom objects or concealing real ones [11, 16]. Among them, jamming attacks are uniquely severe, lacking effective countermeasures in prior work [16, 17, 23]. Jamming floods the LiDAR with rapid laser pulses, randomizing the point cloud [16, 17] and blinding the vehicle. While modern LiDAR manufacturers have implemented several countermeasures, none of the existing defense mechanisms have effectively mitigated jamming attacks [16]. These risks highlight the urgent need for effective countermeasures against LiDAR jamming in autonomous driving systems.

In this work, we challenge this assumption by advocating that modern LiDAR systems already capture rich but abandoned information that enables discrimination between legitimate and malicious lasers.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

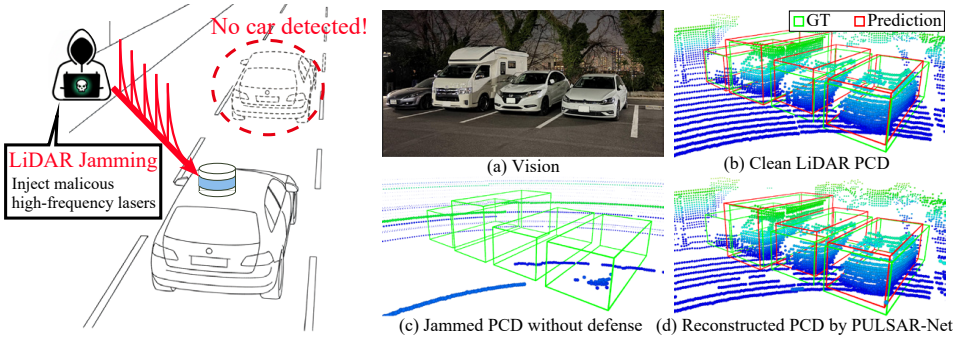


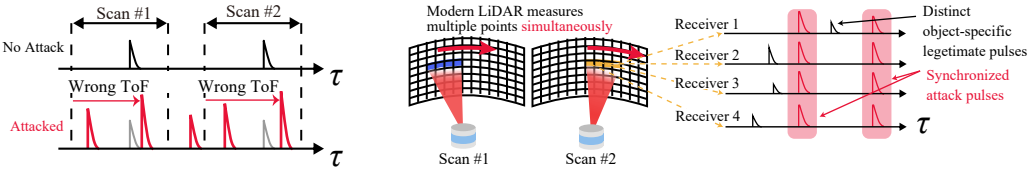
Figure 1: Reconstruction demo of our PULSAR-Net from LiDAR sensing under a jamming attack. Our PULSAR-Net can effectively recover the point cloud (d), which is faithful to the original one (b), by leveraging the intermediate full-waveform representation and simultaneous sensing feature available in modern LiDARs.

We identify two meaningful information sources: full-waveform representation and simultaneous laser sensing. Full-waveform representation, which records the complete temporal profile of returned laser intensity over the entire sensing period, contains subtle but remarkable signatures that distinguish authentic object reflections from attack-induced signals. Simultaneous laser sensing is a common feature in recent LiDARs to measure multiple points simultaneously, which enables us to identify attack signals. Because the attacker's laser beam diffuses over an area larger than the LiDAR's diameter, the resulting attack signal produces identical full-waveforms across simultaneously sensing receivers. In contrast, legitimate returns vary based on the specific objects reflecting them. To leverage the findings, we design PULSAR-Net (Point cloud Un-jamming via LiDAR Signal Adaptive Reconstruction), the first successful defense against jamming attacks on LiDAR systems, specifically High-Frequency Removal (HFR) attacks [16, 17], one of the most effective jamming attacks to date. We adopt a novel U-Net-based architecture with axial spatial attention, explicitly designed for attack signal segmentation in full-waveform representation, to effectively capture the structure of attack pulses in simultaneously sensed full-waveform data.

We first design a realistic synthesis methodology to generate full waveforms under jamming attacks from existing point cloud datasets. Our goal is to enable PULSAR-Net to generalize effectively to real-world LiDAR point clouds despite being trained solely on synthesized data derived from benign point clouds. This approach is necessary because collecting LiDAR data under actual jamming attacks is generally impractical, as deploying such attacks across diverse real-world scenarios is highly challenging. For this purpose, we reconstruct the intermediate full-waveform representation from post-processed point clouds given known LiDAR specifications. Building on these synthesized waveforms and the scan pattern of simultaneous laser sensing, we apply a jamming attack synthesis methodology that injects high-frequency attack pulses with timing randomization to reproduce realistic jamming conditions.

To evaluate PULSAR-Net, we conducted a large-scale evaluation on synthetic datasets derived from popular LiDAR benchmarks. Furthermore, we performed a real-world evaluation involving a physically deployed jamming attack against a production-ready LiDAR system, which was customized in collaboration with a major supplier to stream full-waveform data. Despite training exclusively on synthetic data, PULSAR-Net exhibits robust generalization to physical attacks (Fig. 1 (ii)). These results establish PULSAR-Net as the first practical defense capable of recovering point clouds under LiDAR jamming. This work makes the following contributions:

- We design PULSAR-Net, the first successful defense against jamming attacks on LiDAR systems, featuring a U-Net-based architecture with axial spatial attention specifically designed for attack-signal segmentation in the full-waveform representation.
- We develop a physics-based synthesis pipeline that generates realistic full-waveform data under jamming attacks from existing point cloud datasets.
- We demonstrate that PULSAR-Net achieves a recovery rate of $\geq$82% for vehicles obscured by jamming attacks on our generated dataset, representing an $\geq$2-fold improvement over a naive baseline approach on our datasets.



(i) LiDAR jamming attack (ii) Spatial-temporal correlation by simultaneous sensing

Figure 2: Conceptual illustration of a LiDAR jamming attack and simultaneous sensing.

- In collaboration with a major supplier, we integrated PULSAR-Net into a production-ready LiDAR with full-waveform access. Despite training solely on simulated data, it achieves reconstruction rates of 92% and 73% in real-world static and driving scenarios, respectively.

2 Related Work

2.1 LiDAR Spoofing Attacks

LiDAR sensors introduce a fundamental vulnerability: attackers can spoof systems by injecting malicious, same-wavelength infrared laser signals [20, 12]. These attacks deceive vehicles by creating phantom objects or concealing real obstacles [11, 16]. In particular, HFR attack [16, 17] is currently known as one of the most effective LiDAR spoofing attacks, which emits laser pulses at a frequency significantly higher than the LiDAR's own pulse rate, saturating the sensor's receiver. Unlike conventional attacks that require precise synchronization with the victim LiDAR, the HFR attack can disrupt a wide range of LiDAR models with different sensing architectures, including those with randomized laser firing patterns. This disruption blinds the victim vehicle to its environment, potentially causing fatal consequences like rear-end collisions, as depicted in Fig. 1 (i). Since existing commercial and academic defenses remain ineffective against jamming [16, 17, 23], developing robust countermeasures is an urgent necessity.

2.2 Full-Waveform Representation in LiDAR

While jamming attacks, such as HFR attack, can make the point cloud completely random, this does not mean there is no way to obtain information from the legitimate laser reflection, because the legitimate laser reflection does actually reach the LiDAR sensor, albeit typically weaker than the attack pulses. For each point in the point cloud, the LiDAR obtains a time-resolved full waveform from the returning laser light, and then a digital signal processor (DSP) pipeline detects a peak in the waveform [18, 14]. The peak is generally one of the strongest attack pulses as shown in Fig. 2 (i). Critically, only the final point cloud is typically stored and used for downstream perception tasks. The intermediate full-waveform data, which contains rich temporal information, is discarded after peak detection to reduce data storage, bandwidth, and latency requirements. A recent study [18] finds that the full-waveform representation can fit a tokenization of Transformer [24] and demonstrates the performance improvements, particularly in foggy conditions.

2.3 Simultaneous Laser Sensing

Modern LiDAR systems employ a critical design feature: they fire lasers from multiple emitters simultaneously and receive them within the same time [16] as listed in Table 1, and the number of simultaneously sensed points continues to increase with advances in semiconductor technology [27]. Fig. 2 (ii) illustrates how LiDAR scans several adjacent points at the same time with multiple laser emitters and receivers. This creates strong spatial-temporal correlations in the full-waveform data, allowing us to distinguish the attack laser pulses from legitimate laser reflections. Specifically, the same pattern of the attack pulses will be observed in multiple full waveforms received at the same time, as shown in Fig. 2 (ii). The attacker's laser is emitted from outside the LiDAR and is dispersed beyond the diameter of the LiDAR, making it impossible to selectively attack a specific receiver. On the other hand, the legitimate laser is emitted inside the

Table 1: Simultaneous laser sensing capabilities of commercial LiDAR systems. ϕ is the number of simultaneous sensing lasers.

	Robosense M1P [8]	Hesai ATX [2]	Ouster VLS128 [1]
			
ϕ	32	64	8

LiDAR, and thus each receiver can receive the corresponding laser due to the directness of light; i.e., the detected laser patterns differ based on the object hit by each laser beam.

3 Full-Waveform Jamming Attack Simulation

To effectively train our PULSAR-Net, we first design a methodology to generate attacked full-waveform datasets with simultaneous laser-sensing patterns from existing LiDAR point cloud data. Ideally, the data should be collected in the real world without any synthesis, but gathering full-waveform data and annotating it at scale is highly costly. However, data collection requires significant effort to cover a wide variety of LiDARs, scenes, attack setups, and victim objects. Moreover, setting up jamming attacks at every data collection point is practically infeasible. To address this, we simulate the intermediate full-waveform representation from given LiDAR specifications and point cloud data in existing LiDAR datasets such as KITTI [13], nuScenes [10], and Waymo Open Dataset [22], which include only the final point cloud outputs $(x, y, z, \text{intensity})$ without the full-waveform representations essential for our defense mechanism. Our goal is to achieve sufficient real-world generalizability while training only on simulated data.

Notation. We denote the full-waveform tensor as $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{H\times W\times D}$, where H, W, and D denote the number of channels (vertical angles), horizontal resolution (azimuth samples), and temporal bins (full-waveform bin length), respectively. $\boldsymbol{\mathcal{A}}$ represents a single LiDAR scan of 360°. For each measurement direction (i, j) where $i \in \{1, \ldots, H\}$ and $j \in \{1, \ldots, W\}$, the LiDAR captures a full waveform $\boldsymbol{\mathcal{A}}_{ij} \in \mathbb{R}^D$ over time bins $t \in \{0, \ldots, D-1\}$. We denote the discrete scan timestamp of a single 360° scan as $\Theta = \{0, 1, \ldots, \tau_{\max} - 1\}$.

3.1 Full-Waveform Representation Modeling

Given benign point cloud data from an existing LiDAR dataset, we model its full-waveform representation. The point cloud data only provides the location of the strongest return at each measurement (i, j). A naive approach would be to assign a non-zero intensity value in $\boldsymbol{\mathcal{A}}_{ij}$ at the corresponding time bin, and set the others to zero. However, this digital approach fails to capture the analog characteristics of the reflected laser pulse. Following prior work [28], we adopt a Gaussian pulse model $p(t) = \exp\left(-\frac{t^2}{2\sigma^2}\right)$ to simulate realistic reflected pulse returns. We model the full-waveform tensor $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{H\times W\times D}$ of a given point cloud as $\mathcal{L}_{ijt} = \gamma\, i_{ij} \cdot p(t - 2d_{ij}/c_0)$, where γ is the pulse amplitude scaling factor to convert the unit Gaussian pulse to an appropriate amplitude for the target LiDAR (we used γ=12 for the KITTI dataset and γ=0.156 for the nuScenes dataset to surpass the legitimate pulses sufficiently), d_{ij} and i_{ij} are the distance and intensity at measurement direction (i, j), and c_0 is the light speed.

3.2 Simultaneous Laser Sensing Modeling

As described in §2.3, we leverage the unique characteristic of modern LiDARs that emit and sense multiple lasers simultaneously. To model this behavior, we introduce a scan timing matrix $\boldsymbol{\mathcal{T}} \in \Theta^{H\times W}$ whose element $\mathcal{T}_{ij} \in \Theta$ contains the scan start timestamp of each measurement direction (i, j). Two measurements (i, j) and (s, k) are scanned simultaneously if and only if $\mathcal{T}_{ij} = \mathcal{T}_{sk}$.

3.3 Jamming Attack Modeling for Full Waveforms

We model the attacked full-waveform tensor as the superposition of the benign signal, the attack signal, and the noise. Let $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{H\times W\times D}$ denote the benign full-waveform tensor, $\boldsymbol{\mathcal{F}} : \Theta^{H\times W} \to \mathbb{R}^{H\times W\times D}$ denote the attack pulse generation function parameterized by the scan timing matrix, and $\boldsymbol{\mathcal{N}} \in \mathbb{R}^{H\times W\times D}$ denote the observation noise. The attacked full-waveform tensor $\check{\boldsymbol{\mathcal{A}}} \in \mathbb{R}^{H\times W\times D}$ is given by $\check{\boldsymbol{\mathcal{A}}} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{T}}) + \boldsymbol{\mathcal{N}}$. The function $\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{T}})$ ensures that measurements with identical scan timestamps receive the same attack waveform, reflecting the simultaneous laser sensing. For each measurement direction (i, j), the attack pulse $\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{T}})_{ijt}$ is constructed as a periodic train of Gaussian pulses: $\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{T}})_{ijt} = \gamma_A \sum_{k=1}^{K} p(t - (\tau_k + \epsilon_k))$, where $K = \lfloor T_{\max}/\delta_{\text{attack}} \rfloor$ is the number of attack pulses within the observation window, $T_{\max}$ is the total time duration of the full waveform, δ_{attack} is the attack pulse interval (e.g., $\delta_{\text{attack}} = 100$ ns for the 10 MHz attack pulse frequency used in our experiments), γ_A is the attack pulse amplitude, and $\tau_k = k \cdot \delta_{\text{attack}}$ is the nominal timing of the k-th attack pulse. We add timing jitter $\epsilon_k \sim \mathcal{U}(-20\text{ ns}, 20\text{ ns})$ to each pulse timing τ_k. This design can model more general jamming attacks than current state-of-the-art attacks, such as HFR [16]

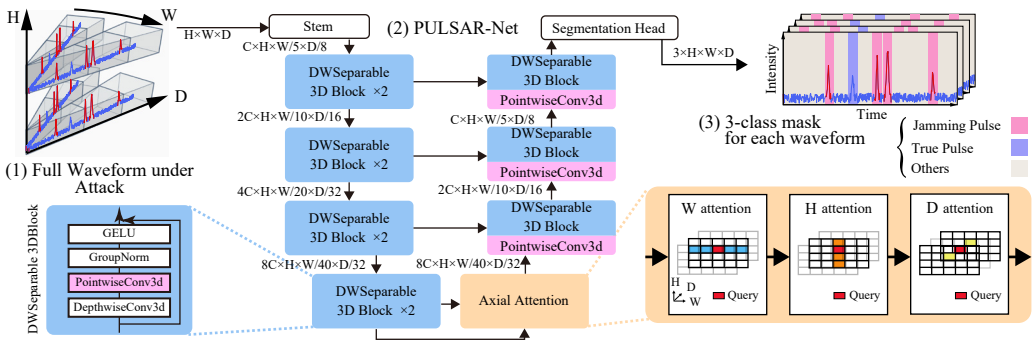


Figure 3: Overview of PULSAR-Net, a U-Net-inspired architecture using depthwise-separable 3D convolutions for efficient, lightweight feature extraction. It incorporates axial attention to model both spatial and temporal dependencies within full-waveforms.

and A-HFR attack [17], which inject periodic attack pulses that may be predictable to the defense side. The timing jitter ϵ_k disrupts the periodic structure of the attack pulses, whereas $\mathcal{N}$ denotes observation noise. This enables us to simulate randomized jamming pulses. Appendix G details the dataset-generation parameters.

4 PULSAR-Net: Neural LiDAR PCD Reconstruction

To fully leverage the full-waveform representation (§2.2) and the simultaneous laser sensing (§2.3), we design PULSAR-Net, a novel U-Net-based neural network architecture with axial spatial attention specifically designed for attack signal segmentation in full-waveform data as depicted in Fig. 3. Our depthwise-separable 3D convolutions and axial attention, can capture LiDAR-agnostic physical principles rather than dataset-specific artifacts, enabling robust defense performance across diverse LiDAR systems without requiring hardware-specific retraining.

4.1 Segmentation-based Full-Waveform Denoising

We formulate the LiDAR point cloud reconstruction as a segmentation-based denoising in full-waveform representation. For each measurement, we predict three classes: (1) background (no reflection), (2) legitimate reflection from an object, and (3) attack pulse. Our goal is to train a denoising model $f_D(\cdot)$ that can accurately predict the binary segmentation mask $\boldsymbol{\mathcal{M}}_{attack}$ corresponding to the malicious attack pulses in the full waveforms. We apply the mask to suppress the attack pulses in the full waveform as follows: $\hat{\boldsymbol{\mathcal{A}}} = \check{\boldsymbol{\mathcal{A}}} \odot (1 - \boldsymbol{\mathcal{M}}_{attack})$. We obtain the reconstructed point cloud by taking the peak of each denoised full waveform. This design reduces computational cost and even improves performance. A naive approach is to predict the denoised full waveform via regression. However, this approach underperformed since information was lost during resizing before feeding to the reconstruction model. In contrast, our masking-based approach can preserve rich information of the full waveform in the high-resolution original space.

4.2 Model Architecture

We design a dedicated 3D U-Net architecture that operates directly on LiDAR waveform data, as illustrated in Fig. 3, to satisfy the two key requirements: **i)** the model should fully exploit both spatial and temporal information to accurately identify malicious peaks within the waveform data accurately **ii)** the model should be computationally efficient to enable real-time signal processing, thereby improving its practical applicability. The encoder is composed of multiple depthwise-separable 3D convolutional blocks and progressive downsampling layers to extract multi-scale spatial–temporal features. The decoder mirrors the encoder with corresponding upsampling stages, where skip connections integrate high-resolution details with semantically rich features from the encoder. Finally, a segmentation head predicts the attack mask for each voxel, enabling precise separation between malicious and clean waveform peaks. To further strengthen these two design objectives, we incorporate two specialized blocks into the network.

Depthwise-Separable 3D Convolution.

Processing full-waveform LiDAR data with standard 3D convolutions is computationally prohibitive due to strict memory constraints. To address this, we employ depthwise separable 3D convolutions [25]. By decoupling spatial and channel-wise operations, this design reduces parameters and computational overhead by over 97% compared to standard 3D convolutions. This efficiency enables real-time deployment while preserving the model's representational capacity.

Axial Spatial-Temporal Attention. While depthwise-separable 3D convolutions in the U-Net structure capture local spatial–temporal dependencies, they remain limited in modeling global waveform correlations. To overcome this, we introduce an axial attention module that applies sequential self-attention along the azimuth, elevation, and temporal axes. Unlike standard 3D attention's cubic complexity, this axial decomposition captures long-range dependencies within each dimension at a fraction of the computational cost. This design is ideally suited for exploiting simultaneous laser sensing patterns. Legitimate returns from simultaneously fired lasers exhibit varying waveforms based on the encountered objects. In contrast, because attackers cannot selectively target individual receivers (§2.3), injected signals manifest as identical waveforms across all simultaneously-sensed channels. By independently applying attention along the spatial axes, our architecture efficiently compares these channels to detect the identical patterns characteristic of attacks, precisely where legitimate signals naturally differ.

Loss Function Design. To handle imbalance classes, we use a weighted cross entropy loss, $\mathcal{L}_{\text{WCE}} = -\sum_{c\in\{0,1,2\}} w_c \sum_i y_{i,c} \log p_{i,c}$, where $(w_0, w_1, w_2) = (0.1, 0.3, 0.6)$. With this loss and Dice loss [21], we train the PULSAR-Net with the loss function $\mathcal{L}_{\text{WCE}} + 0.5 \cdot \lambda_{\text{Dice}}$.

5 Evaluation

We evaluate PULSAR-Net on synthesized jamming-attack datasets generated using the methodology in §3 and on a real-world attack case study. Our evaluation targets two questions: (i) can the model recover the original point geometry at the point level, and (ii) does the recovered point cloud preserve downstream perception performance for 3D object detection.

5.1 Experimental Setup

Evaluation Dataset. We construct our evaluation datasets from KITTI [13] and nuScenes [10] datasets. We synthesize the jamming attack-impacted data according to the methodology described in §3. For each dataset, we extracted 1,100 scenes: 1,000 for training and 100 for testing. We inject the jamming attack effects into $\pm 45^\circ$ in front of the victim with the target LiDAR. We call the datasets as KITTI-J and nuScenes-J, respectively. More details of our training setup are included in Appendix C.

LiDAR Specifications. We use (32, 1800, 800), i.e., 32 vertical channels, 0.2° horizontal resolution, and 800 full-waveform dimensions, as the input from LiDAR. This specification is consistent with HDL-32E [3] LiDAR used in nuScenes datasets. The KITTI dataset uses HDL-64E [4], which has more vertical channels and horizontal resolution. We downsample it to 32 vertical channels and 0.2° horizontal resolution. We note that this downsampling is only applied when training the attack mask as described in §4.1 and the final point-cloud dimension will not be changed. For the simultaneous laser sensing, we assume that the consecutive horizontal points are scanned at the same time, according to the modern design of LiDAR, such as Huawei L106 [5] and Hesai ATX [2].

Metrics. We evaluate the denoising capability using two metrics: point-level recovery accuracy and object detector-level mAP. *Point-level recovery accuracy* measures whether each reconstructed point in the 90° attack area is located within 0.5 m of the corresponding original benign point. Following LiDAR physics principles, this 0.5 m threshold is evaluated along the laser direction from the LiDAR origin, meaning the accuracy is calculated independently for each laser emission. *Object detector-level mAP* is a metric to evaluate the autonomous driving application-level defense capability. We calculate the mAP of all objects in the attack area, the 90° area in front of the victim. We use PointPillars [15], CenterPoint [26] and PV-RCNN [19], implemented in mmdetection3d [6]. We employed these models pre-trained on the KITTI and nuScenes datasets to evaluate KITTI-J and nuScenes-J, respectively.

Baseline Methods. We compare our approach against two baselines: naive average subtraction and the Neural DSP [18]. Naive average subtraction simply computes the mean of simultaneously sensed full waveforms and subtracts this average from each individual waveform. We expect this averaged

Table 2: *Point-level recovery accuracy* of PULSAR-Net (Ours) and the baselines (Naive Average Subtraction and Neural DSP) on the nuScenes-J and KITTI-J datasets. ϕ is the number of simultaneous sensing lasers. Results for learned methods are reported as mean $\pm$ standard deviation over three training runs with different random seeds; deterministic baselines are evaluated once.

	Ours		Naive Average Sub.		Neural DSP [18]		No Defense	
ϕ	nuScenes-J	KITTI-J	nuScenes-J	KITTI-J	nuScenes-J	KITTI-J	nuScenes-J	KITTI-J
1	45.90 ± 6.90	64.16 ± 7.55	-	-	4.94 ± 0.49	5.80 ± 1.63	3.37	1.32
4	79.19 ± 1.07	87.83 ± 0.34	38.22	28.56	6.46 ± 1.23	17.28 ± 3.57	3.36	1.32
5	83.16 ± 0.73	88.25 ± 0.14	**38.26**	30.28	10.56 ± 2.02	21.29 ± 4.41	3.39	1.31
10	84.01 ± 0.64	88.36 ± 0.07	37.77	35.81	36.22 ± 1.17	$\mathbf{25.42 \pm 3.50}$	3.39	**1.34**
25	85.28 ± 0.40	$\mathbf{88.46 \pm 0.07}$	36.91	41.95	46.88 ± 1.81	21.74 ± 0.64	**3.42**	1.30
55	85.38 ± 0.39	88.32 ± 0.06	35.50	**44.63**	52.15 ± 3.35	16.72 ± 2.57	3.41	1.31
110	85.18 ± 0.48	88.15 ± 0.06	32.60	34.20	$\mathbf{52.97 \pm 5.34}$	16.69 ± 2.82	3.39	**1.34**
225	$\mathbf{85.70 \pm 0.05}$	88.34 ± 0.03	34.89	15.54	45.01 ± 6.66	17.15 ± 3.34	3.35	1.31

Table 3: *Object detector-level mAP* of PULSAR-Net (Ours) and baselines on the nuScenes-J and KITTI-J datasets. ϕ denotes the number of simultaneous sensing lasers. Results for learned methods are reported as mean $\pm$ standard deviation over three training runs with different random seeds; deterministic baselines are evaluated once.

		nuScenes-J (mAP)		**KITTI-J (mAP)**	
ϕ	**Method**	PointPillars	CenterPoint	PointPillars	PV-RCNN
25	No Defense	0.39	0.08	0.00	0.00
	Naive Average Sub.	1.94	0.00	21.81	41.06
	Neural DSP [18]	22.07 ± 4.87	27.92 ± 2.99	4.20 ± 2.22	6.11 ± 3.04
	Ours	$\mathbf{64.19 \pm 0.38}$	$\mathbf{59.23 \pm 0.22}$	$\mathbf{48.22 \pm 0.14}$	$\mathbf{56.66 \pm 0.71}$
110	No Defense	0.95	0.32	0.00	0.00
	Naive Average Sub.	2.56	4.04	16.35	32.94
	Neural DSP [18]	44.26 ± 6.11	48.31 ± 1.95	2.65 ± 1.83	4.61 ± 2.79
	Ours	$\mathbf{65.41 \pm 0.31}$	$\mathbf{58.68 \pm 0.28}$	$\mathbf{47.86 \pm 0.24}$	$\mathbf{56.01 \pm 0.37}$
225	No Defense	0.77	0.49	0.00	0.00
	Naive Average Sub.	4.09	9.11	0.14	2.34
	Neural DSP [18]	40.41 ± 8.60	46.77 ± 3.61	2.98 ± 2.36	4.59 ± 3.46
	Ours	$\mathbf{64.26 \pm 0.17}$	$\mathbf{58.91 \pm 0.29}$	$\mathbf{49.19 \pm 0.22}$	$\mathbf{56.26 \pm 0.53}$
	No Attack	69.87	72.72	57.51	62.75

waveform to preserve the attack pulses while canceling out legitimate laser reflections, since the timing of legitimate peaks varies across lasers, as discussed in §2.3. Neural DSP is a transformer-based architecture that directly processes full waveforms rather than point cloud data. While it was not designed to handle attack-compromised data, we include it to highlight the impact of our design over the mere integration of full-waveform information. Appendix D provides implementation details of Neural DSP.

We do not include conventional point-cloud denoising filters in the main comparison because jamming randomizes the final point cloud and destroys its coherent local geometry. We evaluate Radius Outlier Removal (ROR) and Statistical Outlier Removal (SOR) and find that they remove isolated points but do not reconstruct objects erased by the attack. Detailed results are provided in Appendix B.

5.2 Results

Tables 2 and 3 show the evaluation results of our method and baselines on the KITTI-J and nuScenes-J datasets. Complete object-detection results across all evaluated simultaneous-sensing configurations are provided in Appendix F. As shown, our PULSAR-Net reconstructs $\geq$79% of points on nuScenes-J and $\geq$87% of points on KITTI-J when simultaneous sensing is available (i.e., the number of simultaneous sensing lasers ϕ is > 1), whereas the naive average subtraction method recovers only $\leq$39% and $\leq$45%, respectively. On the object detector-level mAP, our method can recover almost the same mAP ($\leq$65.41 mAP) as the benign one (69.87 mAP) without attack, while the naive average subtraction method can recover only up to 9.11%, on the nuScenes-J dataset. On the KITTI-J dataset, while PULSAR-Net outperforms the naive average subtraction method, the performance is not as significant as the results in the nuScenes-J, and the naive average subtraction method also shows relatively high performance. We attribute the performance difference to the capabilities of the object detectors trained on each dataset, since the point-level accuracy is similar across both datasets.

Table 4: Model transferability analysis on the point-level accuracy metric. The model is trained on the source dataset (row) and evaluated on the target dataset (column). The number of simultaneous sensing ϕ is 5.

		Target	
		nuScenes-J	KITTI-J
Source	nuScenes-J	**83.17**	81.73
	KITTI-J	71.61	**88.39**

Table 5: Ablation study of axial spatial-temporal attention on the nuScenes-J dataset with ϕ=5.

Attention			Point	Obj.	
W	H	D	Acc.	mAP	FPS
			67.85	25.01	32.89
✓			82.98	62.05	31.94
	✓		80.34	62.50	32.25
		✓	81.93	62.55	32.36
✓	✓	✓	**83.17**	**63.26**	31.74

Since the KITTI dataset is dated compared to the nuScenes, we will focus on the evaluation of the nuScenes-J in the rest of this paper.

For the lower performance of the naive average subtraction method, we find that it caused a large number of false positives because the attack pulses were not completely removed. The majority of attack pulses are dismissed by the naive average subtraction method, but it was not perfect, and the remaining attack pulses are still detected as points. Since the naive average subtraction method algorithm averages over multiple waveforms by design, legitimate laser pulses are inherently suppressed, making it difficult to distinguish true laser pulse peaks from false positives. These results indicate that using full-waveform data and simultaneous laser sensing information can provide significant benefits in denoising the impacts of jamming attacks. However, a straightforward average subtraction approach fails to handle the information adequately, and thus, the neural network approach of our PULSAR-Net plays a significant role in leveraging it. Neural DSP demonstrates some attack-mitigation capability by leveraging full-waveform information, but falls short of the performance achieved by PULSAR-Net. This is because Neural DSP is not designed for jamming attack scenarios, where the full waveform contains an excessive number of peaks that exceed the capacity of its current tokenization granularity. Finer-grained tokenization may alleviate this limitation, but remains an open question given the associated increases in computational and memory requirements.

Transferability Analysis. We evaluate the transferability of our PULSAR-Net trained on the different datasets to evaluate if our model learns LiDAR-agnostic physical principles rather than dataset-specific artifacts, enabling robust defense performance across diverse LiDAR systems. Table 4 shows the model transferability analysis between KITTI-J and nuScenes-J datasets, on the point-level accuracy metric. The evaluation model is trained on one source dataset and evaluated in another target dataset. The number of simultaneous sensing lasers ϕ is 5, where the performance is already saturated. As shown, the PULSAR-Net trained on the nuScenes-J maintains high point-level accuracy even in the KITTI dataset. This indicates that the PULSAR-Net can acquire LiDAR-agnostic knowledge. However, it is still hard to deny the possibility that our PULSAR-Net may rely on some artifacts introduced by our attack simulation. We will further evaluate the transferability against the actual real-world attacks in §6.

Inference Efficiency. PULSAR-Net achieves 31.74 FPS on an RTX 5090 even with the full size of the model as shown in Table 5. The perception module of autonomous driving typically works at 10-20 FPS [7, 9], PULSAR-Net's throughput is on par with these industry standards. To further minimize computational burden, our defense can be paired with a lightweight attack detector such as [23], ensuring PULSAR-Net executes conditionally only under jamming attacks.

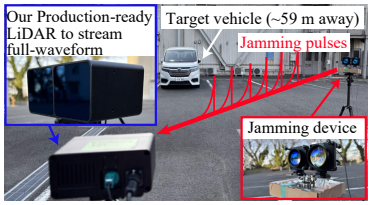
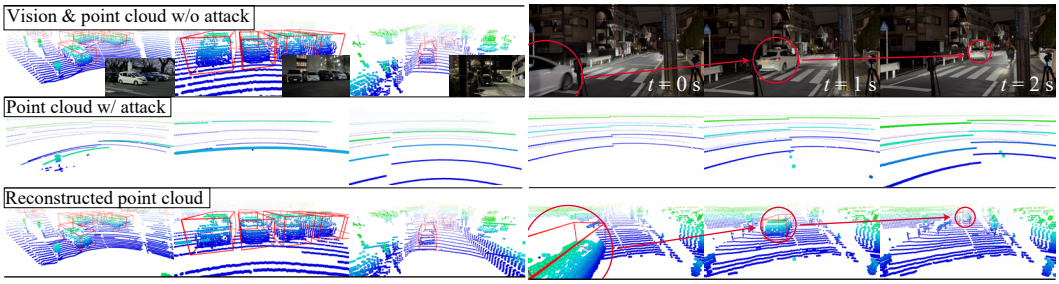


Figure 4: Experimental setup of a jamming attack on our LiDAR developed with a major LiDAR supplier to stream full waveform

Attention Study. To evaluate the effectiveness of the axial spatial-temporal attention, we conduct an ablation study by applying attention along each individual axis and measuring its contribution to segmentation performance. We used the nuScenes-J dataset with ϕ=5. As shown in Table 5, applying attention to a single axis provides marginal improvements, whereas the best performance is achieved when attention is jointly applied to all three axes. Since the FPS drop is negligible with axial attention enabled, our denoising model employs attention along all three axes of the waveform data. Further ablation results are provided in Appendix E.



(i) Static scenarios (ii) Driving scenario

Figure 5: Results of the real-world experiment : (i) benign, attack-compromised, and reconstructed point clouds in a static scenario with a reference vision, and (ii) 3 frames of camera frames and point clouds of benign, attack-compromised, and reconstructed point clouds in driving scenario

6 Real-World Evaluation

We conduct real-world evaluations to validate whether PULSAR-Net achieves sufficient generalizability when trained exclusively on simulated data. To this end, we develop a production-ready LiDAR sensor in collaboration with a major LiDAR manufacturer, capable of outputting full-waveform data while preserving the existing point cloud output, ensuring that our defense pipeline can operate in parallel with existing LiDAR perception systems in autonomous driving.

Experimental Setups. The LiDAR (Fig. 4) we developed has a 90.24° horizontal and 26.3° vertical field of view (FoV), an angular resolution of 0.47° in both horizontal and vertical directions, and the number of simultaneous sensing lasers ϕ is 96. We launch jamming attacks against the LiDAR. We reproduce the HFR attack [17], the current state-of-the-art LiDAR jamming attack. We use the PULSAR-Net trained on the generated dataset based on nuScenes without any LiDAR-specific customization. We conducted extensive experiments across seven scenarios, comprising three driving and four static sequences, totaling 700 frames (100 frames per sequence). To ensure a comprehensive evaluation, we manually annotated 3,106 vehicle instances. The ground-truth bounding boxes exhibit significant spatial diversity, with distances ranging from approximately 3.4 m to 59 m from the ego-vehicle. The LiDAR sensor was positioned at vehicle height, with a jamming device placed at various positions and angles relative to the sensor, as illustrated in Fig. 4. Our PULSAR-Net, trained on the nuScenes-J dataset, was evaluated using three representative 3D object detectors: PointPillars [15], CenterPoint [26], and PV-RCNN [19]. These models were trained on the nuScenes dataset and implemented via the mmdetection3d framework [6].

6.1 Results

Table 6 presents the results for the four static and three driving scenarios. Fig. 5 illustrates the benign point cloud, the point cloud jammed by the attack, and the point cloud reconstructed by PULSAR-Net.

Table 6: Quantitative results of the case study. We evaluate the mAP across three driving (moving) and four static scenarios. Note that benign (no attack) data is not available for moving scenarios.

	Static (4 scenarios)			Moving (3 scenarios)	
Model	No Attack	No Defense	Ours	No Defense	Ours
PointPillars	49.7	0.10	**49.8**	0.70	**50.7**
CenterPoint	67.1	0.00	**69.4**	0.20	**63.1**
PV-RCNN	44.0	0.00	**49.8**	0.00	**53.8**

Static Scenarios. Table 6 shows that jamming nearly collapses detection performance (mAP ≈0) in static scenes, while our reconstruction restores detector accuracy to near the benign level. Specifically, PULSAR-Net achieves 49.8/69.4/49.8 mAP for PointPillars/CenterPoint/PV-RCNN under attack, matching or slightly exceeding the no-attack baselines (49.7/67.1/44.0). With the best model (CenterPoint with IoU threshold=0.4), PULSAR-Net can reconstruct 91.7% reconstruction accuracy for vehicles erased by attacks. This indicates that PULSAR-Net can not only reconstruct point clouds preserving the vehicle geometry needed for reliable detection, but also improve the detection performance with full-waveform information, as also observed in prior work [18].

Driving Scenarios. Table 6 shows that jamming reduces detection to near zero in dynamic scenes (0.70/0.20/0.00 mAP), while PULSAR-Net restores strong performance (50.7/63.1/53.8) across all three detectors. This consistent recovery under motion indicates that the reconstruction preserves

sufficient temporal and geometric cues for reliable detection even when vehicles are moving at up to 30.5 km/h. With the best model (CenterPoint with IoU threshold=0.4), PULSAR-Net achieves a reconstruction accuracy of 73.2% for vehicles erased by attacks. We note that point clouds without attack are unavailable because we could not reliably replay driving scenarios in our testing facility, as shown in Fig. 5 (ii). A demonstration video is available on our project page and a description of the video is provided in A. Further discussion on our real-world evaluation is included in Appendix H.

7 Conclusion

We establish PULSAR-Net, the first effective defense mechanism capable of reconstructing original point clouds from LiDAR sensing under jamming attacks. By leveraging previously unutilized intermediate full-waveform representations and simultaneous laser sensing in modern LiDAR systems, we demonstrated that it is possible to distinguish malicious attack signals from legitimate object reflections, a task previously considered theoretically impossible at the point cloud level. We evaluate PULSAR-Net against the HFR attack, the state-of-the-art LiDAR jamming attack, in both synthetic and real-world settings. PULSAR-Net demonstrates over 82% point-level recovery accuracy, representing more than 2 times improvement over a naive baseline. To validate its real-world viability, we collaborated with a major supplier to integrate PULSAR-Net into a production-ready LiDAR system. Despite being trained only on synthetic data, our approach demonstrates high generalizability, achieving reconstruction success rates of 92% in static scenarios and 73% in driving scenarios.

Acknowledgments

We gratefully acknowledge Takahiro Kado, Ibuki Fujioka, and Taro Beppu at Sony Semiconductor Solutions for their generous support and for providing the Sony LiDAR sensor used in this work. The Sony LiDAR system played an essential role in enabling this research. This research was supported in part by JST Next-generation Edge AI Semiconductor JPMJES2515, JST CREST JPMJCR23M4, JST PRESTO JPMJPR22PA, JST ASPIRE JPMJAP2515, and JSPS KAKENHI 24K02940.

References

[1] Alpha Prime VLS-128. https://velodynelidar.com/products/alpha-prime/.

[2] ATX Hybrid Solid-State Lidar User Manual. https://www.hesaitech.com/wp-content/uploads/2026/02/ATX_User_Manual_A07-en-260210.pdf.

[3] HDL-32E Datasheet. https://www.mapix.com/wp-content/uploads/2018/07/97-0038_Rev-M_-HDL-32E_Datasheet_Web.pdf.

[4] HDL-64E Datasheet. https://www.mapix.com/wp-content/uploads/2018/07/63-9194_Rev-J_HDL-64E_S3_Spec-Sheet-Web.pdf.

[5] Huawei L106. https://www.techinsights.com/blog/deep-dive-teardown-huawei-96-line-lidar-l106-lidar-sensor.

[6] MMDetection3D. https://github.com/open-mmlab/mmdetection3d.

[7] openpilot. https://github.com/commaai/openpilot.

[8] Robosense RS-LiDAR-M1P User Guide. https://cdn.robotshop.com/media/U/UTR/RB-Utr-08/pdf/robosense-rs-lidar-m1-plus-advanced-2d-mems-smart-chip-lidar-200m-range-user-guide.pdf.

[9] Apollo Auto. Baidu Apollo. https://github.com/ApolloAuto/apollo.

[10] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020.

[11] Yulong Cao, S Hrushikesh Bhupathiraju, Pirouz Naghavi, Takeshi Sugawara, Z Morley Mao, and Sara Rampazzi. You Can't See Me: Physical removal attacks on LiDAR-based autonomous vehicles driving frameworks. In *Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23)*, pages 2993–3010. USENIX Association, 2023.

[12] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. Adversarial sensor attack on LiDAR-based perception in autonomous driving. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 2267–2281, 2019.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[14] Félix Goudreault, Dominik Scheuble, Mario Bijelic, Nicolas Robidoux, and Felix Heide. LiDAR-in-the-loop hyperparameter optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13404–13414, 2023.

[15] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019.

[16] Takami Sato, Yuki Hayakawa, Ryo Suzuki, Yohsuke Shiiki, Kentaro Yoshioka, and Qi Alfred Chen. LiDAR spoofing meets the new-gen: Capability improvements, broken assumptions, and new attack strategies. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2024.

[17] Takami Sato, Ryo Suzuki, Yuki Hayakawa, Kazuma Ikeda, Ozora Sako, Rokuto Nagata, Ryo Yoshida, Qi Alfred Chen, and Kentaro Yoshioka. On the realism of LiDAR spoofing attacks against autonomous driving vehicle at high speed and long distance. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2025.

[18] Dominik Scheuble, Hanno Holzhüter, Steven Peters, Mario Bijelic, and Felix Heide. LiDAR waveforms are worth $40 \times 128 \times 33$ words. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 28913–28924, 2025.

[19] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-voxel feature set abstraction for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10529–10538, 2020.

[20] Hocheol Shin, Dohyun Kim, Yujin Kwon, and Yongdae Kim. Illusion and dazzle: Adversarial optical channel exploits against LiDARs for automotive applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 445–467. Springer, 2017.

[21] Thorvald Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter*, 5:1–34, 1948.

[22] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020.

[23] Ryo Suzuki, Takami Sato, Yuki Hayakawa, Kazuma Ikeda, Ozora Sako, Rokuto Nagata, Ryo Yoshida, Qi Alfred Chen, and Kentaro Yoshioka. From lab to road: Realizing and detecting LiDAR spoofing attacks against autonomous vehicles at high speed and long distance. *IEEE Sensors Journal*, 25(13):25661–25681, 2025.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

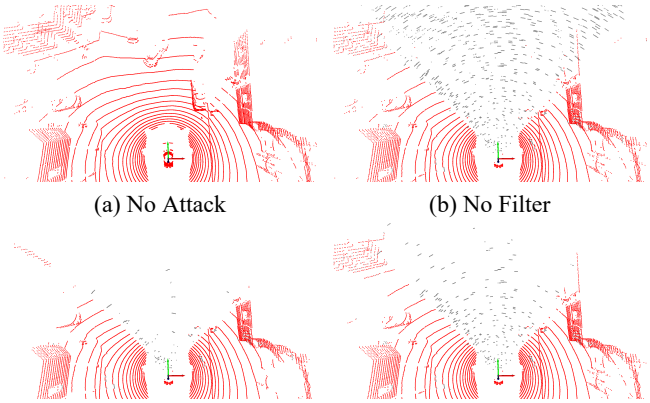
[25] Rongtian Ye, Fangyu Liu, and Liqiang Zhang. 3D depthwise convolution: Reducing model parameters in 3D vision tasks. In *Canadian Conference on Artificial Intelligence*, pages 186–199, 2019.

[26] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, 2021.

[27] Kentaro Yoshioka. A tutorial and review of automobile direct ToF LiDAR SoCs: Evolution of next-generation LiDARs. *IEICE Transactions on Electronics*, E105.C(10):534–543, 2022.

[28] Kentaro Yoshioka, Hiroshi Kubota, Tomonori Fukushima, Satoshi Kondo, Tuan Thanh Ta, Hidenori Okuni, Kaori Watanabe, Masatoshi Hirono, Yoshinari Ojima, Katsuyuki Kimura, et al. A 20-ch TDC/ADC hybrid architecture LiDAR SoC for 240×96-pixel 200-m range imaging with smart accumulation technique and residue quantizing SAR ADC. *IEEE Journal of Solid-State Circuits*, 53(11):3026–3038, 2018.

[29] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*, 2018.



(a) No Attack (b) No Filter

(c) Radius-based Outlier Filter (d) Statistical Outlier Filter

Figure 6: Point cloud denoising methods fail under jamming attacks. Red points indicate valid structures (neighbors within 5cm), and gray points indicate isolated points.

Appendix

A Attack Demonstration Video

An attack demonstration video showing PULSAR-Net's performance against an actual LiDAR jamming attack, as described in §6, is available on our project page. The video consists of driving scenarios (0-24s) and static scenarios (24-35s). Object detection bounding boxes from CenterPoint are overlaid to show downstream task performance.

The video demonstrates three key findings:

- **Attack severity**: The target vehicle completely disappears from the point cloud throughout the entire attack duration, confirming the jamming attack's persistent effectiveness across all frames
- **PULSAR-Net effectiveness**: Our method consistently recovers the vehicle structure in every frame during the reconstruction phase, demonstrating stable and reliable performance without intermittent failures
- **Detection recovery**: Object detection fails in almost 100% of frames under attack but achieves reconstruction success rate of 92% in static scenarios and 73% after PULSAR-Net reconstruction

This dynamic visualization effectively illustrates both the complete sensor blindness caused by jamming attacks and PULSAR-Net's robust recovery capability in real-world conditions.

B Limitations of Prior Point Cloud-Level Denoising Methods

As stated in §5.1, traditional point cloud denoising methods cannot recover objects under jamming attacks. Fig. 6 visualizes why these methods fundamentally fail.

We tested Radius Outlier Removal (ROR) and Statistical Outlier Removal (SOR) [29] on jammed point clouds. The figure colors points red if neighbors exist within 5cm (valid structure) and gray otherwise (isolated noise). Panel (a) shows no attack with dense red structures representing real objects. Panel (b) reveals that jamming attacks completely destroy local geometric coherence—virtually no red points remain in the attacked region. Panels (c) and (d) show that both ROR and SOR filters remove some noise but cannot reconstruct any object structure.

This complete information loss at the point cloud level validates our approach of leveraging full-waveform representations, where attack and legitimate signals remain distinguishable by their temporal signatures and simultaneous sensing patterns, despite spatial randomization. Unlike point-based

methods that operate on already-corrupted data, PULSAR-Net operates at the raw signal level, where legitimate reflections, though overwhelmed, still exist and can be separated from attack pulses.

C Training Details of PULSAR-Net

We train PULSAR-Net using AdamW with a weight decay of 5×10^{-2}, a learning rate of 3×10^{-4}, and a cosine annealing learning-rate schedule (CosineAnnealingLR). We use a batch size of 1 for 50 epochs without data augmentation. Training takes approximately 5.8 hours and uses 21,252 MiB of GPU memory. The model contains approximately 1.65 million parameters.

D Implementation details of Neural DSP [18] baseline

We reproduce Neural DSP [18] as an additional baseline for point cloud reconstruction. We reimplement its primary SWIN-based architecture with a Patch Tokenizer, while replacing the original Score and Offset classification heads with our segmentation head to ensure a fair comparison. As reported in the main paper, Neural DSP generally underperforms PULSAR-Net by several tens of percentage points in mAP and may collapse under certain settings, particularly at $\phi = 225$ on the KITTI-J dataset.

The main bottleneck of Neural DSP arises from its low-resolution Patch Tokenization. Our target LiDAR waveforms have large spatial dimensions, e.g., $(32, 1800, 800)$, and directly tokenizing them into Transformer inputs of $(40, 128, 33)$ inevitably discards significant fine-grained waveform information. In particular, horizontal waveform correlations are heavily compressed, eliminating important cues for distinguishing real objects from malicious jamming signals. Although increasing the patch resolution could alleviate this problem, it would dramatically increase the token sequence length, leading to quadratic growth in memory usage and computational cost. For waveforms of $(32, 1800, 800)$, further increasing the tokenizer resolution becomes infeasible due to GPU memory constraints even on an RTX 5090. In contrast, PULSAR-Net explicitly focuses on efficient architectural design to capture fine-grained waveform structures while reducing computational overhead via DWSeparable 3D blocks, achieving a strong balance between performance and efficiency.

E Additional Ablation Study

Impact of Channel Dimension: The hidden channel dimension (C in Fig. 3) reflects the model's representational capacity, with larger values generally improving its ability to capture complex waveform structures. We examine segmentation performance under different channel dimensions, and the results in Table 7 show steady improvement as C increases from 16 to 32. Meanwhile, the FPS measured on an RTX 5090 GPU drops from 69.93 to 31.74, it remains well above the threshold required for real-time processing. Hence, we selected $C = 32$ for the denoising model as an optimal trade-off between accuracy and efficiency.

Impact of Simultaneous Laser Sensing: We investigate the impact of simultaneous laser sensing. Our 3D convolution approach (`3D-CONV`) is designed to capture not only temporal information in the full waveform representation but also spatial information in simultaneous laser sensing. To highlight the impact of simultaneous laser sensing, we compare our approach with a 1D convolution along with the spatial axis.

Table 8 shows the performance comparison between `1D-CONV` and `3D-CONV`. As shown, `3D-CONV` shows significantly higher performance than `1D-CONV`, meaning modeling waveform shapes solely along the temporal axis is insufficient to differentiate object peaks from jamming attack's ones. Even though the full-waveform representation contains both legitimate and attack pulses, it is hard to remove the attack effects solely with the full-waveform representation. The simultaneous laser sensing plays an essential role in identifying the attack pulses as expected in §2.3. Furthermore, convolution along the spatial axis can achieve significant efficiency gains, as it acts as downsampling in the spatial dimensions. In Table 8, it enables >70 times faster inference.

Waveform-Level Segmentation Metrics: While the main text evaluates downstream performance, point-level recovery and object detection mAP, this appendix section directly measures PULSAR-Net's segmentation quality in the full-waveform domain. Since PULSAR-Net performs three-class

Table 7: Impact of channel dimensions (C) on PULSAR-Net performance.

C	Point Cloud level: Acc.	Detection level: mAP	FPS
16	81.03	61.86	**69.93**
24	80.66	63.05	41.66
32	**83.17**	**63.26**	31.74

Table 8: Comparison of performance between `1D-CONV` and `3D-CONV`.

Model	Point Cloud level: Acc.	Detection level: mAP	FPS
`1D-CONV`	2.57	0.17	0.43
`3D-CONV`	**83.17**	**63.26**	**31.74**

segmentation (background, legitimate reflection, attack pulse) on the full-waveform tensor, evaluating segmentation accuracy at this level provides a more precise understanding of the model's internal behavior.

We report IoU separately for legitimate object reflections (*Object IoU*) and attack pulses (*Attack IoU*). Object IoU measures how well PULSAR-Net preserves legitimate full-waveform returns, while Attack IoU measures how accurately it identifies malicious jamming pulses.

Table 9 shows that `3D-CONV` achieves substantially higher IoU for both legitimate object pulses and attack pulses than `1D-CONV`. This indicates that temporal information alone is insufficient and that incorporating spatial correlations from simultaneous laser sensing is essential for accurate waveform-level segmentation.

Table 9: Waveform-level segmentation comparison between `1D-CONV` and `3D-CONV`. The `3D-CONV` achieves higher IoU for both legitimate object pulses and attack pulses.

Model	**Object IoU**	**Attack IoU**
`1D-CONV`	0.51	0.066
`3D-CONV`	**0.548**	**0.910**

Table 10: Dataset generation parameters

Parameter	**Value**
Attack interval δ_{attack}	100 ns (10 MHz)
Attack amplitude γ_A	[3.0, 8.0]
Timing jitter ϵ_k	$\mathcal{U}(-20, 20)$ ns
Pulse scaling γ (KITTI)	12.0
Pulse scaling γ (nuScenes)	0.156
Time duration T_{max}	800 ns

F Full Experimental Results for Object Detection

Table 11 provides the comprehensive performance evaluation of PULSAR-Net and all baselines across all evaluated simultaneous sensing configurations on both nuScenes-J and KITTI-J datasets, revealing that PULSAR-Net consistently outperforms baseline methods across all configurations where simultaneous sensing is available ($\phi > 1$). Within this range, regardless of the number of simultaneous lasers, our method consistently maintains high mAP, demonstrating the robustness of our approach to leveraging simultaneous sensing information.

G Details of Dataset Generation

We provide details of the setup used in our full-waveform synthesis pipeline, designed to create realistic and challenging attack scenarios that encompass and extend beyond current state-of-the-art jamming attacks.

G.1 Parameter Configuration

Table 10 summarizes the key parameters used for dataset generation. These parameters were carefully selected to create a comprehensive threat model that surpasses existing jamming attacks.

To ensure that our reconstructed model remains robust under worst-case conditions, we design a jamming model whose strength surpasses that of the strongest known attacks (HFR [16] and A-HFR [17]). By incorporating substantially higher attack frequency, randomized amplitude modulation, and non-deterministic temporal jitter in our dataset generation pipeline, our threat model generalizes beyond existing HFR and A-HFR attack and represents a strictly stronger adversarial setting. Specifically, the

Table 11: *Object detector-level mAP* of PULSAR-Net (Ours) and baselines on the nuScenes-J and KITTI-J datasets. ϕ denotes the number of simultaneous sensing lasers. Results for learned methods are reported as mean $\pm$ standard deviation over three training runs with different random seeds; deterministic baselines are evaluated once.

		nuScenes-J (mAP)		**KITTI-J (mAP)**	
ϕ	**Method**	PointPillars	CenterPoint	PointPillars	PV-RCNN
1	No Defense	0.16	0.00	0.00	0.00
	Naive Average Sub.	-	-	-	-
	Neural DSP [18]	0.33 ± 0.08	0.09 ± 0.12	0.00 ± 0.00	0.00 ± 0.00
	Ours	$\mathbf{1.14 \pm 0.59}$	$\mathbf{0.37 \pm 0.48}$	$\mathbf{17.63 \pm 2.22}$	$\mathbf{24.22 \pm 1.98}$
4	No Defense	0.01	0.00	0.00	0.00
	Naive Average Sub.	1.23	0.00	7.23	20.37
	Neural DSP [18]	0.18 ± 0.03	0.01 ± 0.01	1.31 ± 0.41	1.44 ± 0.29
	Ours	$\mathbf{60.42 \pm 0.77}$	$\mathbf{55.23 \pm 0.62}$	$\mathbf{43.50 \pm 0.56}$	$\mathbf{54.75 \pm 0.44}$
5	No Defense	0.01	0.00	0.00	0.00
	Naive Average Sub.	1.54	0.00	10.61	25.62
	Neural DSP [18]	0.20 ± 0.13	0.04 ± 0.05	2.84 ± 0.42	3.40 ± 0.63
	Ours	$\mathbf{63.14 \pm 0.52}$	$\mathbf{58.66 \pm 0.03}$	$\mathbf{46.96 \pm 0.90}$	$\mathbf{57.09 \pm 0.23}$
10	No Defense	0.05	0.02	0.00	0.00
	Naive Average Sub.	1.12	0.00	14.69	33.34
	Neural DSP [18]	8.54 ± 7.14	14.44 ± 8.10	3.42 ± 0.05	5.61 ± 0.98
	Ours	$\mathbf{64.55 \pm 0.14}$	$\mathbf{59.06 \pm 0.43}$	$\mathbf{48.16 \pm 0.42}$	$\mathbf{56.45 \pm 0.47}$
25	No Defense	0.39	0.08	0.00	0.00
	Naive Average Sub.	1.94	0.00	21.81	41.06
	Neural DSP [18]	22.07 ± 4.87	27.92 ± 2.99	4.20 ± 2.22	6.11 ± 3.04
	Ours	$\mathbf{64.19 \pm 0.38}$	$\mathbf{59.23 \pm 0.22}$	$\mathbf{48.22 \pm 0.14}$	$\mathbf{56.66 \pm 0.71}$
55	No Defense	0.95	0.48	0.00	0.00
	Naive Average Sub.	2.98	0.72	21.66	38.06
	Neural DSP [18]	40.29 ± 5.27	43.22 ± 2.05	2.41 ± 1.69	3.27 ± 1.97
	Ours	$\mathbf{64.20 \pm 0.59}$	$\mathbf{59.08 \pm 0.43}$	$\mathbf{48.07 \pm 0.29}$	$\mathbf{55.65 \pm 0.45}$
110	No Defense	0.95	0.32	0.00	0.00
	Naive Average Sub.	2.56	4.04	16.35	32.94
	Neural DSP [18]	44.26 ± 6.11	48.31 ± 1.95	2.65 ± 1.83	4.61 ± 2.79
	Ours	$\mathbf{65.41 \pm 0.31}$	$\mathbf{58.68 \pm 0.28}$	$\mathbf{47.86 \pm 0.24}$	$\mathbf{56.01 \pm 0.37}$
225	No Defense	0.77	0.49	0.00	0.00
	Naive Average Sub.	4.09	9.11	0.14	2.34
	Neural DSP [18]	40.41 ± 8.60	46.77 ± 3.61	2.98 ± 2.36	4.59 ± 3.46
	Ours	$\mathbf{64.26 \pm 0.17}$	$\mathbf{58.91 \pm 0.29}$	$\mathbf{49.19 \pm 0.22}$	$\mathbf{56.26 \pm 0.53}$
	No Attack	69.87	72.72	57.51	62.75

10 MHz attack interval ($\delta_{\text{attack}} = 100$ ns) introduces far more pulses per LiDAR scan than existing attacks. Furthermore, the randomized amplitude $\gamma_A \in [3.0, 8.0]$ prevents simple threshold-based filtering while maintaining sufficient signal strength to overwhelm legitimate returns. Each pulse receives an independent random amplitude within this range, eliminating exploitable patterns. The temporal jitter $\epsilon_k \sim \mathcal{U}(-20, 20)$ ns applied to each pulse disrupts periodic patterns that could enable frequency-domain filtering. This 40 ns jitter window is sufficient to prevent coherent cancellation techniques while maintaining pulse integrity.

G.2 LiDAR Signal Calibration

LiDAR Intensity Scaling: The pulse amplitude scaling factors γ were calibrated separately for each dataset to account for differences in their LiDAR intensity representations: KITTI uses normalized intensity values in [0, 1], while nuScenes uses 8-bit values in [0, 255]. This scaled intensity directly represents the amplitude of the legitimate pulse within the full LiDAR waveform.

The selected scaling factors (12.0 for KITTI, 0.156 for nuScenes) satisfy two critical constraints: maintaining $\geq$95% attack success rate while preserving sufficient legitimate pulse amplitude to prevent trivial intensity-based classification. Lower values would make defense unrealistically simple, while higher values would completely mask legitimate returns. Specifically, the average original intensities are 0.254 for KITTI and 19.23 for nuScenes, resulting in a mean legitimate pulse amplitude of approximately 3.0 for both datasets after scaling. This overlaps with the attack amplitude range of [3, 8] (mean 5.5), explaining why attack pulses can disrupt legitimate returns without necessarily exceeding their maximum amplitude.

Scan Duration of Full Waveform: Time duration of the scan $T_{\text{max}} = 800$ ns corresponds to the temporal dimension D, which translates to a maximum sensing range of 120 meters (using $d = c \cdot T/2$). This range matches the valid detection distance of the HDL-64E used in KITTI, while exceeding the 100m range of the HDL-32E in nuScenes, ensuring complete coverage of all legitimate returns. Our U-Net-inspired architecture can generalize to longer temporal durations without retraining, as the convolutional operations naturally extend to different input sizes. This design choice ensures that PULSAR-Net can adapt to various LiDAR configurations with different maximum ranges, making it practical for deployment across diverse sensor specifications.

H Additional Discussions and Limitations

False Positive Analysis: PULSAR-Net predicts a mask to remove attack-induced returns rather than generating new waveform returns, so it does not synthesize ghost geometry. However, attack pulses that escape masking may still produce false detections. We did not observe prominent false positives in our real-world demonstration, but we do not provide a systematic false-positive analysis for driving scenarios. While the impact of false positives is included in the mAP metric, this study does not validate PULSAR-Net's usability in actual driving scenarios. Meanwhile, we expect our PULSAR-Net to be used with an attack detection algorithm. As demonstrated in [23], state-of-the-art jamming attack detection achieves $\geq$90% success rates even while the car is traveling at cruising speeds.

Practical Deployment Considerations: An attack-detection module should first detect a jamming attack, and PULSAR-Net should be activated only when such an attack is detected to avoid unnecessary computation during normal driving. However, detecting the presence of an attack does not itself ensure safe autonomous driving. Once an attack is detected, the vehicle still needs to maintain a safe trajectory; for example, slowing down or navigating to a secure area, especially during highway driving. In these situations, PULSAR-Net provides meaningful LiDAR sensing information even when the raw point cloud is unreliable, enabling safer decision-making under severe jamming attacks.

From a computational perspective, a remaining challenge for practical deployment is the inference latency on edge devices. While PULSAR-Net achieves 31.74 FPS on an RTX 5090, we estimate approximately 3.6 FPS (0.28 s per frame) on Jetson AGX Orin based on the memory bandwidth ratio between the two devices (1,792 vs. 204.8 GB/s). This estimate assumes that inference is primarily memory-bandwidth-bound and has not been verified on actual edge hardware. Since the estimated throughput is below the typical perception rate of 10–20 FPS, further optimization is necessary for real-time deployment on edge devices.

Limitations of our Real-World Evaluation: We quantitatively evaluate reconstruction performance only for vehicles, although the demonstration video on our project page shows that PULSAR-Net also reconstructs points from pedestrians, cyclists, and guardrails under attack. These observations suggest that the reconstruction mechanism is not specific to vehicles, while quantitative evaluation of other object categories remains future work. Furthermore, we evaluate moving scenarios only under attack conditions, as reproducing identical driving trajectories without attack is impractical. Consequently, we do not directly compare reconstructed performance with matched benign performance in these scenarios.

Selective Attack in Simultaneous Sensing. PULSAR-Net assumes that attackers cannot inject different pulse patterns within the same simultaneous laser sensing. This assumption generally holds because it requires aiming multiple lasers with beam diameters $\leq$1 mm at a moving LiDAR from distances like 100 m. Detecting and targeting a LiDAR at such distances is already challenging; accurately directing multiple thin beams simultaneously is even less feasible in practice.

Generality to Other LiDARs. PULSAR-Net requires full-waveform data and simultaneous laser sensing. Full-waveform data should be available in most LiDAR as it is directly received by their single-photon avalanche diodes. While simultaneous sensing may not be available, it is commonly used to achieve multiple vertical scan channels with no major technological barriers. For example, as shown in the Table 1, modern multi-channel LiDAR systems already scan points simultaneously. PULSAR-Net is thus applicable to most modern LiDARs, and we advocate that manufacturers should provide access to full-waveform data to enable more secure LiDAR applications.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims regarding the feasibility of LiDAR point cloud reconstruction under jamming attacks using full-waveform data and simultaneous sensing are clearly stated in Section 1 and validated through extensive experiments in Sections 5 and 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss several limitations in Section 7.7 (Additional Discussions and Limitations), including the lack of false-positive analysis in driving scenarios and the necessity of pairing PULSAR-Net with an attack detection module for practical deployment.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper is primarily empirical and does not present new theoretical theorems or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide comprehensive details on the network architecture, dataset generation pipeline, hyperparameters, and source code for PULSAR-Net in Sections 3 and 4, the Appendix (A.6), and the supplementary materials, which are sufficient for reproducing our experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The source code for PULSAR-Net is included in the supplemental material for review and will be made publicly available upon publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental settings, including LiDAR specifications, data split details, and training configurations, are specified in Section 5.1 and expanded upon in Appendix A.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report performance across multiple metrics (point-level accuracy and mAP) and include evaluations on various simultaneous sensing configurations (ϕ) in Tables 2, 3, and 11 to show consistent statistical trends.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources used for training and inference, specifically the NVIDIA RTX 5090 GPU and the resulting FPS, are reported in Sections 5.2 and Appendix A.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research, including the real-world evaluation involving a partner supplier, conforms to these guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: In Section 1 and Section 7, we discuss the positive impact on autonomous vehicle safety and acknowledge the potential risk of our vulnerability analysis being used to refine future attacks.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk models like large language models or image generators that require specific gated release safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly credit and cite the KITTI and nuScenes used in our study in Section 3 and the References section.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We describe the PULSAR-Net architecture and the physics-based synthesis pipeline in Sections 3 and 4; the associated code is provided in the supplemental material.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our research does not involve crowdsourcing or experiments with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Our research does not involve human subjects and therefore does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as a core method, original component, or for non-standard development in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.