Dual-Supervised Deep Learning Approach for Automotive Radar Spoofing Detection

J. Chen, I. Stainvas, and I. Bilik

School of Electrical and Computer Engineering
Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

***Abstract*— This paper presents a multi-radar spoofing detection framework based on PointNet++ and Plug-and-Play (PnP) point cloud registration. The method first extracts hierarchical geometric features of multi-radar point clouds using PointNet++, which are embedded into PnP optimization to achieve robust cross-radar alignment. On this basis, we propose DB-PointNet++, which combines DBSCAN clustering and density-aware feature enhancement to construct feature-rich point cloud representations for discriminating genuine and spoofed data. Experimental results on a simulated dataset demonstrate that the framework achieves an accuracy of about 92% in binary classification (genuine vs. spoofed), with an ROC AUC of 0.836, and reaches TPR=88.9% and FPR=34.1% at the optimal operating point. In comparison, the four-class task (genuine, A/B/C spoofed) yields an overall accuracy of 70.5%, where Genuine and B-spoofed achieve the highest recognition rates (82% and 78%), while A- and C-spoofed remain confusable. Further validation on the nuScenes real-world dataset shows performance degradation due to sparse radar points: binary classification accuracy drops to about 52% with an AUC of 0.680; in four-class detection, B-spoofed achieves an AUC of 0.751, whereas A- and C-spoofed only reach 0.672 and 0.701. Overall, the proposed framework demonstrates strong robustness in high-resolution scenarios. Binary detection is more suitable for real-time safety-critical deployment, while four-class classification, though less accurate, provides spoof-source attribution and lays the foundation for further multi-sensor fusion and temporal-consistency enhancements.**

***Index Terms*— Autonomous driving, Automotive radar, Point cloud registration, Spoofing detection, Dual-supervised learning, Radar dataset.**

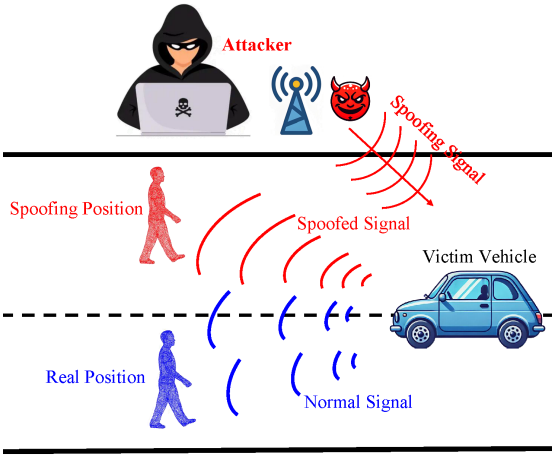


Figure 1: Schematic illustration of the considered scenario in which a malicious adversary attacks the victim's automotive radar by injecting spoofing signals. As a result, the radar detects a false pedestrian position (marked in red), while the actual pedestrian is located elsewhere (marked in blue).

I. Introduction

Sensing suites of autonomous and advanced driver-assistance systems (ADAS)-equipped vehicles include cameras, LiDARs, and radars [1]–[5]. Radars provide robustness in adverse weather and poor lighting conditions, long operation ranges, and direct velocity measurements [6]–[8]. Therefore, their performance is critical for the success of any automotive application [9]–[11]. The reliance of autonomous driving (AD)/ADAS vehicles on sensing capabilities introduces inherent vulnerabilities to malicious threats [12]. While most prior security research has concerned conventional cybersecurity [13], the sensing layer exposes Automated Driving System (ADS)/ADAS to additional threats that extend beyond software and network domains [14]. Classical physical attacks on sensors, such as denial-of-service (jamming) [15], [16] and deception (spoofing and false-data injection) [17], [18], have been widely studied [15]. It was demonstrated that an attacker can compromise perception, degrade vehicle control, and even cause severe accidents [19]. In response, multiple detection and mitigation strategies against jamming and spoofing in automotive radars have been proposed [15], [17].

The recent adoption of deep neural network (DNN)-based processing [20], [21] for cameras, LiDARs [22], and more recently for radars [23]–[26] has also introduced a new class of vulnerabilities, where small perturbations to the input data can result in large and unexpected output errors [27]–[29]. For radar perception, DNNs have significantly improved target detection [30]–[33], parameter estimation [34], [35], tracking [31], [36], and classification performance [37]–[40]. Therefore, protecting these DNN-based radar functions against adversarial attacks is critical and urgent for the safe deployment of radar-enabled ADAS/AD applications.

Adversarial attacks on cameras [41], [42] and LiDARs [43], [44], where attackers manipulate physical-world objects or access the DNN data, have been recently extensively studied in the literature [29], [41], [42], [45]. However, adversarial threats to radar perception remain largely unexplored [27], [46], [47], leaving a critical gap in ADS/ADAS security. Unlike conventional jamming or spoofing, which typically generate interference signals detectable by well-studied radar electronic countermeasures (ECCM), adversarial perturbations can remain covert, because they are embedded within the expected signal structure. Such perturbations might not trigger interference alarms, making them stealthier and potentially more dangerous than conventional spoofing and jamming attacks.

Autonomous and ADAS-equipped vehicles typically utilize multiple radars with overlapping fields of view [9]. This redundancy allows cross-validation of radar measurements, providing opportunities for detecting deceptive attacks [48], [49]. Statistical analysis and machine learning (ML)-based approaches for detecting classical deceptive attacks have been proposed in the literature, evaluating inconsistencies across radar measurements [50]. However, due to the complexity of automotive environments, dynamic object motion, and inherent measurement noise and uncertainty, these conventional methods often struggle, especially against precisely crafted deception attacks involving target position and orientation [51]–[53].

Addressing these challenges, this paper proposes an innovative radar spoofing detection method based on deep learning, leveraging inconsistencies among multi-radar sensor measurements. A realistic

autonomous vehicle scenario is simulated using the Unity software platform [54], generating a comprehensive dataset containing both authentic and spoofed radar-generated point clouds. Building upon this dataset, we introduce the DB-PointNet++ deep learning framework, which combines the PointNet++ network architecture [55]–[57] for point cloud feature extraction with density-based spatial clustering of applications with noise (DBSCAN) [58], [59]. The DB-PointNet++ model employs a dual-supervised learning mechanism, capturing both local geometric details and global spatial consistency among radar measurements, thereby accurately identifying anomalies induced by spoofing attacks. While PointNet++ has been widely utilized in computer vision for 3D object recognition and point cloud segmentation, to our knowledge, this research represents its first application to automotive radar spoof detection.

Furthermore, this paper develops and releases a multi-radar spoofing simulation dataset that incorporates diverse deceptive scenarios and radar resolutions. This dataset enables systematic analysis of how spoofing intensity and radar resolution affect detection performance, and serves as a valuable benchmark for future research on radar spoofing defense.

The main contributions of this work are summarized as follows:

- We propose a novel PointNet++ & Plug-and-Play (PnP) based registration framework that enables robust cross-radar point cloud alignment and feature extraction, effectively addressing the limitations of conventional ICP-based methods.
- We design DB-PointNet++, which integrates DBSCAN clustering and density-aware feature enhancement with PointNet++, constructing a feature-enriched point cloud representation for improved spoofing detection.
- We conduct extensive experiments on a simulated dataset, where the framework achieves about 92% accuracy in binary classification (area under the curve (AUC)=0.836) and 70.5% in four-class classification, demonstrating both strong detection capability and spoof-source attribution.
- We validate the approach on the nuScenes real-world dataset, revealing performance degradation due to sparse radar measurements (binary accuracy ≈52%, AUC=0.680). These results highlight the challenges of real-world deployment and provide insights for future multi-sensor fusion and temporal-consistency enhancements.
- We release a high-fidelity, multi-radar simulation dataset generated with the Unity engine, covering displacement- and rotation-based spoofing scenarios across multiple resolutions, which can serve as a benchmark for subsequent research.

The findings of this research are expected to advance the development of robust defenses against radar sensor spoofing attacks, thereby contributing to safer and more reliable autonomous driving systems.

The remainder of this article is organized as follows. The problem addressed in this work is introduced in Section II. Section III introduces the proposed DB-PointNet++ multi-radar spoofing detection approach. The generated dataset of radar point clouds is introduced in Section IV. The performance of the proposed multi-radar spoofing detection approach is evaluated in Section V. Our conclusions are summarised in Section VI.

II. Problem Definition

In modern automotive radar systems, environmental perception relies heavily on high-resolution radar point clouds [60], [61]. However, despite their advanced sensing capabilities, such radar sensors remain susceptible to targeted physical-layer spoofing attacks. These malicious attacks deliberately manipulate the sensor's perception of the environment, thereby severely degrading radar performance and potentially causing accidents due to erroneous environmental interpretation.

In this study, we address the problem of detecting radar spoofing attacks through the deployment of a multi-radar sensor system. Specifically, we investigate a scenario depicted in Fig. 1, in which an attacker injects falsified information regarding object positions, orientations, and extents into one of the radar sensors, thereby corrupting the point cloud data generated by that sensor.

We focus on a representative automotive scenario as shown in Fig. 2, in which the vehicle is equipped with three strategically positioned radars to ensure comprehensive environmental coverage from multiple viewpoints. These radars are mounted at the front roof area and arranged with a 90° angular separation, forming a robust and redundant sensing configuration. Consequently, the radar directly facing the attacker receives manipulated signals and produces distorted point cloud data. In this scenario, only one radar is subjected to malicious signal injection at any given time, while the other two radars acquire accurate and unaltered point cloud representations of the scene. Moreover, the vast majority of the scene's point cloud remains unaffected in terms of position or orientation; the attack selectively targets points corresponding to pedestrians or vehicles, introducing position offsets or angular rotations. Such adversarially manipulated data exhibit significant inconsistencies when compared to the outputs of the two unaffected radars.

The vehicle coordinate frame is denoted as $\mathcal{F}_V$. Three radars $i \in \{M, R, L\}$ are mounted near the front roof area of the vehicle, forming a right-angle layout: the lines connecting the Right and Left radars to the Middle radar satisfy $\angle(t_R - t_M,\, t_L - t_M) = 90^\circ$. Each radar has calibrated extrinsic parameters $\mathbf{T}_i^V = [\mathbf{R}_i^V \mid \mathbf{t}_i^V] \in SE(3)$, mapping radar coordinates to the vehicle frame:

$$\mathbf{x}^V = \mathbf{R}_i^V\, \mathbf{x}^{(i)} + \mathbf{t}_i^V. \tag{1}$$

At time step k, radar i returns a point set $\mathcal{P}_i = \{\mathbf{p}_{i,n}\}$. After instance-level clustering (pedestrian, vehicle, etc.), we obtain an object set $\mathcal{O}$, where instance o observed by radar i is represented by centroid $\boldsymbol{\mu}_i^o$, covariance $\boldsymbol{\Sigma}_i^o$, and radar-specific measurements such as radial velocity $v_{r,i}^o$ and RCS γ_i^o.

Under static or quasi-static conditions, the representation of the same object in the vehicle frame should be consistent:

$$\underbrace{\mathbf{m}_i^o}_{\text{in vehicle frame}} = \mathbf{R}_i^V \boldsymbol{\mu}_i^o + \mathbf{t}_i^V, \quad \mathbf{m}_i^o \approx \mathbf{m}_j^o, \;\; \forall i \neq j,\; o \in \mathcal{O}. \tag{2}$$

Likewise, θ_i^o denotes the geometric orientation of object o, $\theta_i^o \approx \theta_j^o$, $\mathbf{d}_i^o \approx \mathbf{d}_j^o$, and after compensating ego-motion $\mathbf{v}_{ego}$, the projected velocities u_i^o should match.

In each attack instance, only one radar $a \in \{M, R, L\}$ is compromised. The attacker manipulates a subset $\Omega \subset \mathcal{O}$ (mainly pedestrians/vehicles) via a rigid transformation $\mathbf{A} \in SE(3)$ (position displacement/rotation), while leaving the majority of the background points unchanged:

$$\boldsymbol{\mu}_a^o \leftarrow \mathbf{A}\, \boldsymbol{\mu}_a^o, \quad \theta_a^o \leftarrow \theta_a^o + \Delta\theta, \quad \mathbf{d}_a^o \leftarrow \mathbf{d}_a^o + \Delta\mathbf{d}, \quad o \in \Omega. \tag{3}$$

The other two radars $i \neq a$ and the unaffected objects $o \notin \Omega$ follow H_0. This induces noticeable cross-view inconsistencies for the attacked radar, concentrated on specific semantic classes.

After transforming each instance observation into the vehicle frame:

$$\mathbf{m}_i^o = \mathbf{R}_i^V \boldsymbol{\mu}_i^o + \mathbf{t}_i^V, \quad \theta_i^o = \theta_i^o, \quad \mathbf{D}_i^o = \mathbf{d}_i^o, \quad u_i^o = v_{r,i}^o - (\mathbf{R}i^V)\mathbf{v}ego \cdot \hat{n},$$

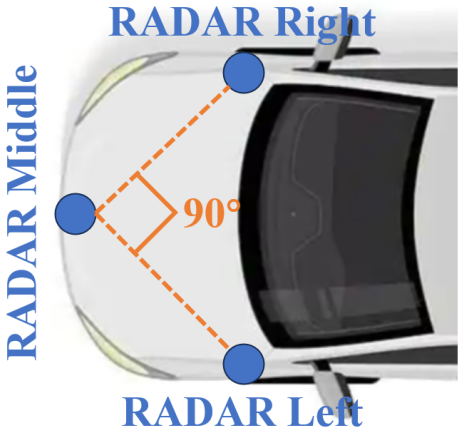


Figure 2: The figure shows a three-radar configuration on the vehicle roof, including a front-middle radar (radar middle) and two lateral radars (radar right and radar left). The angle formed by the lines connecting the two lateral radars to the central radar is 90°, resulting in a right-angle layout. This arrangement facilitates cooperative detection by multiple radars, significantly improving the accuracy and coverage of object detection through the fusion of radar data from different perspectives. Consequently, it reduces blind-spot risks, making it particularly suitable for ADAS.

define for any two radars $i \neq j$ and instance o the vector residual:

$$\mathbf{r}_{ij}^{o} = \begin{bmatrix} \mathbf{m}_i^o - \mathbf{m}_j^o \\ \alpha_\theta \, \mathrm{wrap}(\theta_i^o - \theta_j^o) \\ \alpha_d \, (\mathbf{D}_i^o - \mathbf{D}_j^o) \\ \alpha_v \, (u_i^o - u_j^o) \end{bmatrix},$$

where $\mathrm{wrap}(\cdot)$ maps angles to $(-\pi, \pi]$ and α_* are scaling weights, which are determined based on the magnitude of each quantity and empirical validation experiments, balancing the contribution of different features to the consistency metric. Given the measurement noise covariance $\mathbf{\Sigma}_{ij}^{o}$, define the Mahalanobis consistency score:

$$s_{ij}^{o} = (\mathbf{r}_{ij}^{o})^{\top} (\mathbf{\Sigma}_{ij}^{o})^{-1} \mathbf{r}_{ij}^{o}. \tag{4}$$

Under H_0 and Gaussian noise, $s_{ij}^{o} \sim \chi_d^2$, where d is the dimension of $\mathbf{r}$.

Aggregate the pairwise scores for each radar:

$$S_i = \sum_{j \neq i} \sum_{o \in \mathcal{O}_{ij}} w_o \, s_{ij}^{o}, \tag{5}$$

where $\mathcal{O}_{ij}$ is the set of instances observed by both i and j, and w_o are confidence/point-count weights. Under the single compromised radar assumption, the attacked radar a satisfies $S_a \gg S_b$, $b \neq a$:

$$\hat{a} = \arg \max_{i \in \{M,R,L\}} S_i, \quad \text{declare attack if } S_{\hat{a}} > \tau. \tag{6}$$

The threshold τ can be set via χ^2 approximation or validation to control the false alarm rate α; robust statistics can be applied to reduce occlusion-related outliers.

To mitigate the detrimental effects of radar spoofing attacks, we propose an innovative neural network–based detection approach specifically designed to identify adversarially manipulated point cloud data. Furthermore, the proposed method exploits radar-specific metadata embedded within the point clouds to determine which radar sensor has been compromised. Our approach is grounded in the principle of multi-view spatial consistency, enabling the extraction of robust features from radar point clouds for reliable authenticity assessment. By leveraging cross-view discrepancies among multiple radars, the proposed technique substantially enhances the radar system's resilience to sophisticated spoofing attacks, thereby contributing to safer autonomous driving environments.

III. The Proposed Approach

This section introduces DB-PointNet++, a dual-supervised framework for spoofed radar point clouds detection in multi-radar systems. It combines point-level supervision via DBSCAN and cluster-level supervision via PointNet++, achieving enhanced detection accuracy and robustness in complex scenarios.

Fig. 3 shows the processing flow of the proposed spoofing detection approach, containing three major components. First, the *Registration* block, detailed in Subsection A, aligns point clouds obtained from radars "A" and "C", PC_A and PC_C, to the radar "B" coordinate system and merges them with the PC_B to create a unified point cloud, PC. Second, the *Heuristic spoofing Detection* block, detailed in Subsection B, pre-labels points as "real" or "spoofed" based on cross-radar consistency, generating labeled point cloud PC'. Then the *Clustering* block partitions PC' into k sub-clusters using DBSCAN, followed by the *Density Compute* block that appends local density features to create augmented clusters, $PC_1'', \ldots, PC_k''$. Finally, the *Merge* block consolidates all sub-clusters into PC'', which is classified by the third, *PointNet++ Classification* block, to identify which radar is spoofed, as detailed in Subsection C.

A. Point Clouds Registration

Point cloud registration is a fundamental challenging task, particularly in the presence of noise, occlusions, and incomplete data [62]. As illustrated in Fig. 3, this subsection describes the first processing module, which aims to achieve robust global alignment for complex point clouds. To address these challenges, we propose a fusion-based framework that integrates deep feature extraction through PointNet++ with a probabilistic PnP optimization strategy [63]–[65]. Specifically, the PnP framework incorporates a learned denoising module—implemented via PointNet++—to implicitly encode prior knowledge of geometric structures. This design enables the registration process to benefit from both explicit data-driven regularization and feature-level guidance, thereby substantially enhancing robustness against various types of degradation. In scenarios where one of the three radars is compromised, the corresponding point cloud may contain anomalous or abruptly appearing spoofed points. Within the proposed framework, the registration procedure prioritizes the geometric regions jointly and consistently observed by all radars, thereby anchoring the alignment to structurally reliable features. Consequently, the anomalous observations contributed by the attacked radar are transformed together with the global rigid motion, yet they exert negligible influence on the final registration outcome.

Despite its widespread adoption, the iterative closest point (ICP) [66]–[68] algorithm suffers from several fundamental limitations that restrict its applicability in challenging scenarios. It is highly sensitive to initial alignment, often converging to local minima and resulting in suboptimal registration accuracy. Moreover, because ICP relies solely on nearest-neighbor correspondences, it fails to capture higher-level structural or semantic features within the point cloud, making it vulnerable to noise, outliers, and partial observations [69]. The absence of global or topological constraints further diminishes its robustness, especially when dealing with large-scale or structurally complex data. These issues collectively highlight the necessity for more advanced registration frameworks that can incorporate prior knowledge and adapt to varying data quality.

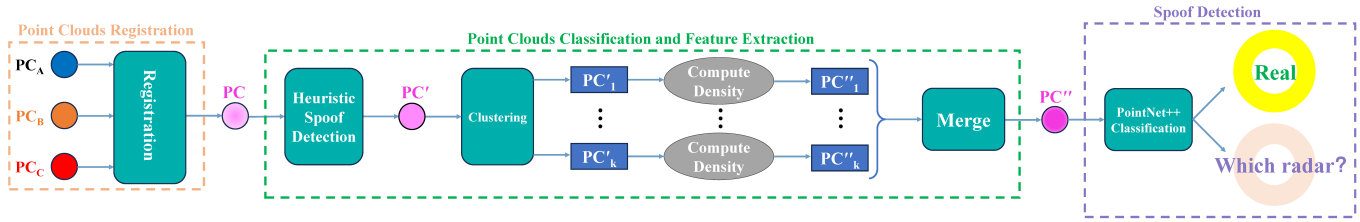


Figure 3: Schematic representation of the proposed DB-PointNet++ for radar point cloud processing. First, the point clouds from radars "A" and "C" are registered to the coordinate system of radar "B" and merged into a unified point cloud. A heuristic-based spoofing detection module then preliminarily labels each point as "spoofed" or "real" based on its spatial relationship with points from the other radars. Next, DBSCAN is used to cluster all detections, and each cluster is analyzed to compute a density feature for every point within it. These features are appended to the original point features to form an augmented merged point cloud, which is finally classified by PointNet++ into "spoofed" or "real".

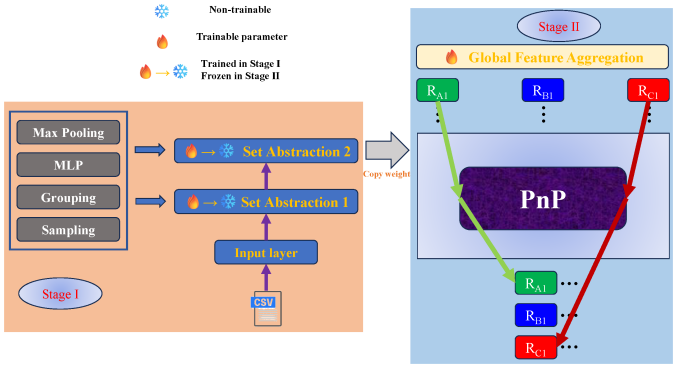


Figure 4: Schematic representation of point cloud registration, performed in two stages. In *Stage I*, PointNet++ is trained to classify radar scenes into object categories. The latent space of PointNet++ is then used as an additional feature for the PnP algorithm. In *Stage II*, the weights obtained from PointNet++ in Stage I are "frozen," and a global feature aggregation module generates high-level feature representations. These features are subsequently employed as prior knowledge to guide the denoising optimization of each point and are used by the PnP algorithm for point cloud registration. The registration process iteratively refines the alignment until the loss function between corresponding point clouds converges.

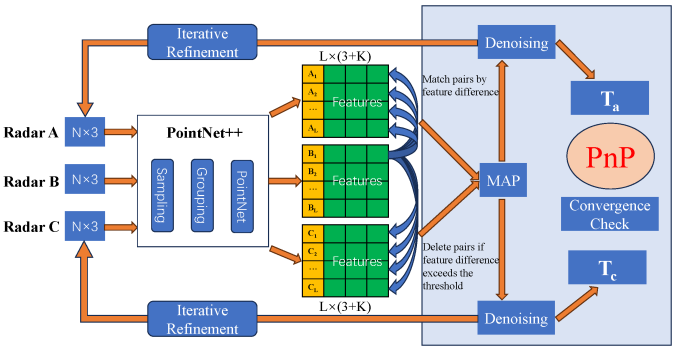


Figure 5: Structure of PointNet++&PnP. PointNet++ samples and extracts features from the source point clouds A and C, as well as the target point cloud B. The output of PointNet++ consists of L feature vectors, each containing K features and the 3D coordinates (x, y, z). Corresponding point pairs are identified based on these feature descriptors. The iteration process terminates when the loss function converges.

In contrast to conventional ICP, which directly minimizes Euclidean distances in Cartesian space, the PnP framework reformulates point cloud registration as a maximum a posteriori (MAP) estimation problem [70]. This probabilistic approach enables the incorporation of learned priors—such as geometric consistency or smoothness—through a denoising module. As a result, the optimization becomes more adaptive to data characteristics, converges more rapidly, and exhibits substantially reduced sensitivity to poor initialization. By leveraging these priors, the PnP framework is also better equipped to escape local minima and achieve more reliable alignment, particularly in the presence of complex structures, noise, or incomplete observations.

Although traditional ICP methods utilize least-squares minimization and singular value decomposition (SVD) [71]–[73], their reliance on explicit spatial correspondences fundamentally limits robustness to large initial misalignments and increases susceptibility to local optima. Even when enhanced with deep features such as those extracted by PointNet++, ICP essentially remains a local optimizer: it cannot recover globally consistent solutions without a favorable initialization [74]. The underlying objective is still confined to minimizing point-wise distances, lacking the flexibility to integrate learned priors or semantic information. Consequently, in scenarios involving noise, sparsity, or ambiguous geometries, ICP—regardless of feature augmentation—often delivers unstable or suboptimal registration outcomes.

As illustrated in Fig. 4, the proposed framework significantly improves both the accuracy and stability of point cloud registration, particularly in multi-view or multi-sensor scenarios. The core objective is to align source point clouds PC_A and PC_C to a common reference frame established by the target point cloud PC_B , through the estimation of optimal rigid transformations—specifically, rotation and translation parameters that best preserve the underlying geometric structures.

To enable feature-driven registration, PointNet++ is employed to embed each input point cloud into a higher-dimensional space enriched with latent geometric features. Formally, we define a feature extraction mapping as $f : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{L\times(3+K)}$, where N denotes the number of input points with spatial coordinates $[x, y, z]^T$, L is the number of sampled key points, and K represents the dimensionality of the learned feature descriptor appended to each point. The resulting $L\times(3+K)$ matrix thus captures both spatial and feature information for robust correspondence estimation as illustrated in Fig. 5.

For the input point clouds PC_A, PC_B, and PC_C, the representative key point sets $PC_{A'}$, $PC_{B'}$, and $PC_{C'}$ are obtained using farthest point sampling (FPS) [75]. Each sampled set is then processed by PointNet++ to obtain feature-augmented point representations:

$$\tilde{\mathbf{A}} = f(PC_A), \quad \tilde{\mathbf{B}} = f(PC_B), \quad \tilde{\mathbf{C}} = f(PC_C), \quad \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}} \in \mathbb{R}^{L\times(3+K)} \tag{7}$$

where $f(\cdot)$ denotes the feature extraction mapping performed by PointNet++, L is the number of sampled key points, and K is the dimension of the learned feature descriptor appended to each point.

For feature correspondence, let $\tilde{\mathbf{A}}_i \in \mathbb{R}^{3+K}$ denote the feature descriptor of the i-th key point in $\tilde{\mathbf{A}}$ (i.e., the i-th row of $\tilde{\mathbf{A}}$). The corresponding point in $PC_{B'}$ is determined by searching for the index $\mathbf{j}^*$ in $\tilde{\mathbf{B}}$ that minimizes the Euclidean distance:

$$\mathbf{j}^* = \arg\min_j \left\| \tilde{\mathbf{A}}_i - \tilde{\mathbf{B}}_j \right\|^2 \tag{8}$$

where $\tilde{\mathbf{B}}_j$ represents the feature descriptor of the j-th key point in $PC_{B'}$, and $\mathbf{j}^*$ is the index of the nearest neighbor in the feature space.

If the minimum feature distance exceeds a specified threshold τ that is continuously adjusted through experimental results, the candidate correspondence is discarded:

$$\text{If} \quad \min_j \left\| \tilde{\mathbf{A}}_i - \tilde{\mathbf{B}}_j \right\|^2 > \tau, \quad \text{then discard the pair.} \tag{9}$$

In the above, $PC_{A'}$, $PC_{B'}$, and $PC_{C'}$ are the sampled key point sets; $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ are the feature matrices produced by PointNet++; $\tilde{\mathbf{A}}_i$ is the feature descriptor of the i-th key point in $PC_{A'}$; $\tilde{\mathbf{B}}_j$ is the feature descriptor of the j-th key point in $PC_{B'}$; $\mathbf{j}^*$ denotes the index of the nearest neighbor; and τ is the feature distance threshold used to control correspondence reliability.

The observed coordinates of the source point clouds A and C, as well as the target point cloud B, are modeled as noisy measurements of their true underlying positions. The observation model can be formulated as:

$$\begin{cases} \mathbf{y}_i^a = T_a \mathbf{x}_i^a + \epsilon_{i,a} \\ \mathbf{y}_i^b = \mathbf{x}_i^b + \epsilon_{i,b} \\ \mathbf{y}_i^c = T_c \mathbf{x}_i^c + \epsilon_{i,c} \end{cases} \qquad \epsilon_{i,\cdot} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \tag{10}$$

where $\mathbf{y}_i^a$, $\mathbf{y}_i^b$, and $\mathbf{y}_i^c$ denote the observed (noisy) coordinates of the i-th point in source clouds A, C and target cloud B, respectively; $\mathbf{x}_i^a$, $\mathbf{x}_i^b$, and $\mathbf{x}_i^c$ are the corresponding true (noise-free) coordinates; T_a and T_c are the unknown spatial transformations from source clouds A and C to the target frame (including rotation and displacement); and $\epsilon_{i,a}$, $\epsilon_{i,b}$, and $\epsilon_{i,c}$ are Gaussian noise terms simulating measurement errors.

The overall objective is to jointly estimate the spatial transformations T_a, T_c and the true point locations $\{\mathbf{x}_i^a\}$, $\{\mathbf{x}_i^c\}$, $\{\mathbf{x}_i^b\}$, such that the aligned point clouds A, C, and B are maximally consistent while preserving reasonable internal structure via prior constraints. This leads to the following optimization problems:

$$\begin{aligned} &\min_{T_a, \{\mathbf{x}_i^a\}, \{\mathbf{x}_i^b\}} \left[\sum_i d\big(T_a \mathbf{x}_i^a,\, \mathbf{x}_i^b\big) - \sum_i \log p(\mathbf{x}_i^a) - \sum_i \log p(\mathbf{x}_i^b) \right] \\ &\min_{T_c, \{\mathbf{x}_i^c\}, \{\mathbf{x}_i^b\}} \left[\sum_i d\big(T_c \mathbf{x}_i^c,\, \mathbf{x}_i^b\big) - \sum_i \log p(\mathbf{x}_i^c) - \sum_i \log p(\mathbf{x}_i^b) \right] \end{aligned} \tag{11}$$

where T_a and T_c are the unknown spatial transformations, $\mathbf{x}_i^a$, $\mathbf{x}_i^c$, and $\mathbf{x}_i^b$ are the true (denoised) coordinates, $d(\cdot,\cdot)$ denotes the distance metric, $p(\cdot)$ is the prior probability describing geometric regularity, and $\log p(\cdot)$ ensures the prior is combined in a MAP framework.

To quantify the alignment error between transformed and target point clouds, the chamfer distance is used to measure the distance between the transformed point cloud A, C and B:

$$\begin{aligned} d_{\text{Chamfer}}(A', B) &= \frac{1}{|A|} \sum_{a' \in A'} \min_{b \in B} \|a' - b\|^2 + \frac{1}{|B|} \sum_{b \in B} \min_{a' \in A'} \|b - a'\|^2 \\ d_{\text{Chamfer}}(C', B) &= \frac{1}{|C|} \sum_{c' \in C'} \min_{b \in B} \|c' - b\|^2 + \frac{1}{|B|} \sum_{b \in B} \min_{c' \in C'} \|b - c'\|^2 \end{aligned} \tag{12}$$

This distance is symmetric and penalizes both missing points and outliers in both the source and target point clouds. In the definition of Chamfer distance, A' and C' denote the source point cloud after spatial transformation, and B is the target point cloud. $|A|$, $|C|$ and $|B|$ represent the number of points in each cloud, respectively. For each point a' in A', c' in C', the Chamfer distance computes the squared Euclidean distance to its nearest neighbor in B, and averages this over all points in A', C'. Similarly, for each point b in B, it computes the squared Euclidean distance to its nearest neighbor in A' and C', and takes the average.

In the alternating optimization scheme, the first step involves fixing the current denoised point coordinates $\mathbf{x}_i^{a,k}$, $\mathbf{x}_i^{c,k}$, and $\mathbf{x}_i^{b,k}$, and optimizing the spatial transformations T_a and T_c by minimizing the registration loss:

$$\begin{aligned} T_a^{k+1} &= \arg\min_{T_a} \sum_i d\left(T_a \mathbf{x}_i^{a,k},\, \mathbf{x}_i^{b,k}\right) \\ T_c^{k+1} &= \arg\min_{T_c} \sum_i d\left(T_c \mathbf{x}_i^{c,k},\, \mathbf{x}_i^{b,k}\right) \end{aligned} \tag{13}$$

where k denotes the current iteration, and $d(\cdot,\cdot)$ is the distance metric measuring alignment error between the transformed source and target points.

In the subsequent step of the alternating optimization, the spatial transformations T_a and T_c are fixed, and point cloud denoising is performed to enhance the quality of the latent coordinates via prior information:

$$\begin{aligned} \mathbf{x}_i^{a,k+1} &= D_\sigma\left(\mathbf{x}_i^{a,k}\right) \\ \mathbf{x}_i^{b,k+1} &= D_\sigma\left(\mathbf{x}_i^{b,k}\right) \\ \mathbf{x}_i^{c,k+1} &= D_\sigma\left(\mathbf{x}_i^{c,k}\right) \end{aligned} \tag{14}$$

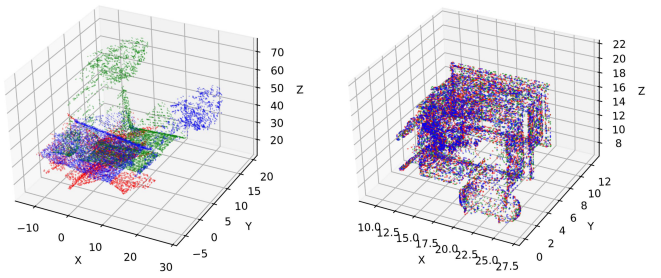
In the denoising process, the features extracted by PointNet++ are used as prior knowledge to guide the optimization of each point's denoising. The denoising operator $D_\sigma(\cdot)$ has a parameter σ that controls the strength of the denoising, typically related to the assumed noise level. At iteration k, $\mathbf{x}_i^{a,k}$, $\mathbf{x}_i^{b,k}$, and $\mathbf{x}_i^{c,k}$ represent the current coordinates, while the denoised updates are given by $\mathbf{x}_i^{a,k+1}$, $\mathbf{x}_i^{b,k+1}$, and $\mathbf{x}_i^{c,k+1}$.

The complete alternating optimization process can be compactly described by the following PnP gradient update rule:

$$\begin{aligned} \mathbf{x}_{k+1}^a &= \mathbf{x}_k^a - \alpha_a \nabla_{\mathbf{x}} d(T_a \mathbf{x}_k^a, \mathbf{x}_k^b) + \beta_a \left[D_\sigma(\mathbf{x}_k^a) - \mathbf{x}_k^a \right] \\ \mathbf{x}_{k+1}^c &= \mathbf{x}_k^c - \alpha_c \nabla_{\mathbf{x}} d(T_c \mathbf{x}_k^c, \mathbf{x}_k^b) + \beta_c \left[D_\sigma(\mathbf{x}_k^c) - \mathbf{x}_k^c \right] \end{aligned} \tag{15}$$

In this PnP gradient update formula, $\mathbf{x}_k$ denotes the coordinates of the point cloud at the k-th iteration, and α is the learning rate, which controls the update step size. $\nabla_{\mathbf{x}} d(T_a \mathbf{x}_k^a, \mathbf{x}k^b)$ and $\nabla_{\mathbf{x}} d(T_c \mathbf{x}_k^c, \mathbf{x}k^b)$ are the gradient of the registration loss with respect to the point cloud coordinates, indicating the direction to minimize the alignment error. $D\sigma(\mathbf{x}_k)$ is the denoised version of the point cloud, so $D\sigma(\mathbf{x}_k) - \mathbf{x}_k$ represents the gradient direction given by the denoising prior.

As shown in Fig. 6, the radar point clouds are captured using their respective independent coordinate systems, with each radar system collecting data from different physical locations, resulting in scattered data. This leads to disorganized point cloud data, as



(a) Raw point clouds. (b) Registered point clouds.

Figure 6: Example of point cloud alignment: (a) The original input point clouds from three radars with partial overlap (green point cloud from Radar A, red point cloud from Radar B, and blue point cloud from Radar C); (b) The registered point clouds aligned to the common reference frame of Radar B.

depicted in Fig. 6a. Due to the differences in the coordinate systems of each radar, the spatial positions of the point clouds are misaligned, making it difficult to directly compare or merge the data from different radars. These discrepancies in coordinate systems cause inconsistent distributions of the point clouds, uneven density, and even the possibility of overlap or gaps in the data. In order to align these point clouds and unify them into a global coordinate system, precise point cloud registration is necessary. By using a trained registration model, the point clouds can be iteratively optimized, gradually achieving accurate alignment, as shown in Fig. 6b. This ensures that the point clouds are reliably aligned, providing a solid foundation for subsequent analysis and processing.

To further assess the stability of the proposed PnP framework under varying spoofing intensities, we plot convergence curves at injection rates of 0%, 10%, 20%, and 30% (Fig. 7). The x-axis represents iteration steps, while the y-axis denotes the registration loss (Chamfer distance) during testing. The "injection rate" is defined as the ratio of spoofed points injected to the number of genuine points in the original cloud. The results show that under clean conditions (0% injection) the model converges rapidly to a low residual, while higher injection rates lead to slower convergence and higher residuals; nevertheless, the framework consistently exhibits a stable decreasing trend. This robustness arises from formulating registration as a MAP estimation and embedding a learned denoiser, which makes the optimization less sensitive to initialization and more resistant to spoofed or outlier points. By integrating denoising priors and feature-level guidance, the PnP framework significantly improves resilience against cross-view inconsistencies and poor initialization, and empirically converges stably across injection rates.

In summary, the proposed registration framework alternates between optimizing the spatial transformation parameters and denoising the point clouds based on prior knowledge. This iterative process continues until the registration error falls below a predefined threshold or convergence is achieved, thereby ensuring accurate and robust alignment of the input point clouds. During registration, the framework primarily relies on the regions jointly observed by all three radars that remain unaffected by spoofing, while the compromised radar points are simultaneously transformed as part of the global rigid motion. The complete algorithmic workflow is presented in Algorithm 1, which systematically outlines each step of the procedure.

B. Point Clouds Classification and Feature Extraction

This subsection systematically presents a heuristic point-level labeling strategy based on multi-radar spatial consistency, followed by

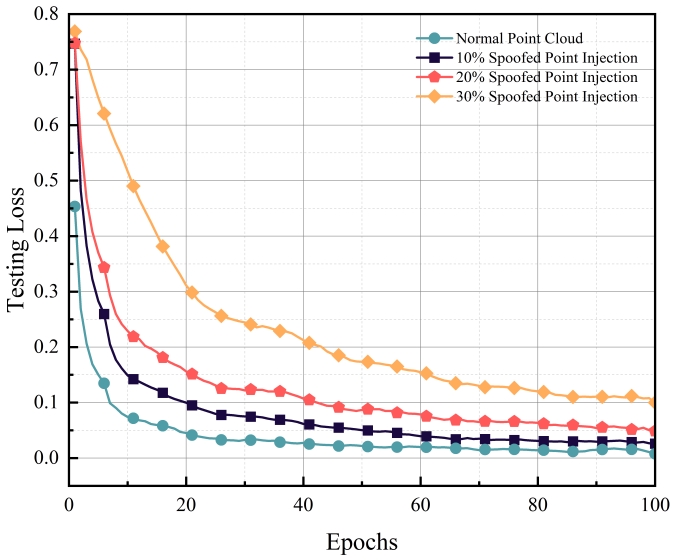


Figure 7: Convergence curves of the proposed PnP framework under different spoofing injection rates (0%, 10%, 20%, 30%). The x-axis denotes the iteration steps and the y-axis indicates the registration loss (Chamfer distance) during testing. Higher injection rates lead to slower convergence and higher residuals, yet the PnP framework consistently achieves stable alignment.

DBSCAN-based clustering and density-aware feature extraction. This approach is specifically tailored to leverage the spatial distribution properties of multi-radar point cloud data, thereby enhancing both the robustness and accuracy of spoofing detection.

In the fused point cloud, each point is treated as a core point, around which a spherical neighborhood with a predefined radius is constructed. In this study, the neighborhood is defined solely based on the Euclidean distance in the 3D spatial domain (x, y, z), without considering the normal vector components (N_x, N_y, N_z). Based on the typical inter-radar detection deviations observed in the dataset, the neighborhood radius is empirically set to 0.3 meters.

To preserve the source information of each point, a unique radar identifier (Radar ID) is assigned to every radar, which is then converted into a neural-network-compatible vector feature using one-hot encoding. For example, the three radars correspond to the vector forms $[1, 0, 0]$, $[0, 1, 0]$, and $[0, 0, 1]$, thereby avoiding any unintended interpretation of numerical magnitude relationships between categories. The authenticity of each core point is evaluated by examining whether its neighborhood $N(p)$ contains corresponding detections from both of the other radars, as indicated by distinct Radar IDs.

Formally, for each point $p \in PC$, its spherical neighborhood is defined as:

$$N(p) = \{q \in PC \mid \|q - p\|_2 \leq r,\ \text{radar\_id}(q) \neq \text{radar\_id}(p)\}\,, \tag{16}$$

where r is the neighborhood radius, and $\text{radar\_id}(\cdot)$ denotes the radar identifier of a point. The point-level label l_p is assigned as:

$$l_p = \begin{cases} 1, & \text{if } |\{\text{radar\_id}(q) \mid q \in N(p)\}| = 2 \\ 2, & \text{if } |\{\text{radar\_id}(q) \mid q \in N(p)\}| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

That is, if the neighborhood $N(p)$ contains points from both of the other two radars, the point p is labeled as "real" ($l_p = 1$); if $N(p)$ contains points from only one of the other two radars, p is labeled as "spoofed but not caused by this radar" ($l_p = 2$); otherwise, p is

Algorithm 1: Plug-and-Play Registration for Multi-View Point Clouds via PointNet++

Require: Source point clouds PC_A, PC_C; Target point cloud PC_B; Feature extractor $f(\cdot)$; Denoising network $D_\sigma(\cdot)$; Distance metric $d(\cdot,\cdot)$; Threshold τ; Max iterations $K_{\max}$; Tolerance ϵ

Ensure: Estimated rigid transformations T_a, T_c; aligned point clouds A', C'

Keypoint Sampling: For each point cloud PC_A, PC_B, PC_C, sample L keypoints using FPS, obtaining $PC_{A'}$, $PC_{B'}$, $PC_{C'}$.

Feature Extraction: Compute feature-augmented matrices $\tilde{\mathbf{A}} = f(PC_{A'})$, $\tilde{\mathbf{B}} = f(PC_{B'})$, $\tilde{\mathbf{C}} = f(PC_{C'})$, where $f(\cdot)$ is PointNet++.

Feature Matching:

for each keypoint i in $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{C}}$ **do**

  Find nearest neighbor index $\mathbf{j}^* = \arg\min_j \|\tilde{\mathbf{A}} - \tilde{\mathbf{B}}_j\|^2$ (similarly for $\tilde{\mathbf{C}}$).

  if $\min_j \|\tilde{\mathbf{A}} - \tilde{\mathbf{B}}_j\|^2 > \tau$ **then**

    Discard correspondence $(i, \mathbf{j}^*)$

  else

    Add $(i, \mathbf{j}^*)$ to correspondence set $\mathcal{M}_A$ (similarly for $\mathcal{M}_C$)

  end if

end for

Initialization:

Initialize denoised coordinates $\mathbf{x}^{a,0}$, $\mathbf{x}^{b,0}$, $\mathbf{x}^{c,0}$ from $PC_{A'}$, $PC_{B'}$, $PC_{C'}$.

for $k = 0$ to $k_{\max}$ **do**

  (A) Rigid Registration Step:

  Estimate $T_a^{k+1} = \arg\min_{T_a} d(T_a \mathbf{x}^{a,k}, \mathbf{x}^{b,k})$ using correspondences $\mathcal{M}_A$;

  Estimate $T_c^{k+1} = \arg\min_{T_c} d(T_c \mathbf{x}^{c,k}, \mathbf{x}^{b,k})$ using $\mathcal{M}_C$;

  Apply T_a^{k+1}, T_c^{k+1} to update coordinates for A, C;

  (B) Denoising Step:

  Update denoised coordinates: $\mathbf{x}^{a,k+1} = D_\sigma(\mathbf{x}^{a,k})$;

  $\mathbf{x}^{b,k+1} = D_\sigma(\mathbf{x}^{b,k})$;

  $\mathbf{x}^{c,k+1} = D_\sigma(\mathbf{x}^{c,k})$;

  (C) Convergence Check:

  Compute registration loss $L_k = d(T_a^{k+1}\mathbf{x}^{a,k+1}, \mathbf{x}^{b,k+1}) + d(T_c^{k+1}\mathbf{x}^{c,k+1}, \mathbf{x}^{b,k+1})$

  if $|L_k - L_{k-1}| < \epsilon$ **then**

    break

  end if

end for

Output final transformations $T_a = T_a^{k+1}$, $T_c = T_c^{k+1}$, and aligned/denoised point clouds.

labeled as "spoofing" ($l_p = 0$).

Fig. 8 demonstrates this clustering procedure. Fig. 8a shows a case where the point under test is surrounded by multiple neighboring detections from the other two radars. Thus, this detection is considered highly reliable and labeled as "real". In Fig. 8b, the cluster on the left indicates that the test point lacks point cloud data from one of the radars, suggesting that it may belong to a spoofed radar point cloud, but the spoofing was not caused by this radar. The cluster on the right shows that the test point lacks neighboring detections from both of the other radars, indicating that it is more likely a spoofed point, with the spoofing primarily occurring in the point cloud of this radar. According to the proposed heuristic method, such points are labeled as "spoofing" ($l_p = 0$) if the spoofing is primarily caused by this radar, or as "spoofed but not caused by this radar" ($l_p = 2$) if spoofing originates from another radar. As a result of this process, the unified point cloud PC is transformed into a preliminarily labeled version, PC', which serves as input for the following DBSCAN clustering.

DBSCAN partitions the labeled point cloud, PC', into a set of k clusters: $PC'_1, PC'_2, \ldots, PC'_k$. The number of the clusters, k, is determined by the data distribution and the selected parameters, ε and $minPts$. For each cluster, the average ratio of class-specific points is calculated based on the initial point-level labels ($l_p \in \{0, 1, 2\}$) assigned to the points in the PC'. Specifically, for each cluster, PC'_k, generated by DBSCAN, the average ratio of real points is calculated as follows:

$$R_k = \frac{1}{|PC'_k|} \sum_{p \in PC'_k} \mathbb{I}(l_p = 1) \,, \tag{18}$$

where $l_p \in \{0, 1, 2\}$ is the label of point p (1 for "real", 2 for "spoofed but not caused by this radar", 0 for "spoofing"), $|PC'_k|$ is total number of points in cluster PC'_k, and $\mathbb{I}(\cdot)$ denotes the indicator function. It is important to note that points labeled as $l_p = 2$ are deliberately excluded from the real-point ratio calculation. Including such points would implicitly assign credibility to data segments that, although unaffected by the current radar, still originate from spoofed regions. This would artificially inflate the estimated authenticity, thereby increasing the risk of misclassification. In safety-critical domains such as autonomous driving, such overestimation could lead to severe consequences, including potentially catastrophic traffic accidents.

Additionally, for each point, $p_j \in PC'_k$, the local point density within its spherical neighborhood is calculated as:

$$D_j = \log\left(1 + \sum_{p_i \in PC'_k,\ i \neq j} \mathbb{I}\left(\|p_i - p_j\|_2 \leq \varepsilon\right)\right) \,, \tag{19}$$

where ε is consistent with the DBSCAN clustering parameter, and the logarithmic transformation is applied to compress the value range in high-density regions. The local densities D_j are normalized to the range $[0, 1]$ as:

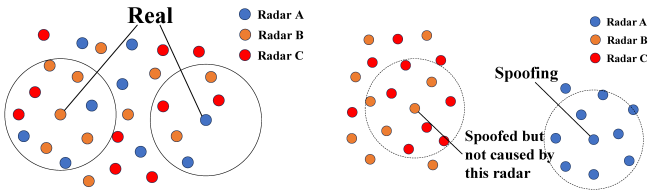
$$\tilde{D}_j = \frac{D_j - \min(D)}{\max(D) - \min(D)} \in [0, 1] \,, \tag{20}$$

where $D = \{D_i\}_{i=1}^{|PC'_k|}$ is the set of local densities for all points within the considered cluster. Finally, the cluster-level real-ratio, R_k, and point-level normalized density, $\tilde{D}_j$, are combined to define the final density weight for each point as:

$$w_d(j) = R_k \tilde{D}_j \,, \tag{21}$$

which provides a localized assessment of point authenticity by capturing the distribution of label types within the cluster's neighborhood. This weighting scheme emphasizes points that are both locally dense and belong to clusters with a high proportion of "real" points, thereby reinforcing spatial consensus as an indicator of authenticity.

Through this process, each cluster is enriched with localized, density-aware features, resulting in enhanced sub-clouds, $PC''_1, PC''_2, \ldots, PC''_k$. These enriched sub-clouds are then merged into a refined, feature-augmented point cloud, PC''. In the data authenticity verification stage, the point cloud, PC'', is normalized to ensure consistent representation across all feature dimensions, thereby facilitating effective downstream model training. Each point retains its 3D spatial coordinates and is further augmented with rich feature descriptors, including surface normals. These enhanced features are processed in PointNet++, which performs hierarchical feature extraction for ultimate robust spoofing detection.



(a) Potential real points. (b) Potential spoofed points.

Figure 8: DBSCAN-based spatial consistency verification concept, exemplifying real and spoofed detections in (a) and (b), respectively.

C. Spoofing Detection

This subsection presents the third stage of the processing pipeline, with the aim of detecting the spoofed radar. The merged and aligned point cloud, PC'', is classified into four categories: genuine (no radar is spoofed), Radar A-spoofed, Radar B-spoofed, and Radar C-spoofed. The unified point cloud PC'' is represented as an 8-dimensional feature representation, $p \in \mathbb{R}^8$: $p = (x, y, z, N_x, N_y, N_z, l_p, w_d)$, where $l_p \in \{1, 2, 0\}$ denotes the preliminary point-level label (1: real; 2: spoofed but not caused by this radar; 0: spoofing) assigned by the heuristic detection module (Section III-B) and w_d is the local density weight computed using DBSCAN clustering in (21). This feature construction approach effectively preserves spatial and geometric information, while integrating local structural characteristics and global consistency descriptors. These comprehensive features provide the network with strong discriminative capability for spoofing detection. Compared to conventional point cloud descriptors, this enriched representation enables the network to capture both fine-grained geometric variations and global density anomalies associated with spoofing attacks, which is crucial for robust multi-radar spoofing detection.

Table I outlines the architecture of the PointNet++ network, which is designed to perform fine-grained classification of the merged and aligned point cloud PC'' into four categories: genuine, Radar A-spoofed, Radar B-spoofed, and Radar C-spoofed. The network processes an input consisting of 1024 eight-dimensional points (each with 3 spatial coordinates and 5 additional features). Initially, these points are sampled and grouped in the SA_1 step, reducing them to 512 centroids. Each centroid aggregates 32 neighboring points, producing a tensor of the shape, $(512, 32, 8)$. Local features are extracted via a PointNet module, resulting in an output tensor of the size, $(512, 131)$, with 3 spatial coordinates and 128 features. A subsequent sampling and grouping step, SA_2, reduces these to 128 centroids, each again grouping 32 neighboring points, yielding a tensor of the shape, $(128, 32, 131)$. Another PointNet module extracts higher-level features, producing an output tensor of the size, $(128, 259)$ with 3 spatial coordinates and 256 features. Finally, in the third set abstraction step, SA_3, these 128 local features are globally aggregated via global pooling into a single vector of size, $(1, 1024)$, capturing the overall feature representation of the point cloud. This global feature is then passed through three fully connected layers ($FC_1 : 1024 \longrightarrow 512, FC_2 : 512 \longrightarrow 256$, and $FC_3 : 256 \longrightarrow 4$), which progressively transform the feature for four-class classification. The final output is a $(1, 4)$ tensor representing the predicted probabilities for each class.

To further ensure that the network effectively utilizes the newly introduced features l_p and w_d, an auxiliary regression head is inserted after the second set abstraction (SA_2) layer of the PointNet++ architecture, which outputs local feature representations $\mathbf{F}_2 \in \mathbb{R}^{128\times259}$. We perform global mean and max pooling across all points, resulting in a 259-dimensional mean vector $\mathbf{s}_{\text{mean}}$ and a 259-dimensional max vector $\mathbf{s}_{\max}$. These two vectors are concatenated to form a 2×259 statistical descriptor, which captures both average and most prominent feature activations across all local regions. This vector is then passed through a lightweight fully connected network to predict two global statistics of the input point cloud: the mean and the maximum value of the w_d feature, denoted as $\hat{\mu}_{w_d}$ and $\hat{m}_{w_d}$. The auxiliary regression loss, defined as the sum of mean squared errors (MSE) [76], [77] between the predicted and ground-truth global statistics, is formulated as:

$$L_{\text{aux}} = \text{MSE}(\hat{\mu}_{w_d}, \mu_{w_d}) + \text{MSE}(\hat{m}_{w_d}, m_{w_d}), \tag{22}$$

where μ_{w_d} and m_{w_d} denote the true mean and maximum values of w_d in the input point cloud. The training objective of DB-PointNet++ consists of two components: a main classification loss and an auxiliary regression loss. The total loss is defined as

$$L = L_{\text{main}} + \lambda L_{\text{aux}}, \qquad \lambda = 0.1. \tag{23}$$

For the four-class spoofing detection task, we adopt the categorical cross-entropy loss. Given the predicted logits $\mathbf{z}_i$ for sample i and the corresponding softmax probability

$$p_i^{(c)} = \frac{e^{z_i^{(c)}}}{\sum_{k\in\mathcal{Y}} e^{z_i^{(k)}}}, \tag{24}$$

the main loss is written as

$$L_{\text{main}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c\in\mathcal{Y}} \mathbb{1}\{y_i = c\} \log p_i^{(c)}. \tag{25}$$

This loss penalizes the negative log-likelihood of the ground-truth class. Theoretically, $L_{\text{main}} \in [0, \infty)$, where $L_{\text{main}} = 0$ indicates perfect classification and larger values correspond to greater prediction errors. During training, the loss typically converges to a small positive value as the network improves.

To enhance feature robustness against varying point densities, we introduce an auxiliary regression task that supervises the global statistics of point-wise density weights w_d. The network predicts both the mean $\hat{\mu}_{wd}$ and the maximum $\hat{m}_{wd}$, which are compared with the ground-truth statistics (μ_{wd}, m_{wd}) by the mean squared error:

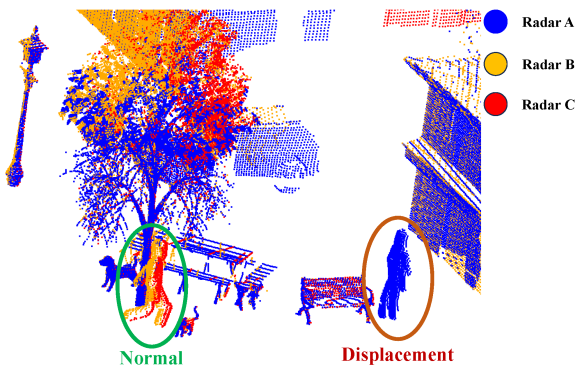
$$L_{\text{aux}} = (\hat{\mu}_{wd} - \mu_{wd})^2 + (\hat{m}_{wd} - m_{wd})^2. \tag{26}$$

Since MSE is non-negative, we have $L_{\text{aux}} \in [0, \infty)$, with lower values indicating better alignment between predicted and true statistics.

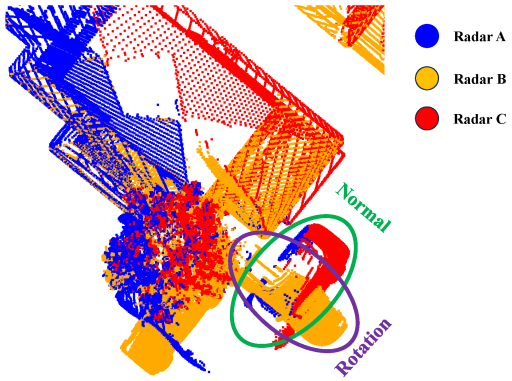
The total loss L combines the two terms, where the auxiliary regression loss is scaled by $\lambda = 0.1$ to avoid dominating the optimization process. This formulation ensures that the primary training signal comes from classification, while the auxiliary loss provides regularization and improves generalization under diverse spoofing scenarios. As a result, the overall loss function provides both

Table I: Hierarchical architecture of the PointNet++ classification network, processing a point cloud of $1,024$ points with 8-dimensional features per point.

Module	**Input (Pts × Ch)**	**Output (Pts × Ch)**
Input	1024 × (3+5)	1024 × (3+5)
SA_1	1024 × (3+5)	512 × (3+128)
SA_2	512 × (3+128)	128 × (3+256)
SA_3	128 × (3+256)	1 × 1024
Global Feature	1 × 1024	1024
FC_1	1024	512
FC_2	512	256
FC_3	256	4



(a) In this scenario, radars "B" and "C" operate normally and yield consistent pedestrian point clouds. Radar "A" is spoofed, resulting in a positional displacement of the detected pedestrian, leading to spatial misalignment with the normal cluster.



(b) In this scenario, radars "A" and "C" function correctly and capture consistent vehicle point clouds. Radar "B" is spoofed via an angular attack, causing the detected vehicle point cloud to be rotated and misaligned from the genuine cluster.

Figure 9: Radar Point Cloud Inconsistency under Spoofing Attacks

discriminative learning for spoof detection and structural robustness to density variations in radar point clouds.

Given a unified point cloud $PC'' = \{\mathbf{p}_i \mid \mathbf{p}_i \in \mathbb{R}^8, i = 1, ..., N\}$, each point is characterized by an 8-dimensional vector $\mathbf{p}_i = (x_i, y_i, z_i, N_{x,i}, N_{y,i}, N_{z,i}, l_{p,i}, w_{d,i})$. The feature extraction and classification process proceeds as follows:

1. Set Abstraction and Local Feature Learning. In each set abstraction (SA) layer, a subset of points is sampled to serve as centroids, and local neighborhoods are constructed:

$$\mathcal{C}^{(l)} = \text{FarthestPointSample}\big(\mathcal{P}^{(l-1)}, K^{(l)}\big), \tag{27a}$$

$$\mathcal{N}_j^{(l)} = \left\{\mathbf{p}_i \in \mathcal{P}^{(l-1)} \;\middle|\; \left\|\mathbf{p}_i - \mathbf{c}_j^{(l)}\right\| < r^{(l)}\right\}, \tag{27b}$$

$$\mathbf{f}_j^{(l)} = \text{PointNet}\left(\left\{\mathbf{p}_i - \mathbf{c}_j^{(l)} \;\middle|\; \mathbf{p}_i \in \mathcal{N}_j^{(l)}\right\}\right). \tag{27c}$$

where $\mathcal{P}^{(l-1)}$ denotes the set of input points to the l-th SA module, $K^{(l)}$ denotes the number of centroids sampled at the l-th set abstraction layer, $r^{(l)}$ the neighborhood radius, and $\mathbf{c}_j^{(l)}$ the j-th centroid.

2. Hierarchical Feature Aggregation. The local features are progressively abstracted through stacked SA modules (e.g., SA_1, SA_2, SA_3), yielding feature tensors of decreasing spatial resolution but increasing feature richness:

$$\mathbf{F}_1 = SA_1(\mathcal{P}_0), \tag{28a}$$

$$\mathbf{F}_2 = SA_2(\mathbf{F}_1), \tag{28b}$$

$$\mathbf{F}_3 = SA_3(\mathbf{F}_2) \tag{28c}$$

with $\mathbf{F}_3 \in \mathbb{R}^{1\times 1024}$ representing the global point cloud descriptor.

3. Global Feature Fusion and Classification. The global descriptor $\mathbf{F}_3$ is passed through a sequence of fully connected layers for final classification:

$$\mathbf{h}_1 = \text{ReLU}(\mathbf{W}_1\mathbf{F}_3 + \mathbf{b}_1), \tag{29a}$$

$$\mathbf{h}_2 = \text{ReLU}(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2), \tag{29b}$$

$$\mathbf{h}_3 = \mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3, \tag{29c}$$

$$\mathbf{o} = \text{softmax}(\mathbf{h}_3), \tag{29d}$$

where $\mathbf{o} \in \mathbb{R}^4$ provides the class probabilities for genuine, Radar A-spoofed, Radar B-spoofed, Radar C-spoofed.

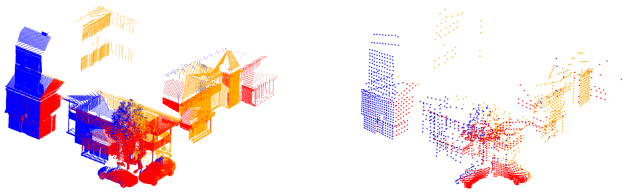
This work uses a PointNet++ model for detecting cyber attacks on automotive radar point clouds. Although PointNet++ has been widely used in 3D object scene classification and segmentation [78]–[80], as far as we are aware, PointNet++ has not been specifically reported for radar spoofing detection. The network is trained from scratch on a custom multi-radar simulated dataset using supervised learning with cross-entropy loss and a step-based exponential decay learning rate schedule.

IV. Radar Dataset

This section details the dataset generated for the performance evaluation of the proposed spoofing detection approach. The multi-angle, multi-scenario spoofed radar point cloud dataset is generated using the Unity software. This dataset was generated by considering three radars positioned at different angles and locations to capture point cloud data from the same scene synchronously. In addition, the influence of the radar resolution on the spoofing capabilities is evaluated by considering radars with low- and high-resolution of $5,000 - 10,000$ and $30,000 - 50,000$ detections per frame, respectively.

The scenarios in which one of the radars is compromised by spoofing are investigated, where the spoofing effects are simulated not only as positional displacements of radar targets but also as variations in their detected orientations. Specifically, angular spoofing is modeled by rotating the target around the vertical (Z) axis by different degrees, thereby emulating sophisticated attack patterns that distort both location and heading information, as may occur in real-world automotive environments. Fig. 9 illustrates representative examples of such simulated attacks. In Fig. 9a, spoofing is realized through a positional shift, where the spoofed radar "A" detects the pedestrian at a displaced position relative to the genuine detections from radars "B" and "C." In contrast, Fig. 9b demonstrates angular spoofing, in which radar "B" operates normally, but the spoofed radar perceives the vehicle's point cloud as rotated due to an artificial transformation about the Z-axis. To enhance realism and diversity, the simulated dataset includes multiple object categories—such as walking and running pedestrians, vehicles, and static obstacles—thereby representing the complexity of dynamic urban environments.

The low-resolution imaging radar considered in this study has a measurement range of up to 100 meters and operates with a two-dimensional angular resolution of $2^\circ \times 0.4^\circ$ across a field of view (FOV) of $120^\circ \times 30^\circ$. For each frame, this configuration generates $5,000$ to $10,000$ detection points, effectively simulating the sparse data typically produced by entry-level automotive radars in wide-area surveillance tasks. In contrast, the high-resolution imaging radar is configured with an extended measurement range of 200 meters and



(a) High-resolution radar with $30,000$–$50,000$ detections per frame.

(b) Low-resolution radar with $5,000$–$10,000$ detections per frame.

Figure 10: Representative example of the simulated measurements of the high- and low-resolution radars, in subplots (a) and (b), respectively.

a finer 2D angular resolution of $0.2^\circ \times 0.1^\circ$, maintaining a FOV of $120^\circ \times 25^\circ$. This radar produces substantially denser point cloud data, with $30,000$ to $50,000$ detections per frame, reflecting the data richness achievable with state-of-the-art advanced imaging radars. Representative examples of both high- and low-resolution radar point clouds are presented in Fig. 10, where Fig. 10a corresponds to the high-resolution radar and Fig. 10b illustrates the low-resolution case.

Each radar frame in the dataset includes the (x, y, z) coordinates of detected points, the three components of the normal vector, the preliminary label l_p, and the density weight w_d. The normal vectors are estimated using principal component analysis (PCA) [81], [82] applied to local neighborhoods, with parameter choices adapted to the radar resolution. For low-resolution radars, a larger neighborhood size is used to ensure robust estimation under sparse point density, whereas for high-resolution radars, a smaller neighborhood is adopted to preserve fine-grained geometric details. To simulate position spoofing, the location of the spoofed object in the attacked radar is displaced by 2, 5, 10, or 20 meters from its true position. In addition, angular spoofing is modeled by rotating the spoofed object within the attacked radar's view by 30°, 45°, or 90° about the vertical (Z) axis, thereby altering the apparent orientation of the target in the point cloud. These manipulations enable a comprehensive evaluation of the detection approach under varying spoofing perturbation magnitude and attack modalities. To the best of our knowledge, this is the first simulated automotive radar point cloud dataset to comprehensively cover both positional and angular spoofing across diverse real-world scenarios and multiple sensor resolutions. In total, a dataset of 30 GB was generated through simulation.

V. Performance Evaluation

This section provides a comprehensive evaluation of the proposed DB-PointNet++ framework for radar spoofing detection. A diverse set of performance metrics is employed to assess its effectiveness across various automotive radar scenarios, covering both high- and low-resolution configurations as well as multiple spoofing perturbation magnitude.

The evaluation relies on standard classification metrics—detection accuracy [56], F1-score [83], recall [84], false positive rate (FPR) [85], and false negative rate (FNR) [86]—to quantitatively assess the framework's capability in identifying spoofed radar point clouds. These metrics are chosen because accuracy provides an overall measure of recognition correctness, recall and FNR reveal the model's ability to capture true attacks and avoid missed detections; FPR reflects the system's robustness in rejecting genuine samples and ensuring practical applicability; and the F1-score balances precision and recall, which is particularly important under class imbalance

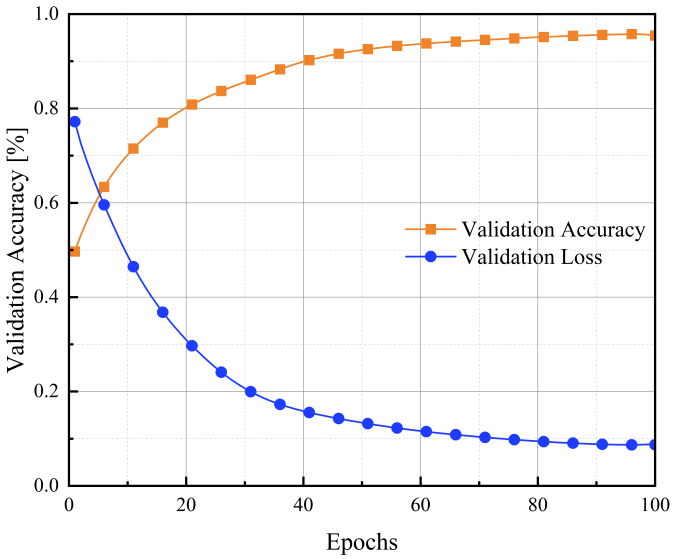


Figure 11: Validation performance of the proposed DB-PointNet++ framework on the simulated radar dataset. The figure illustrates a steady increase in validation accuracy accompanied by a consistent decrease in validation loss over 100 epochs, indicating stable convergence and effective model learning.

in spoofing detection tasks. Together, these metrics provide a comprehensive assessment of both the strengths and limitations of the framework. Beyond radar resolution, we further examine the influence of different target types. Specifically, vehicles and pedestrians are selected as representative classes, and both position-displacement and angular-rotation spoofing attacks are applied to them. This setup enables a thorough analysis of the model's robustness against diverse attack modalities.

a) Per-class confusion accounting

We consider a single-label multi-class setting with C classes. For any class $c \in \{1, \ldots, C\}$, we adopt a one-vs-rest view to define true positives TP_c, false positives FP_c, true negatives TN_c, and false negatives FN_c. All class-wise metrics below are computed from this four-tuple.

b) Accuracy

Detection accuracy is the most straightforward metric:

$$\mathrm{Acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{\hat{y}_i = y_i\}, \tag{30}$$

where N is the number of test samples, $\hat{y}_i$ the prediction, and y_i the ground truth.

c) Recall/TPR, FNR, Precision, FPR

For a binary classifier with true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), we use the standard definitions:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad \mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{TPR}, \tag{31}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{32}$$

These denominators are the numbers of positive and negative cases and are nonzero in our setting; hence no numerical stabilizer is required.

d) Multi-class extension (one-vs-rest)

For a class c in a K-class problem, we form a one-vs-rest reduction by treating c as the positive class and all other classes as negative,

yielding $\mathrm{TP}_c, \mathrm{FP}_c, \mathrm{TN}_c, \mathrm{FN}_c$. We then reuse the binary formulas:

$$\mathrm{TPR}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}, \qquad \mathrm{FNR}_c = 1 - \mathrm{TPR}_c, \tag{33}$$

$$\mathrm{Precision}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}, \qquad \mathrm{FPR}_c = \frac{\mathrm{FP}_c}{\mathrm{FP}_c + \mathrm{TN}_c}. \tag{34}$$

If $\mathrm{TP}_c + \mathrm{FN}_c = 0$ (no positive instances) or $\mathrm{FP}_c + \mathrm{TN}_c = 0$ (no negative instances), the corresponding metric is undefined and the class is excluded from the aggregate.

e) Averaging

When reporting a single summary across classes, we use macro averaging $\mathrm{Metric}_{\text{macro}} = \frac{1}{K}\sum_{c=1}^{K} \mathrm{Metric}_c$.

f) F1-score

The class-wise F1-score, used throughout this study, is defined as the harmonic mean of precision and recall:

$$\mathrm{F1}_c = \frac{2\,\mathrm{Prec}_c\,\mathrm{Rec}_c}{\mathrm{Prec}_c + \mathrm{Rec}_c} = \frac{2TP_c}{2TP_c + FP_c + FN_c}. \tag{35}$$

It jointly penalizes both false positives and false negatives, providing a balanced measure of detection accuracy.

g) Final evaluation metric.

Among the defined metrics, we adopt the Macro-F1 score as the primary criterion for model comparison, since it jointly penalizes false positives and false negatives while treating all classes equally. For each class c, the F1-score is computed as

$$\mathrm{F1}_c = \frac{2\,\mathrm{Prec}_c\,\mathrm{Rec}_c}{\mathrm{Prec}_c + \mathrm{Rec}_c + \varepsilon}, \tag{36}$$

and the macro-average is obtained by

$$\text{Macro-}F1 = \frac{1}{C}\sum_{c=1}^{C} \mathrm{F1}_c. \tag{37}$$

The experimental findings yield valuable insights into the practical deployment of the DB-PointNet++ framework in real-world autonomous driving. The results demonstrate that the proposed framework substantially enhances the robustness of radar perception systems against spoofing attacks, thereby contributing to safer and more reliable autonomous navigation.

A. Performance Testing of Simulated Radar Dataset

To evaluate the performance of the proposed DB-PointNet++ framework, we first conducted experiments on a simulated radar dataset generated using Unity. This dataset encompasses diverse spoofing scenarios, including both position displacement and angular rotation attacks, applied across low- and high-resolution radar configurations. By providing a controlled yet realistic approximation of real-world conditions in autonomous driving, the dataset enables a thorough and reproducible assessment of the model's spoofing detection capability under varying environmental dynamics.

We initially formulated spoofing detection as a binary classification problem, aiming to distinguish whether a radar point cloud was genuine or spoofed. This simplified setting established a clear baseline for assessing the core discriminative capability of DB-PointNet++, without the added complexity of identifying which radar unit was compromised. We then extended the task to a more demanding multi-class formulation, requiring the model to differentiate between genuine samples and three distinct spoofing sources (A, B, and C). Finally, we conducted a direct comparison between binary and multi-class detection to evaluate how increasing task complexity impacts overall detection performance.

V-A1 Binary Classification: Genuine vs. Spoofed Point Clouds

To establish a baseline for evaluating spoofing detection, we first formulated the task as a binary classification problem. In this setting, the objective is simply to determine whether a radar point cloud is genuine or spoofed. This formulation isolates the fundamental discriminative capability of DB-PointNet++, allowing us to assess its ability to capture spoofing-induced anomalies without the added complexity of multi-class attribution.

The validation dynamics of the model are illustrated in Fig. 11. Validation accuracy rises sharply during the first 20 epochs and gradually stabilizes, converging to approximately 95% after 100 epochs. At the same time, validation loss decreases consistently and plateaus in the later stages. These complementary trends demonstrate both stable convergence and strong generalization, confirming that the model effectively captures discriminative features from the simulated dataset. The validation set plays a key role in this process, as it is independent of the training data and serves to monitor progress, identify potential overfitting, and guide model selection and threshold calibration.

Beyond overall detection performance, we further investigated how spoofing type and radar resolution affect detection outcomes, as summarized in Fig. 12. Fig. 12a–d correspond to displacement attacks, while Fig. 12e–h depict rotation attacks. The results reveal that radar resolution has a decisive impact: high-resolution radars consistently outperform low-resolution counterparts, especially under small displacements and rotations. In high-resolution settings, detection accuracy exceeds 90% even under large perturbations, whereas low-resolution configurations often remain below 70% and exhibit substantially higher FPR and FNR. These findings emphasize that fine-grained spatial detail is critical for reliable spoofing detection.

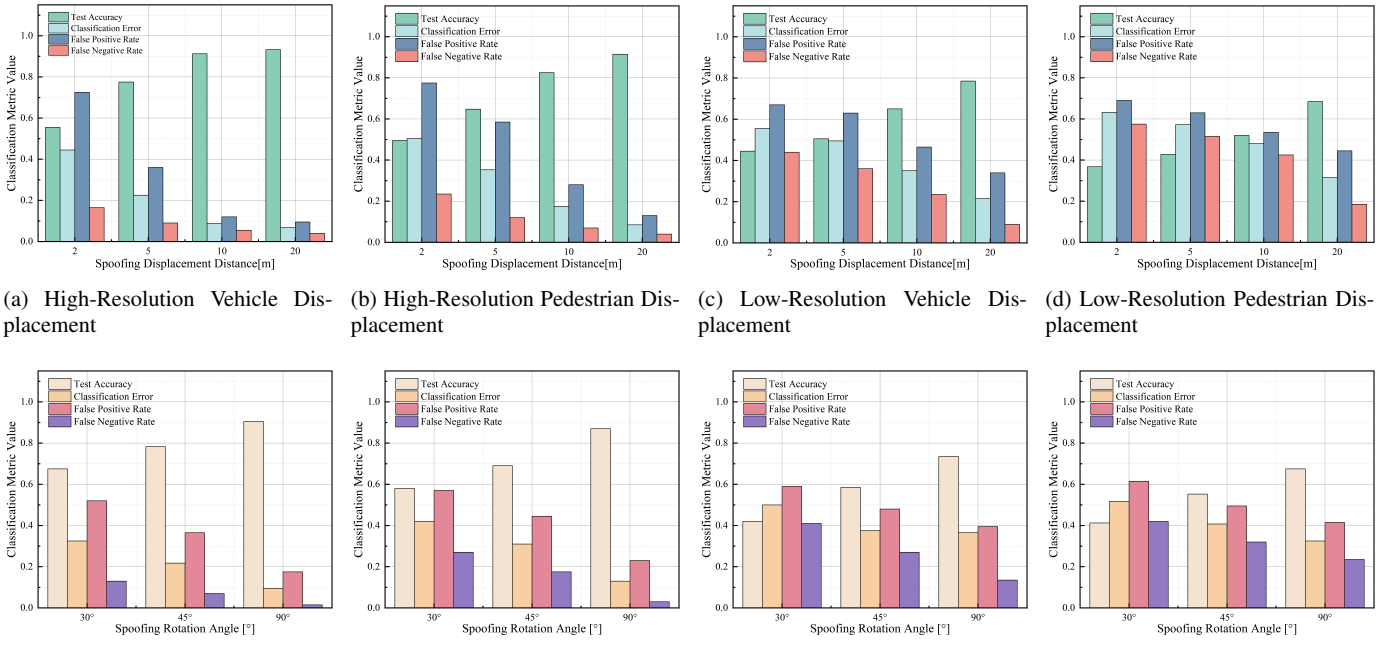
A clear difference is also observed between displacement- and rotation-based spoofing. Rotation attacks are generally more difficult to detect, as their effects on local orientation and object geometry often remain plausible to the model. For instance, displacement attacks of 20m (Fig. 12c–d) can still be identified with reasonable accuracy even in low-resolution pedestrian settings, whereas 90° rotations (Fig. 12g–h) remain challenging. This highlights that displacement introduces more distinct cross-radar inconsistencies, making it relatively easier to capture, while rotation perturbations yield subtler anomalies that complicate detection.

Target type further influences detection performance. As shown in Fig. 12a, c, e, and g, vehicle spoofing achieves consistently higher accuracy and lower error rates than pedestrian spoofing, shown in Fig. 12b, d, f, and h. Vehicles, owing to their larger size and rigid geometric structure, produce more stable point cloud signatures, facilitating robust detection. Pedestrians, by contrast, generate fewer reflection points and greater variability, especially in low-resolution radar, leading to degraded accuracy and more frequent misclassifications.

Finally, we identify a consistent trend with attack magnitude across both displacement and rotation scenarios. Detection performance improves as displacement increases from 2m to 20m and as rotation grows from 30° to 90°. Under stronger perturbations, test accuracy and recall rise while both FPR and FNR decline. This indicates that larger spoofing disturbances introduce more pronounced anomalies in radar point clouds, which the DB-PointNet++ framework can more effectively exploit.

V-A2 Multi-Class Classification: Genuine vs. Spoofed from Radars A/B/C

To further increase task complexity, we reformulated spoofing detection as a multi-class classification problem. In this setting, the model was required not only to determine whether a radar point cloud was genuine or spoofed, but also to localize the spoofing source by identifying attacks targeting radar A, B, or C. This task formulation better reflects real-world adversarial conditions in autonomous driv-



(a) High-Resolution Vehicle Displacement

(b) High-Resolution Pedestrian Displacement

(c) Low-Resolution Vehicle Displacement

(d) Low-Resolution Pedestrian Displacement

(e) High-Resolution Vehicle Rotation

(f) High-Resolution Pedestrian Rotation

(g) Low-Resolution Vehicle Rotation

(h) Low-Resolution Pedestrian Rotation

Figure 12: Classification performance of DB-PointNet++ under diverse spoofing scenarios, including displacement and rotation attacks on vehicles and pedestrians with both high- and low-resolution radar configurations, evaluated on a simulated radar dataset. The results show that high-resolution radar consistently outperforms low-resolution radar, especially under small perturbations, achieving accuracies above 90% for large displacements and rotations. In contrast, low-resolution settings often fall below 70%, with substantially higher FPRs and FNRs. Moreover, rotation-based spoofing is generally more challenging to detect than displacement-based spoofing, as rotated point clouds can still resemble plausible object orientations, whereas displacement introduces clearer spatial inconsistencies.

ing, where both accurate detection and precise attribution are critical for effective mitigation.

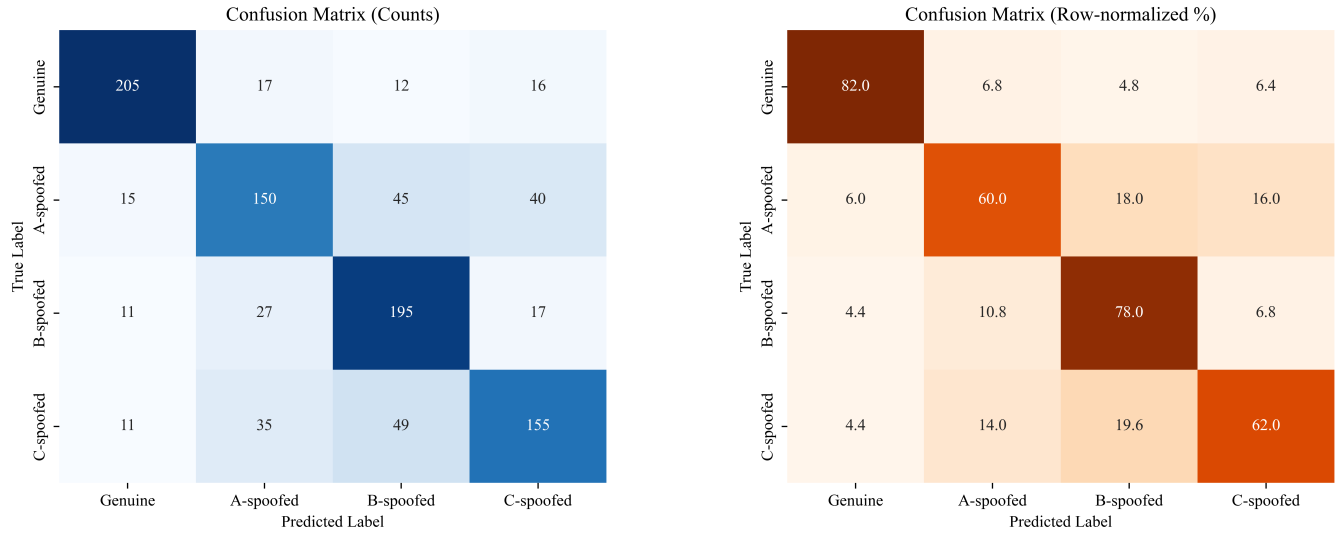
Fig. 13 presents the four-class confusion matrices (rows: ground truth; columns: prediction; $N = 1,000$). Subfig. 13a shows the absolute prediction counts, with the diagonal summing to 705, yielding an overall accuracy of 70.5%. Errors are concentrated among the spoofed classes: Radar A–spoofed is most often misclassified as Radar B–spoofed or Radar C–spoofed (45 and 40 instances), while Radar C–spoofed is confused with Radar B–spoofed and Radar A–spoofed (49 and 35 instances). By contrast, spoofed samples incorrectly labeled as Genuine are relatively few (A/B/C $\rightarrow$ Genuine: 15/11/11). Subfig. 13b reports the row-normalized percentages, highlighting relative error patterns independent of class size. Here, Genuine and Radar B–spoofed achieve the highest per-class accuracies (82% and 78%), whereas Radar A–spoofed and Radar C–spoofed remain less separable, dominating residual misclassifications. These results indicate that while the detector is reliable in identifying whether spoofing is present, it is less effective in localizing which radar is compromised. The comparatively strong performance of Radar B may be attributed to its placement and field of view, which provide more stable coverage of primary objects.

The overall classification performance of DB-PointNet++ is further evaluated using the receiver operating characteristic (ROC) curve [87], [88], as shown in Fig. 14a. The ROC curve lies well above the diagonal random baseline, achieving an AUC of 0.836, which indicates moderate discriminative capability between genuine and spoofed radar point clouds. This result validates DB-PointNet++ as a reasonable baseline model; however, it also reveals a limitation—its current level of performance remains insufficient for safety-critical autonomous driving scenarios, where extremely low false-alarm rates and high operational reliability are mandatory.

Two representative operating points are highlighted. The Youden point (orange marker)—the threshold maximizing (TPR − FPR)—achieves TPR = 88.9% and FPR = 34.1%, representing a balanced compromise that correctly detects nearly 89% of spoofing attacks while missing approximately 11%. Although statistically acceptable, such a miss rate would be problematic in safety-critical contexts. In contrast, the business-priority point (red cross) follows a conservative design principle of "better false alarms than missed detections," attaining TPR = 96.3% at the expense of FPR = 50%. This configuration is more suitable for a preliminary screening module that prioritizes recall, where subsequent refinement—via sensor fusion, temporal consistency checks, or physics-based plausibility validation—can effectively reduce false alarms downstream.

Fig. 14b compares the discriminative responses of DB-PointNet++ to displacement-based (blue) and rotation-based (orange) spoofing attacks. The displacement curve consistently remains above the rotation curve across most thresholds, confirming stronger separability for displacement perturbations. Quantitatively, displacement achieves an AUC of 0.906, surpassing 0.852 for rotation. Since AUC reflects global discriminability, this difference indicates that displacement attacks introduce more pronounced geometric inconsistencies among cross-radar observations, making them easier to identify. The performance gap is particularly notable in the low-FPR region (e.g., FPR < 0.1), where the displacement curve rises sharply and yields higher true positive rate (TPR), suggesting that displacement spoofing can be detected more reliably under strict false-alarm constraints.

This discrepancy originates from inherent geometric characteristics. Displacement directly perturbs inter-radar correspondences, producing visible misalignments and larger registration residuals. In



(a) Confusion Matrix (Counts)

(b) Confusion Matrix (Row-normalized %)

Figure 13: Confusion matrices of DB-PointNet++ evaluated on the simulated dataset. Subplot (a) shows the absolute prediction counts, while subplot (b) presents the row-normalized percentages, highlighting the relative distribution of misclassifications across different spoofing categories. These results demonstrate that under ideal simulation conditions, the model achieves high accuracy with limited confusion between genuine and spoofed samples.

contrast, rotation tends to preserve pairwise distances and local point density structures. Small-angle rotations are often masked by sensor noise, and object symmetries—such as those found in vehicles or pedestrians—further obscure the distinction. Consequently, the model exhibits inherently lower sensitivity to rotation-based spoofing.

To further examine class-wise separability, Fig. 14c illustrates the one-vs-rest (OvR) ROC curves. All curves lie distinctly above the random baseline, confirming that the model achieves meaningful discrimination across spoofing categories. The corresponding AUC values are 0.825 for B-spoofed, 0.767 for A-spoofed, and 0.751 for C-spoofed, yielding the ranking B-spoofed $>$ A-spoofed $\approx$ C-spoofed. In the low-FPR region, the B-spoofed curve rises more sharply, maintaining a higher TPR under stringent false-alarm constraints, whereas the A- and C-spoofed curves increase more gradually—indicating that conservative thresholds disproportionately suppress recall for these two classes. Overall, DB-PointNet++ exhibits strong performance in coarse-level "attack vs. no-attack" discrimination but shows reduced effectiveness in fine-grained spoof-type attribution. For safety-critical deployment scenarios emphasizing low FPR, improving the recognition of A- and C-type spoofing should therefore be prioritized.

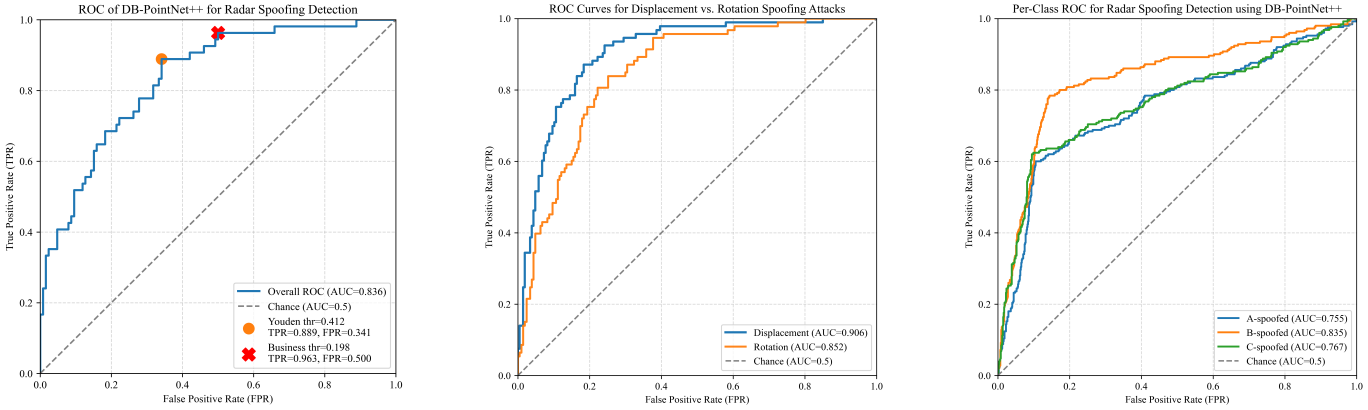
We also assess detection performance under varying spoofing magnitudes in Fig. 15. As shown in Fig. 15a and Fig. 15d, accuracy consistently improves with attack strength, regardless of whether the perturbation is translational or rotational. For instance, accuracy rises from 64.5% at 2m displacement to 83.5% at 20m, and from 63.5% at 30° rotation to 80.5% at 90°. This does not imply stronger attacks are trivial; rather, larger perturbations introduce more pronounced violations of geometric consistency—centroid and neighborhood translation or orientation and normal rotation—that amplify separability and reduce uncertainty, yielding more stable decision boundaries.

Resolution further reinforces this phenomenon. Across all scenarios, high-resolution radars outperform low-resolution ones. High-resolution data provide denser, more detailed point clouds that preserve fine-grained structures, enhancing the stability of geometric and density-based features. This additional detail allows the model to capture spoofing-induced anomalies more effectively, leading to substantial performance gains. In contrast, low-resolution radars produce sparse, noisy point clouds in which subtle perturbations are easily masked, making spoofing detection more difficult.

Target category also impacts detection. Vehicles, with their larger reflective surfaces and stable geometry, produce stronger and more consistent signatures, making spoofing effects more evident—especially under high resolution. Pedestrians, by contrast, generate weaker and noisier reflections with smaller spatial footprints, leading to greater susceptibility to noise and lower separability. This gap is especially pronounced under low-resolution settings, where pedestrian spoofing often overlaps with background noise, resulting in degraded accuracy. These findings suggest that spoofing detection for small, weak-reflection targets remains particularly challenging and may benefit from integrating temporal or Doppler features.

A direct comparison of displacement- and rotation-based spoofing further underscores these differences. As shown in Fig. 15a and Fig. 15d, translation-based spoofing consistently outperforms rotation-based spoofing under equivalent conditions. For example, in high-resolution vehicle experiments, accuracy reaches 83.5% under 20m displacement but only 80.5% under 90° rotation. This performance gap stems from the more salient geometric disruptions introduced by translation compared with the smoother, lower-saliency variations of rotation. At lower resolutions, rotational perturbations become even harder to distinguish from noise, further widening the gap.

Error-rate analysis provides additional insight. Fig. 15b and Fig. 15c show FPR and FNR under displacement, while Fig. 15e and Fig. 15f show the corresponding rotation results. In all cases, both metrics decrease as spoofing intensity increases, confirming that stronger attacks are easier to detect. The largest improvements occur in high-resolution vehicle scenarios, where FPR falls from about 55% to 23% and FNR from 15% to below 10%. Low-resolution pedestrian cases remain weakest, with FPR only dropping from 60% to 55% and FNR from 35% to 29%. When comparing attack types, displacement achieves greater reductions than rotation, consistent with its stronger



(a) Binary spoofing detection. (b) Displacement vs. rotation attacks. (c) Multi-class spoofing detection.

Figure 14: Detection performance of DB-PointNet++ under multiple spoofing evaluation settings, evaluated on a simulated radar dataset. (a) Binary ROC curve for overall spoofing detection achieves an AUC of 0.836. Two key operating points are highlighted: the Youden point ($TPR = 88.9\%$, $FPR = 34.1\%$) balancing sensitivity and specificity, and the recall-first point ($TPR = 96.3\%$, $FPR = 50\%$) prioritizing safety at the cost of higher false alarms. (b) Comparison between displacement- and rotation-based spoofing attacks demonstrates stronger robustness against displacement ($AUC = 0.906$) than rotation ($AUC = 0.852$), as displacement introduces more severe disruptions to geometric consistency across radars. (c) One-vs-rest ROC curves for multi-class spoofing detection show higher separability for B-spoofed ($AUC = 0.825$) compared to A- and C-spoofed ($AUC = 0.767$, 0.751), revealing challenges in fine-grained type discrimination. Overall, these results validate DB-PointNet++ as a solid baseline for spoofing detection, while highlighting areas needing improvement for safety-critical deployment.

geometric impact.

Overall, displacement spoofing outperforms rotation spoofing in both FPR and FNR. High-resolution radars consistently yield stronger results than low-resolution ones, and vehicles are more reliably detected than pedestrians due to their larger radar cross-sections and more stable reflective structures.

Finally, Fig. 16 illustrates the evolution of classification error, F1 score, and recall under displacement spoofing. As displacement increases, vehicle detection consistently outperforms pedestrian detection, especially under high resolution, with lower errors and higher F1/recall. Pedestrian detection under low resolution, however, remains limited, with persistently high errors and suppressed recall.

Under high-resolution conditions, both displacement and rotation lead to marked improvements as spoofing intensity increases. For example, vehicles under displacement reduce classification error from 35.5% to 16.5%, with F1 reaching 75% and recall 88%. Pedestrians also improve, but less substantially, with classification error dropping from 58.5% to 44%. Under rotation, vehicles achieve a minimum classification error of 20% at 90°, while pedestrians remain at 25%. In low-resolution settings, improvements are smaller: vehicles under displacement reduce classification error from 45.5% to 33.5%, while pedestrians improve only from 47.75% to 42%. Similar trends hold under rotation, though improvements are again weaker than for displacement. These results emphasize that displacement produces more discriminative geometric anomalies, vehicles are inherently more detectable than pedestrians, and high-resolution radar provides the strongest performance gains.

Fig. 17 summarizes the test-set results: overall accuracy $= 65\%$, macro-averaged precision $= 62\%$, recall $= 79\%$, and F1 $= 69\%$. The error profile includes classification error $= 35\%$, macro-averaged FPR $= 49\%$, and macro-averaged FNR $= 21\%$, corresponding to specificity of 51%. Overall, the model operates in a high-recall but low-precision regime—sensitive to attacks with few misses but prone to false alarms. This trade-off may be acceptable for safety-critical contexts, but when stricter false-positive control is required, strategies such as threshold tuning, probability calibration, cost-sensitive learning, or hard-negative mining can reduce FPR while preserving recall.

In summary, DB-PointNet++ demonstrates strong spoofing detection ability in simulation, particularly for displacement-based and vehicle-focused scenarios under high-resolution radar. However, performance degrades for rotation-based spoofing, pedestrian targets, and low-resolution conditions. While the model achieves high recall, its relatively high FPRs highlight the need for improved feature design, multi-modal fusion, or cost-sensitive training to enhance precision and robustness for real-world deployment.

V-A3 Comparative Analysis of Binary and Multi-Class Tasks

To further assess the robustness of the proposed detector, we systematically compare binary and multi-class classification schemes. This analysis quantifies the performance trade-off when moving from a simplified binary formulation to a more demanding multi-class framework, thereby revealing the challenges of fine-grained spoofing attribution.

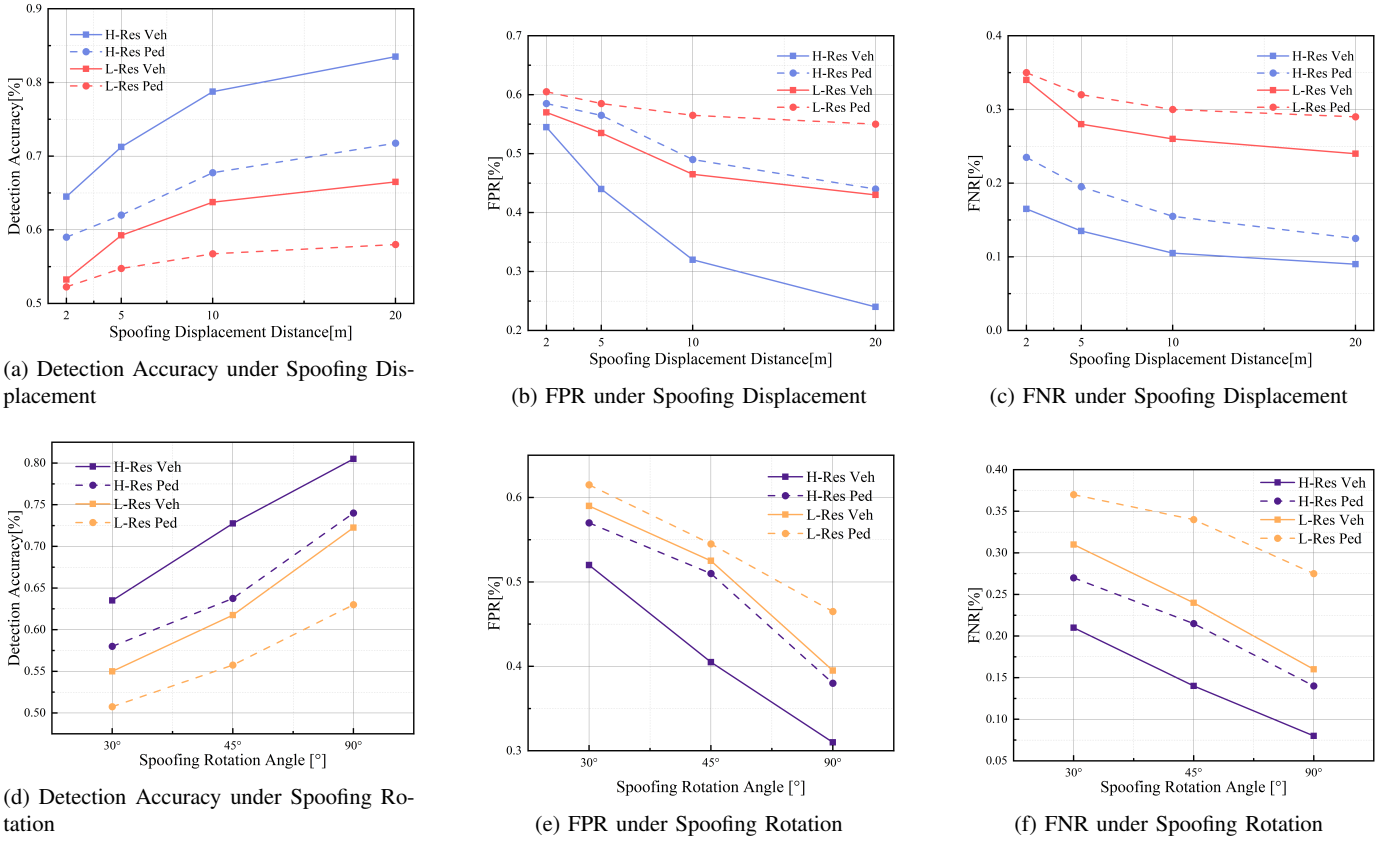
At first glance, collapsing multiple spoofing categories into a single "attack" label should reduce decision complexity and thus improve detection accuracy, as suggested by

$$\begin{aligned} p(\text{attack}) &= 1 - p(\text{no\_attack}) \\ &= p(\text{A attacked}) + p(\text{B attacked}) + p(\text{C attacked}). \end{aligned} \tag{38}$$

From this perspective, the binary setting appears advantageous. However, the experimental evidence shows a more nuanced picture that depends critically on radar resolution and attack type.

As illustrated in Fig. 18a, accuracy patterns diverge across conditions. Under displacement attacks, multi-class classification particularly benefits pedestrian detection at high resolution (65% vs. 57% for binary), while vehicle performance remains comparable across the two schemes. Under rotation attacks, binary classification is favored at high resolution (approximately 76% vs. 72% for multi-class), whereas multi-class yields a modest advantage at low resolution.

These trends are reinforced by the classification error in Fig. 18b: binary classification consistently achieves lower error in high-resolution settings, while multi-class becomes favorable at low res-



(a) Detection Accuracy under Spoofing Displacement

(b) FPR under Spoofing Displacement

(c) FNR under Spoofing Displacement

(d) Detection Accuracy under Spoofing Rotation

(e) FPR under Spoofing Rotation

(f) FNR under Spoofing Rotation

Figure 15: Detection performance of DB-PointNet++ under varying spoofing intensities evaluated on a simulated radar dataset. The plots report detection accuracy, FNR, and FPR for both displacement- and rotation-based attacks. Results show that detection accuracy increases with spoofing magnitude—from 64.5% at 2m displacement to 83.5% at 20 m, and from 63.5% at 30° rotation to 80.5% at 90°. This trend arises because stronger perturbations more severely disrupt geometric consistency across multiple radars, thereby enhancing separability between genuine and spoofed point clouds.

olution—especially for pedestrians. A similar resolution-dependent pattern emerges for recall (Fig. 18c) and FNR (Fig. 18d). When data quality is high, the binary setting provides higher recall and fewer misses, indicating well-separated decision boundaries. In contrast, under degraded inputs, the multi-class setting offers more balanced recall and reduced FNR, reflecting greater resilience to low-resolution noise.

Taken together, these results indicate that the superiority of binary versus multi-class classification is conditional, not absolute. Quantitatively, the binary setting achieves substantially higher overall accuracy (about 92%) than the four-class setting (about 70.5%). Although the multi-class accuracy is non-trivial and provides useful attribution, the resulting gap can be critical in safety-centric applications: in autonomous driving, even moderate increases in error translate into unacceptable risk. Accordingly, we position DB-PointNet++ as a first-stage screener in practice: operate the front end in binary mode with high recall to cover potential attacks, and then apply downstream refinement—temporal-consistency and physics-based plausibility checks, multi-sensor fusion, and targeted training strategies (e.g., hard-example mining and class re-weighting for A/C)—to suppress false alarms and stabilize type-level decisions.

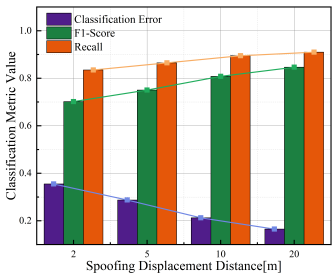
Binary detection delivers the highest top-line accuracy and recall in high-resolution regimes, making it preferable for real-time, safety-critical deployment. Multi-class detection, while less accurate overall, can offer robustness in low-resolution scenarios and provides valuable attribution for offline analysis. A pragmatic deployment strategy is therefore to prioritize binary screening for fast, high-recall coverage, followed by lightweight attribution and verification modules that enforce temporal, geometric, and multi-modal consistency.

B. *Performance Testing of nuScenes Radar Dataset*

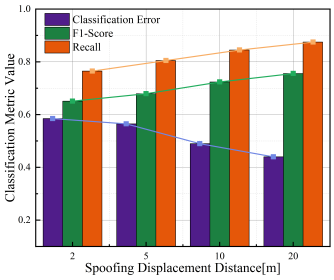
To further examine the generalization capability and real-world applicability of DB-PointNet++, we extended evaluations to the publicly available nuScenes dataset [89]. Developed by Motional (formerly nuTonomy), nuScenes is a large-scale, multi-modal benchmark designed for autonomous driving research. It comprises 1,000 annotated driving scenes, each 20 seconds long, collected from Boston and Singapore—two dense urban environments with complex traffic conditions. The dataset provides diverse scenarios, including variations in weather, lighting, road infrastructure, and traffic behaviors, making it a challenging testbed for perception algorithms under real-world conditions.

In our experiments, we selected three forward-facing radars (Front, Front-Left, Front-Right) from the nuScenes sensor suite, whose spatial configuration—shown in Fig. 19—closely resembles the three-radar setup in our simulated dataset. However, because their fields of view overlap minimally (Fig. 20), the PointNet++&PnP-based registration method could not be applied. Instead, we employed rigid-body transformations to register the Front-Left and Front-Right radars into the Front radar's coordinate frame, enabling consistent cross-sensor fusion despite the limited overlap.

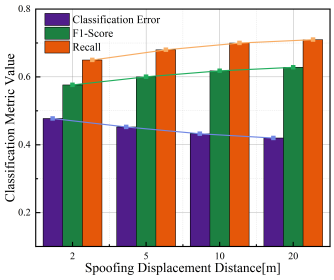
To emulate spoofing, we used Unity to generate high-fidelity radar point clouds of vehicles and pedestrians. Since nuScenes radar data is



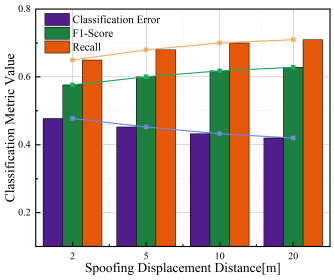
(a) High-Resolution Vehicle Displacement



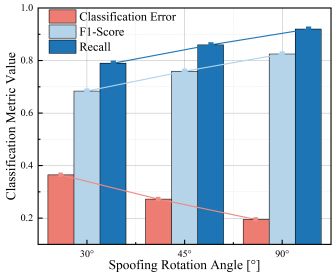
(b) High-Resolution Pedestrian Displacement



(c) Low-Resolution Vehicle Displacement



(d) Low-Resolution Pedestrian Displacement



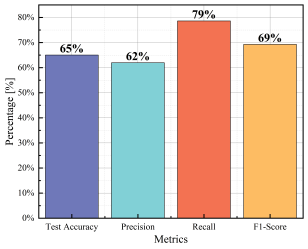
(e) High-Resolution Vehicle Rotation

(f) High-Resolution Pedestrian Rotation

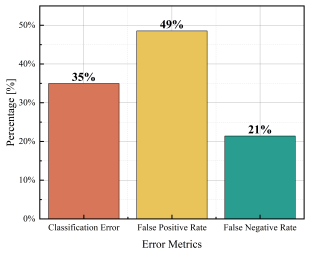
(g) Low-Resolution Vehicle Rotation

(h) Low-Resolution Pedestrian Rotation

Figure 16: Comprehensive performance of DB-PointNet++ under spoofing attacks across targets (vehicles, pedestrians) and radar resolutions (high vs. low) evaluated on a simulated radar dataset. Subplots (a)–(d) report displacement scenarios and (e)–(h) report rotation scenarios; each subplot summarizes classification error, F1-score, and recall. As attack magnitude increases, errors decrease while F1 and recall rise. High-resolution radar consistently outperforms low-resolution, and vehicles are easier to detect than pedestrians. Rotation-based perturbations yield more modest gains than displacement, reflecting subtler geometric changes in the point clouds.



(a) Bar chart of performance metrics: accuracy 65%, precision 62%, recall 79%, and F1-score 69%.



(b) Bar chart of error metrics: classification error 35%, FPR 49%, and FNR 21%.

Figure 17: Performance and error profile of DB-PointNet++ for radar spoofing detection evaluated on a simulated radar dataset.. Subplot (a) summarizes overall performance metrics, reporting accuracy of 65%, precision of 62%, recall of 79%, and F1-score of 69%, reflecting a high-recall/low-precision operating regime. Subplot (b) highlights the error profile, with classification error of 35%, FPR of 49%, and FNR of 21%, indicating that while missed detections are relatively controlled, the false alarm rate remains comparatively high and requires further optimization for safety-critical deployment.

sparse (100–1, 000 points per frame), we applied FPS to downsample the Unity-generated clouds, ensuring density consistency across genuine and spoofed sources. Spoofed targets were injected into two radar streams, while the third radar received a spatially perturbed version (displacement or rotation) to simulate a compromised sensor. Perturbation parameters were aligned with those in the simulated dataset for fair comparison.

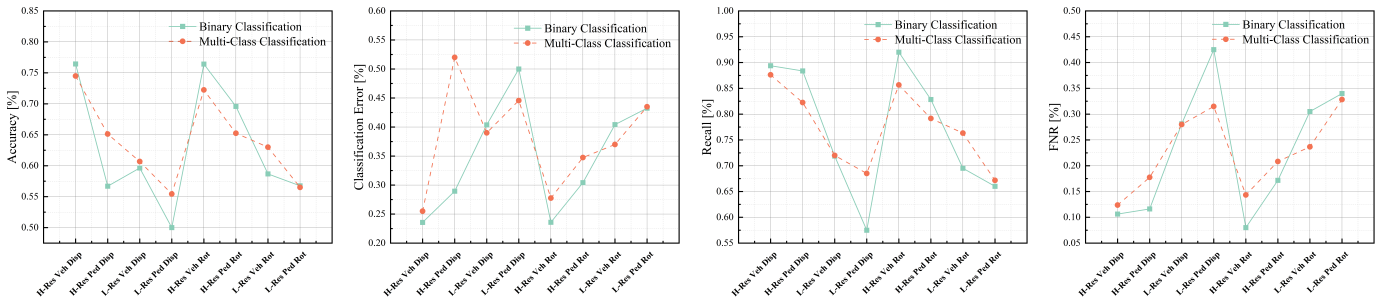
On the simulated dataset, DB-PointNet++ achieves near-ideal performance, confirming its ability to exploit cross-radar consistency under controlled conditions. In contrast, performance drops significantly on nuScenes: Genuine samples reach 79.2% accuracy, while spoofed categories achieve only 47–65% (Fig. 21), exposing a substantial simulation to reality (Sim2Real) generalization gap. The row-normalized confusion matrix shows that A-spoofed is recognized correctly only 47.3% of the time, often confused with B- or C-spoofed, while C-spoofed suffers even higher ambiguity (50.8% correct, ∼30% mislabeled as Genuine or A). B-spoofed performs comparatively better (65.4%), though misclassifications remain around 20%. These results demonstrate limited capability in differentiating attack types in real-world settings.

The performance gap underscores the challenges of Sim2Real transfer. In practice, calibration errors, time synchronization offsets, and complex propagation effects (multipath, scattering) weaken the reliability of cross-radar consistency, leading to degraded robustness.

Fig. 22a presents the overall binary ROC, where the global AUC decreases from 0.836 in simulation to 0.680 on the nuScenes dataset, reflecting a substantial degradation under real-world conditions. The Youden point of nuScenes provides a balanced trade-off between sensitivity and specificity, whereas the recall-oriented point emphasizes safety-critical detection at the expense of higher false-alarm rates.

Fig. 22b further compares displacement and rotation spoofing attacks. Displacement achieves a higher AUC of 0.822 than rotation (0.766), as it introduces more pronounced geometric inconsistencies across radars, while rotation primarily perturbs local surface orientations that are less reliable under measurement noise. Notably, the performance decline from simulation to nuScenes is consistent across both attack types, reinforcing the presence of a Sim2Real domain gap.

Finally, Fig. 22c–e illustrate the per-class ROC curves. Across all spoofing classes, the simulated dataset consistently outperforms



(a) Accuracy (b) Classification Error (c) Recall (d) FNR

Figure 18: Performance comparison between binary and multi-class classification of DB-PointNet++ under spoofing attacks evaluated on a simulated radar dataset.. Subplots (a)–(d) report accuracy, classification error, recall, and FNR separately for vehicles and pedestrians under both high- and low-resolution radar settings. Results show that while binary classification is generally more stable, multi-class classification yields advantages in certain scenarios, such as pedestrian detection at high resolution, thereby highlighting the trade-offs between coarse attack detection and fine-grained spoof type discrimination.

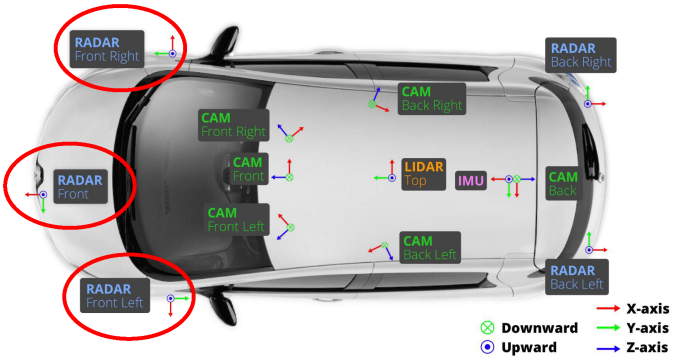


Figure 19: Physical layout of the sensors on the nuScenes autonomous vehicle. The three forward-facing radars (Front, Front-Left, and Front-Right), highlighted in red circles, are selected as the primary sources of radar point cloud data in our experiments. This configuration closely mirrors the simulated three-radar setup used in our simulated dataset, enabling direct cross-validation between simulation and real-world scenarios. By designating the Front radar as the reference and geometrically aligning the other two radars, consistent cross-sensor fusion can be achieved despite the sparse and heterogeneous radar detections in nuScenes.

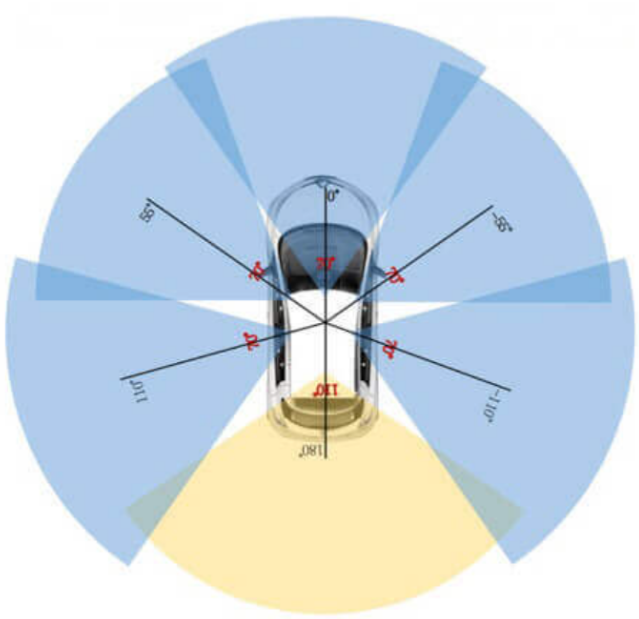
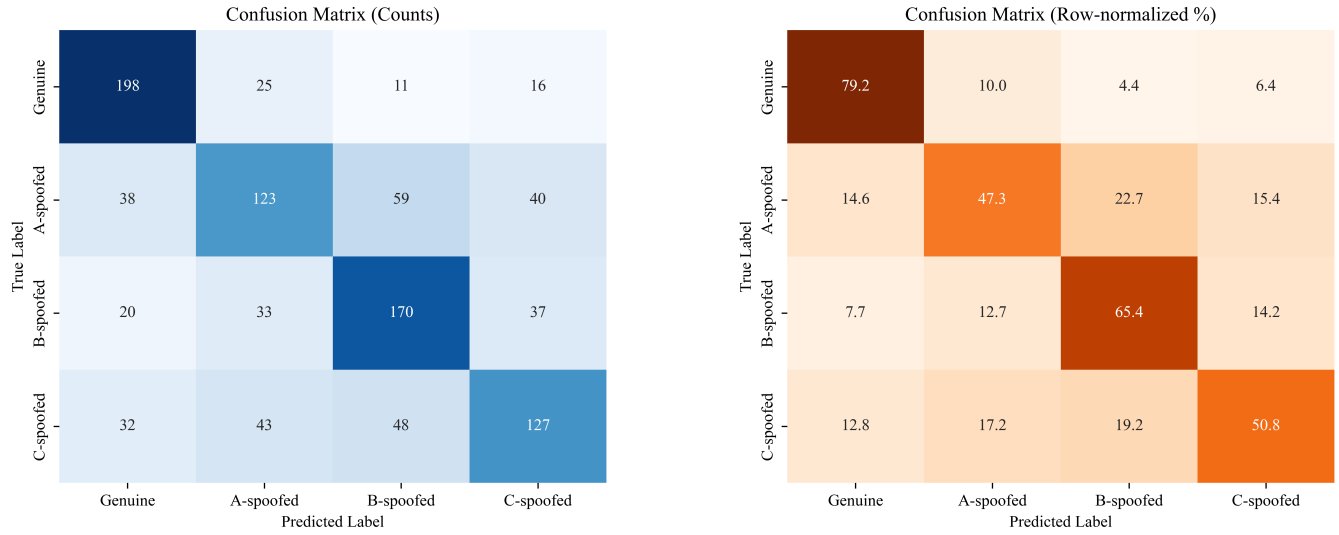


Figure 20: Angular coverage of the forward-facing radars in the nuScenes dataset. The fields of view of the Front, Front-Left, and Front-Right radars exhibit only limited overlap, particularly in the frontal direction. This restriction undermines the applicability of feature-space registration methods (e.g., PointNet++ with PnP alignment), which rely on substantial inter-sensor overlap for robust point cloud matching. Consequently, rigid-body transformations are instead applied to align the radar streams into a unified coordinate frame. The figure highlights the inherent challenge of achieving accurate multi-view fusion in sparse real-world radar setups, in contrast to the idealized overlaps assumed in simulation.

the nuScenes dataset. For example, A-spoofed drops from 0.755 to 0.672, B-spoofed from 0.835 to 0.751, and C-spoofed from 0.767 to 0.701. This systematic degradation highlights the difficulty of transferring models trained in clean, noise-free simulation environments to real radar data characterized by sensor noise, sparse sampling, and imperfect calibration. Among the three classes, B-spoofed maintains the highest separability in both domains, suggesting that vehicle-like targets exhibit more stable geometric patterns and stronger radar reflections, making them less affected by domain shift. Conversely, A- and C-spoofed suffer greater confusion with genuine targets—particularly in nuScenes—underscoring the need for enhanced feature discrimination and improved domain generalization strategies to achieve robust multi-class spoofing detection.

Further evaluation in Fig. 23 confirms that detection improves as perturbation magnitude increases. For vehicles under displacement, accuracy grows from ∼68% (2m) to over 78% (20m), with both FPR and FNR dropping steadily. In contrast, pedestrian spoofing remains challenging, especially under rotation, where recall stays low and FNR/FPR remain near 80% at 30°. With larger perturbations (e.g., 90° rotations), vehicle detection improves substantially (recall >80%), while pedestrian performance, though improved, remains limited due to small radar cross-sections and noisy signatures.

Fig. 24a–b show that both vehicles and pedestrians follow a "larger shift, easier detection" pattern. For vehicles, increasing the displacement from 2 m to 20 m reduces the classification error from 0.42 to 0.29, raises the F1 score from 0.62 to 0.72, and lifts recall from 0.69 to 0.78. These trajectories are smooth and monotonic, indicating stable robustness to displacement magnitude. For pedestrians, classification error declines from 0.31 to 0.24 and F1 improves from



(a) Confusion Matrix (Counts)

(b) Confusion Matrix (Row-normalized %)

Figure 21: Confusion matrices of DB-PointNet++ evaluated on the nuScenes real-world dataset. Subplot (a) shows the absolute prediction counts, while subplot (b) presents the row-normalized percentages, which reveal stronger misclassification tendencies compared to the simulated dataset. In particular, the spoofed classes exhibit lower recognition rates and higher confusion with both genuine and other spoofing categories, highlighting the performance degradation and generalization challenges in real-world scenarios.

0.60 to 0.75; recall peaks at 0.69 around 10 m and then slightly drops to 0.55 at 20 m, suggesting an edge effect at extreme shifts (e.g., distribution sparsity or over-decoupled geometry). To stabilize recall at these extremes while preserving the overall upward trend, displacement-aware thresholds, spatiotemporal cross-radar priors, and trajectory-level constraints are recommended.

Fig. 24c–d indicate that model separability increases consistently as the rotation angle grows from 30° to 90°. For vehicles, the classification error decreases from 0.45 to 0.41, the F1 score rises from 0.60 to 0.65, and recall improves from 0.62 to 0.82. For pedestrians, the improvements are more pronounced: error drops from 0.51 to 0.39, F1 increases from 0.41 to 0.61, and recall climbs from 0.27 to 0.80. These trends indicate that larger rotations more severely disrupt cross-radar geometric consistency, making spoofed point clouds easier to detect. The gain in recall outpacing the gain in F1 also reveals a residual precision–recall tension at high-recall operating points, motivating precision control, confidence calibration, and multi-view consistency regularization to curb false alarms without sacrificing safety.

Despite the sparse and noisy nature of nuScenes radar data, DB-PointNet++ exhibits consistent robustness, particularly for stronger perturbations and rigid targets such as vehicles. At the same time, the marked performance gap between simulation and real-world data underscores the challenges of Sim2Real transfer. Improving feature robustness, incorporating temporal and Doppler cues, and leveraging multi-sensor fusion will be essential for practical deployment in autonomous driving, where the security and reliability of radar perception are critical.

C. Cross-Radar Centroid Distance Detector

To compare with the proposed DB-PointNet++, we introduce an optimistic and computationally efficient baseline, namely the cross-radar centroid distance detector (CRCDD). Given known radar extrinsics and object correspondences, CRCDD computes the centroid of each object for every radar, aligns the centroids from radars A and C to the coordinate frame of radar B, and detects a potential displacement attack when the Euclidean distance between a radar's transformed centroid and the reference centroid (from radar B) exceeds a predefined threshold. This method is simple, interpretable, and serves as a useful geometric baseline for evaluating displacement-type spoofing attacks in multi-radar systems.

The CRCDD is designed as a lightweight geometric defense that relies on radar-level spatial consistency rather than learned representations. When extrinsic calibration between radars is available and object correspondences can be approximately established, the algorithm assesses the approximate geometric consistency of object centroids across radars. Since radar point clouds are inherently sparse and each sensor illuminates different surface patches, exact centroid invariance cannot be guaranteed; however, significant deviations from expected centroid proximity can still indicate potential spoofing.

Assume N_r radars (e.g., A, B, C), each observing an object o with a point cloud $P_i^o = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,n_i}\}$, where $\mathbf{x}_{i,k} \in \mathbb{R}^3$. The algorithm outputs an anomaly score s_o and an alarm flag defined as:

$$\text{alarm}(o) = \begin{cases} 1, & s_o > \tau_{t(o)}, \\ 0, & s_o \leq \tau_{t(o)}, \end{cases} \tag{39}$$

where $\tau_{t(o)}$ denotes the decision threshold specific to the object type $t(o)$ (e.g., vehicle or pedestrian).

Under normal conditions, the centroids of the same object observed by different radars should be spatially close once all observations are transformed into a common coordinate frame. If a displacement spoofing attack occurs, the centroid of the spoofed radar will deviate significantly from the reference centroid:

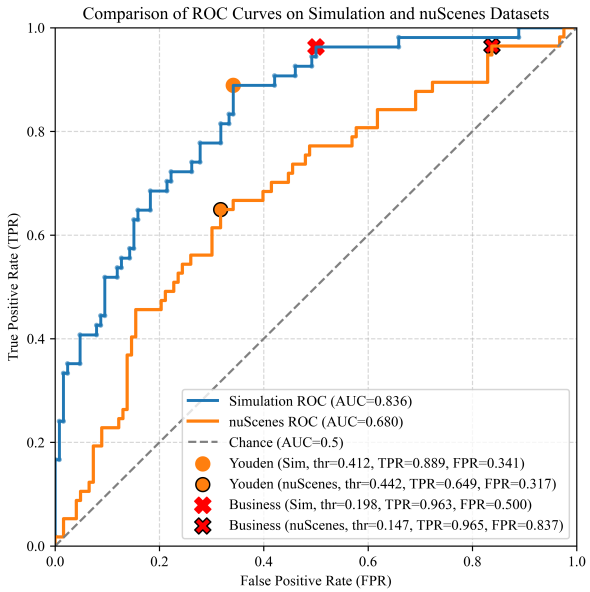
$$\|\tilde{\mathbf{c}}_i^o - \mathbf{c}_B^o\|_2 > \tau_{t(o)}, \tag{40}$$

where $\mathbf{c}_B^o$ denotes the reference centroid in radar B coordinates, and $\tilde{\mathbf{c}}_i^o$ is the transformed centroid from radar i.

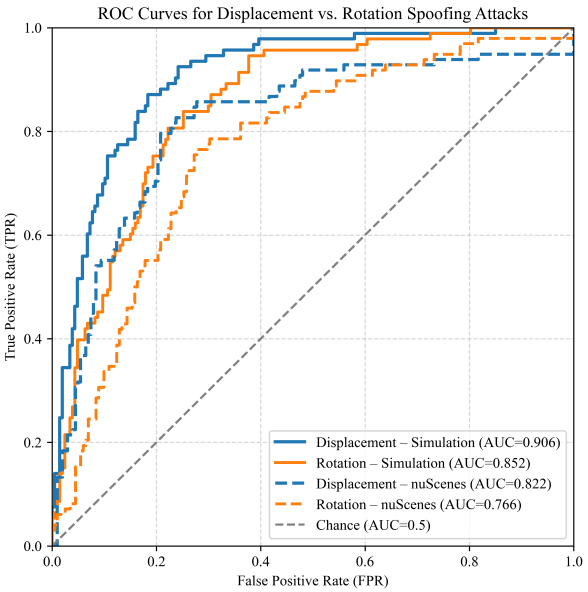
For each radar i and object o, the centroid is computed as:

$$\mathbf{c}_i^o = \frac{1}{|P_i^o|} \sum_{\mathbf{x}_{i,k} \in P_i^o} \mathbf{x}_{i,k}. \tag{41}$$

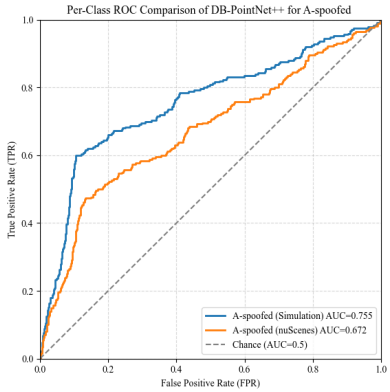
Using known extrinsic transformations $T_{i \to B} = [\mathbf{R}_{i \to B} | \mathbf{t}_{i \to B}]$,



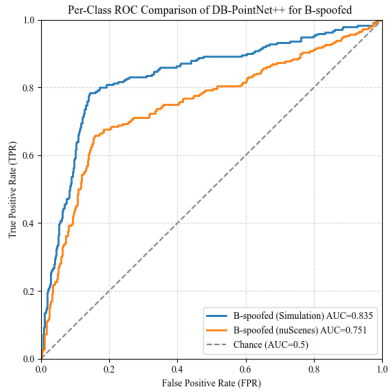
(a) Overall binary spoofing detection



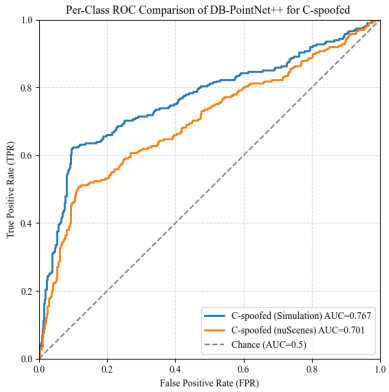
(b) Displacement vs. rotation spoofing detection



(c) Per-class ROC: A-spoofed.



(d) Per-class ROC: B-spoofed.



(e) Per-class ROC: C-spoofed.

Figure 22: ROC curves of DB-PointNet++ evaluated on the nuScenes real-world dataset under different spoofing detection settings. (a) Binary spoofing detection reports an overall AUC of 0.680, with two operating points highlighted: the Youden point ($TPR = 64.9\%$, $FPR = 31.7\%$) balancing sensitivity and specificity, and the business-oriented point ($TPR = 96.5\%$, $FPR = 83.7\%$) prioritizing high recall at the expense of false alarms. (b) Comparison between displacement- and rotation-based attacks demonstrates that displacement is easier to detect ($AUC = 0.822$) than rotation ($AUC = 0.766$), as displacement introduces more severe disruptions to cross-radar geometric consistency. (c-e) One-vs-rest multi-class spoofing detection shows class-dependent separability, with B-spoofed achieving the highest AUC (0.751), followed by C-spoofed (0.701) and A-spoofed (0.672), indicating greater difficulty in discriminating A- and C-spoofed attacks.

the centroids from radars A and C are mapped into radar B's coordinate frame:

$$\tilde{\mathbf{c}}_i^o = \mathbf{R}_{i\to B}\,\mathbf{c}_i^o + \mathbf{t}_{i\to B}, \quad i \in \{A, C\}. \tag{42}$$

The per-radar distance to the reference centroid is:

$$d_i^o = \|\tilde{\mathbf{c}}_i^o - \mathbf{c}_B^o\|_2, \quad i \in \{A, C\}. \tag{43}$$

The overall anomaly score is defined as the maximum spatial deviation:

$$s_o = \max_{i\in\{A,C\}} d_i^o. \tag{44}$$

The threshold $\tau_{t(o)}$ is empirically determined from clean validation data for each object type $t(o)$:

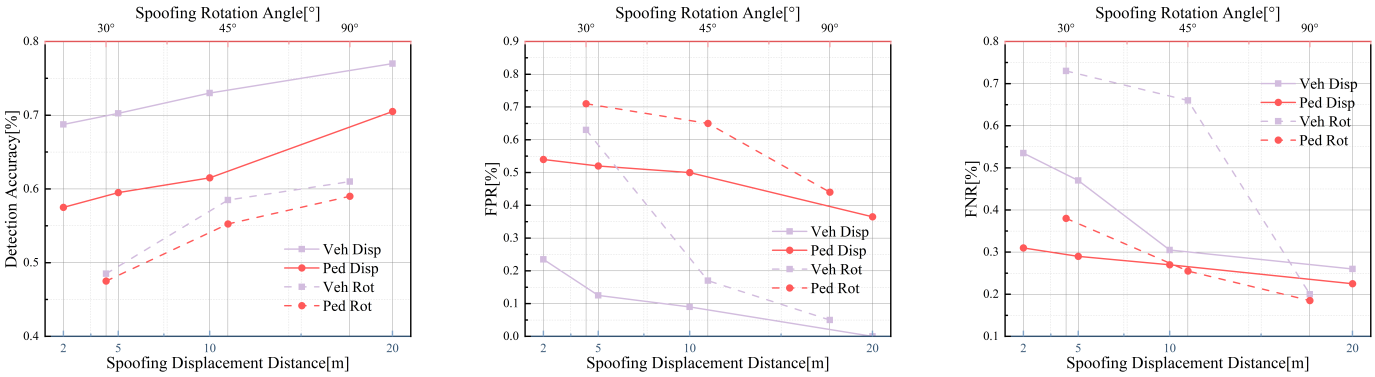
$$\tau_{t(o)} = \text{Percentile}_p(s_o^{\text{clean}}(t(o))), \quad p \in [95, 99]. \tag{45}$$

The complete CRCDD decision rule is therefore summarized as:

$$s_o = \max_{i\in\{A,C\}} \|T_{i\to B}(\mathbf{c}_i^o) - \mathbf{c}_B^o\|_2, \qquad \text{alarm}(o) = \mathbb{I}[s_o > \tau_{t(o)}]. \tag{46}$$

This formulation leverages cross-radar centroid consistency and employs type-specific thresholds to detect displacement attacks in a physically interpretable and computationally efficient manner, , as detailed in Algorithm 2.

Following the description of the CRCDD algorithm, we now

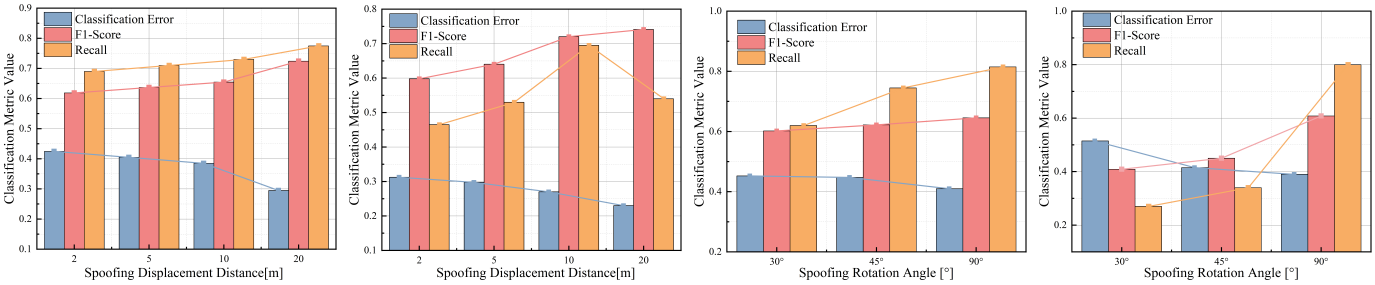


(a) Detection Accuracy under Spoofing Displacement and Rotation

(b) FPR under Spoofing Displacement and Rotation

(c) FNR under Spoofing Displacement and Rotation

Figure 23: Detection performance of DB-PointNet++ on the nuScenes dataset under spoofing attacks with varying displacement distances and rotation angles. Subplots (a)–(d) show detection accuracy, FPR, and FNR for vehicles and pedestrians separately. Results indicate that vehicle spoofing becomes easier to detect with increasing perturbation magnitude, while pedestrian detection remains limited, particularly under rotation-based attacks where both FPR and FNR stay high. Overall, displacement attacks are more readily identifiable than rotations, and DB-PointNet++ demonstrates stronger robustness for rigid targets such as vehicles compared to pedestrians.



(a) Vehicle Displacement

(b) Pedestrian Displacement

(c) Vehicle Rotation

(d) Pedestrian Rotation

Figure 24: Classification error, F1-score, and recall of DB-PointNet++ under spoofing attacks with varying displacement distances and rotation angles evaluated on the nuScenes real-world dataset. Subplots (a)–(d) compare vehicles and pedestrians under displacement- and rotation-based perturbations. Results indicate that vehicle spoofing is generally easier to detect, with lower classification error and higher F1/recall as perturbation magnitude increases. In contrast, pedestrian spoofing remains more challenging, particularly under rotation attacks, underscoring the model's difficulty in handling fine-grained geometric distortions for small, sparse targets. Overall, displacement attacks are more readily identifiable than rotations, highlighting the relative vulnerability of DB-PointNet++ to angular perturbations.

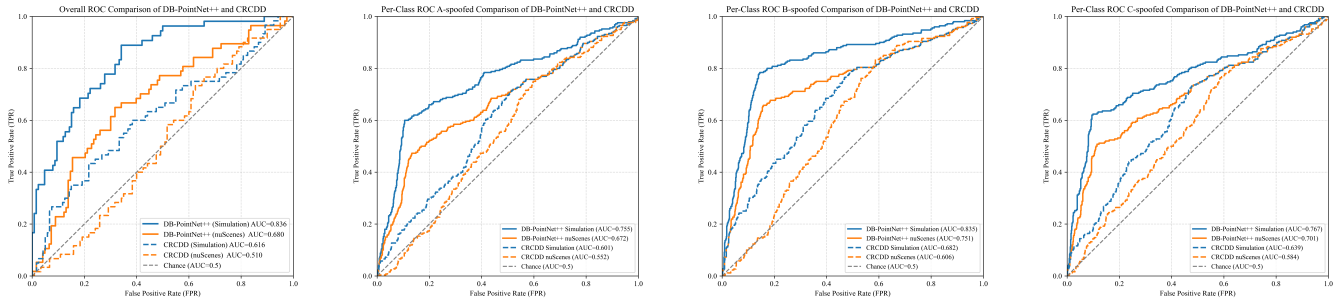
present its experimental performance and compare it with the proposed DB-PointNet++ method. This comparison aims to evaluate the relative effectiveness of the geometric consistency–based baseline (CRCDD) and the learning-based detector (DB-PointNet++) under identical multi-radar spoofing scenarios. In this study, we focus exclusively on displacement-based radar spoofing attacks, where the spoofed radar reports shifted object positions while maintaining otherwise realistic point cloud structures. This restriction allows for a fair and controlled comparison between the two detection paradigms in analyzing geometric deviations caused by spatial misalignment rather than by more complex signal manipulations.

Fig. 25a presents the overall detection performance of CRCDD evaluated on both the simulated and nuScenes datasets. As a purely geometric approach, CRCDD detects potential spoofing attacks by measuring spatial inconsistencies between radar-level object centroids without any learned representation. The results show limited discriminative capability, achieving an AUC of 0.616 on the simulated dataset and 0.510 on the nuScenes dataset. The significant performance drop from simulation to real-world data reveals the sensitivity of CRCDD to measurement noise, calibration errors, and imperfect radar-to-radar correspondences in practice.

When compared with the learning-based DB-PointNet++ model, which achieves an AUC of 0.836 on the same simulated dataset, CRCDD exhibits considerably weaker separability between genuine and spoofed radar measurements. This gap of more than 0.20 in AUC demonstrates that, although centroid-based geometric reasoning can capture coarse spatial deviations, it lacks the feature abstraction and nonlinear decision boundaries learned by DB-PointNet++. Consequently, CRCDD can serve as a transparent and computationally efficient baseline but fails to exploit the complex intra-object spatial structures that DB-PointNet++ leverages for accurate spoofing discrimination.

Under real-world conditions, the performance gap becomes even more pronounced. DB-PointNet++ maintains an AUC of 0.680 on the nuScenes dataset, whereas CRCDD drops to 0.510, nearly approaching random-guessing performance. This degradation confirms that geometric consistency alone is insufficient to counteract domain shifts caused by sensor noise, imperfect synchronization, and environmental clutter. In contrast, DB-PointNet++ demonstrates stronger generalization through learned spatial representations and dual-supervision regularization, highlighting the robustness of learning-based approaches in cross-domain spoofing detection.

Fig. 25b–25d further present the per-class ROC performance of CRCDD on both datasets. As a geometry-only method, CRCDD



(a) Overall binary spoofing detection (b) Per-class ROC: A-spoofed. (c) Per-class ROC: B-spoofed. (d) Per-class ROC: C-spoofed.

Figure 25: ROC analysis of the proposed CRCDD on both simulated and real-world datasets. (a) The overall binary spoofing detection performance shows limited separability, with AUC values of 0.616 on the simulated dataset and 0.510 on the nuScenes dataset, indicating notable degradation under real-world conditions. (b-d) The per-class ROC curves further highlight the reduced discriminability of CRCDD across different spoofing types, where B-spoofed achieves the highest AUC (0.682 on simulation, 0.606 on nuScenes), followed by C- and A-spoofed classes. Compared with learning-based models, CRCDD remains interpretable and computationally lightweight but is more sensitive to noise, calibration errors, and imperfect object correspondences in real environments.

relies solely on centroid-level spatial consistency across radars rather than feature learning. On the simulated dataset, the AUCs for A-, B-, and C-spoofed categories are 0.601, 0.682, and 0.639, respectively, suggesting only moderate separability. Among them, the B-spoofed class achieves the highest performance, likely because displacement perturbations in this configuration cause more pronounced centroid shifts, while A- and C-spoofed attacks remain harder to identify due to smaller geometric deviations. When tested on the nuScenes dataset, all three classes exhibit reduced discriminability, with AUCs dropping to 0.552, 0.606, and 0.584. This decline highlights CRCDD's vulnerability to real-world imperfections, where ideal geometric alignment rarely holds due to calibration drift and dynamic environmental noise.

When compared with DB-PointNet++ on the same simulated dataset, the advantage of learned representations becomes evident. DB-PointNet++ achieves AUCs of 0.755, 0.835, and 0.767 for A-, B-, and C-spoofed categories, outperforming CRCDD by roughly 0.15–0.20 in AUC across all classes. This margin demonstrates that deep spatial feature learning effectively enhances class-wise separability even under controlled simulated conditions. While CRCDD captures only macroscopic centroid displacement, DB-PointNet++ extracts finer intra-object features and contextual cues that improve discrimination between genuine and spoofed radar point clouds.

Under real-world conditions, this difference persists. DB-PointNet++ achieves per-class AUCs of 0.672 (A-spoofed), 0.751 (B-spoofed), and 0.701 (C-spoofed), maintaining robust performance despite environmental uncertainty. In contrast, CRCDD's results drop to 0.552, 0.606, and 0.584, approaching random-level separability. These findings confirm that geometric consistency–based detectors cannot effectively handle complex domain shifts or sensor imperfections, whereas deep learning–based methods can learn invariant representations and adapt to heterogeneous radar configurations.

In summary, while the proposed CRCDD offers interpretability, efficiency, and serves as a meaningful geometric baseline, its reliance on centroid-level consistency limits its detection accuracy and generalization capability. The learning-based DB-PointNet++, by contrast, benefits from hierarchical feature extraction and dual-supervised training, enabling it to capture subtle geometric distortions and maintain robustness across both simulated and real-world domains. Overall, these comparative results emphasize that deep learning approaches, though computationally heavier, provide a more practical and resilient solution for multi-radar spoofing detection in realistic driving environments.

D. Performance Evaluation under Minor Perturbation Attacks

To rigorously evaluate the robustness of the proposed model against fine-grained geometric perturbations in realistic settings, we design a gradient-based point cloud perturbation framework [90]. This framework operates on radar point clouds from the nuScenes dataset that already include embedded spoofing targets, and applies localized transformations—namely slight rotations, small translations, and uniform scalings—on predefined groups of points. These perturbations are carefully constrained to preserve the overall geometric structure while inducing targeted misclassification.

In constructing the perturbed dataset, we follow the single-compromised-radar assumption, where only one radar is adversarially manipulated at a time while the remaining two function normally. High-fidelity radar point clouds of vehicles and pedestrians are generated using Unity to simulate spoofed echoes. Because the native nuScenes radar data are inherently sparse (typically 100–1, 000 detections per frame), FPS is applied to the dense Unity-generated point clouds to ensure consistent point density across sources. The sampled spoofed targets are then injected into two of the three radar streams, while the third stream receives a spatially transformed version—either displaced or rotated—and is further augmented with minor perturbations to emulate the compromised sensor. This procedure yields a perturbed extension of the nuScenes dataset that incorporates minor perturbation attacks for subsequent robustness evaluation.

Formally, each input point cloud is partitioned into a set of local groups, with each group associated with a set of trainable transformation parameters. An adversarial objective function is defined by combining a misclassification-oriented loss term with geometric regularization constraints. These transformation parameters are optimized via gradient descent to maximize classification error while minimizing geometric distortion.

During optimization, the point cloud is iteratively updated until convergence or early stopping criteria are met. The resulting adversarial samples are then saved and evaluated using the DB-PointNet++ model. This framework provides a principled approach to crafting localized geometric attacks and supplies reliable adversarial data for downstream robustness analysis.

- **Perturbation Modeling**

Algorithm 2: Cross-Radar Centroid Distance Detector (CR-CDD)

Input: Per-frame per-radar point clouds $\{P_i\}_{i\in\{A,B,C\}}$; object correspondence list $\mathcal{O}$; extrinsic transforms $T_{i\to B}$; **type-specific thresholds** $\{\tau_t\}$ with $t \in$ Types.

Output: Per-object anomaly flag alarm$(o) \in \{\text{True}, \text{False}\}$ and anomaly score s_o.

foreach *object* $o \in \mathcal{O}$ **do**
  Let $V \leftarrow \{\, i \in \{A,B,C\} \mid P_i^o \neq \emptyset \,\}$;
  if $|V| < 2$ **then**
    alarm$(o) \leftarrow$ False, $s_o \leftarrow 0$; **continue**;
  `// Centroids in native radar frames`
  foreach $i \in V$ **do**
    $\mathbf{c}_i^o \leftarrow \frac{1}{|P_i^o|}\sum_{\mathbf{x}\in P_i^o}\mathbf{x}$;
  `// Map to radar-`B` coordinates when needed`
  foreach $i \in V \setminus \{B\}$ **do**
    $\tilde{\mathbf{c}}_i^o \leftarrow T_{i\to B}(\mathbf{c}_i^o)$;
  `// Choose reference centroid`
  if $B \in V$ **then**
    $\mathbf{c}_{\text{ref}}^o \leftarrow \mathbf{c}_B^o$, $U \leftarrow V \setminus \{B\}$;
  else
    pick any $j \in V$; $\mathbf{c}_{\text{ref}}^o \leftarrow \tilde{\mathbf{c}}_j^o$; $U \leftarrow V \setminus \{j\}$;
  `// Per-radar distances in the common frame`
  foreach $i \in U$ **do**
    $d_i^o \leftarrow \left\|\tilde{\mathbf{c}}_i^o - \mathbf{c}_{\text{ref}}^o\right\|_2$;
  `// Max aggregation (used in our implementation)`
  $s_o \leftarrow \max_{i\in U} d_i^o$;
  `// Type-specific thresholding`
  alarm$(o) \leftarrow \left(s_o > \tau_{t(o)}\right)$;
  output (alarm(o), s_o);
return *alarms and scores*

Let the input point cloud be denoted by $\mathbf{P} = \{\mathbf{x}_i \in \mathbb{R}^3\}_{i=1}^N$, where each point $\mathbf{x}_i = [x_i, y_i, z_i]^T$ represents a 3D coordinate. To introduce localized and structured perturbations, $\mathbf{P}$ is partitioned into G disjoint groups $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_G$ using either frame-index-based slicing or PCA-driven segmentation.

For each group $\mathcal{G}_g$, we define a parameter set:

- $\theta_g \in \mathbb{R}$: rotation angle around the z-axis,
- $s_g \in \mathbb{R}^+$: isotropic scaling factor,
- $\delta_{x,g}, \delta_{y,g} \in \mathbb{R}$: translation along the x and y axes.

The transformation applied to each point $\mathbf{x}_i \in \mathcal{G}_g$ proceeds as:

1) Compute the geometric center of the group:

$$\mathbf{c}_g = \frac{1}{|\mathcal{G}_g|}\sum_{\mathbf{x}_i\in\mathcal{G}_g}\mathbf{x}_i \tag{47}$$

2) Apply scaling:

$$\mathbf{x}_i' = s_g \cdot (\mathbf{x}_i - \mathbf{c}_g) \tag{48}$$

3) Apply 2D rotation in the x–y plane:

$$\mathbf{R}_g = \begin{bmatrix} \cos\theta_g & -\sin\theta_g & 0 \\ \sin\theta_g & \cos\theta_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{49}$$

$$\mathbf{x}_i'' = \mathbf{R}_g \cdot \mathbf{x}_i' \tag{50}$$

4) Apply translation and restore original center:

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i'' + \mathbf{c}_g + [\delta_{x,g}, \delta_{y,g}, 0]^T \tag{51}$$

5) Clip the total displacement to maintain imperceptibility:

$$\|\tilde{\mathbf{x}}_i - \mathbf{x}_i\|_2 \leq \varepsilon, \quad \varepsilon = 0.2 \tag{52}$$

- **Loss Function Design**

To achieve adversarial misclassification while maintaining geometric plausibility, we define the overall loss function as:

$$\mathcal{L}_{\text{attack}} = \mathcal{L}_{\text{cls}} + \lambda_\theta \mathcal{R}_\theta + \lambda_t \mathcal{R}_t + \lambda_s \mathcal{R}_s \tag{53}$$

where the individual terms are defined as follows:

- Classification Loss:

$$\mathcal{L}_{\text{cls}} = \max\left(f_y - \max_{j\neq y} f_j, -\kappa\right) \tag{54}$$

  where f_y is the predicted logit for the true class y, and f_j is the logit for class $j \neq y$. κ is a confidence margin controlling the strength of misclassification.
- Rotation Regularization:

$$\mathcal{R}_\theta = \sum_{g=1}^{G}\theta_g^2 \tag{55}$$

- Translation Regularization:

$$\mathcal{R}_t = \sum_{g=1}^{G}(\delta_{x,g}^2 + \delta_{y,g}^2) \tag{56}$$

- Scaling Regularization:

$$\mathcal{R}_s = \sum_{g=1}^{G}(s_g - 1)^2 \tag{57}$$

Here, λ_θ, λ_t, and λ_s are non-negative hyperparameters that control the trade-off between attack success and perturbation magnitude.

- **Optimization Procedure**

All transformation parameters are initialized to identity values: $\theta_g = 0$, $s_g = 1$, and $\delta_{x,g} = \delta_{y,g} = 0$. During each iteration, the attack loss $\mathcal{L}_{\text{attack}}$ is computed and minimized using the Adam optimizer with learning rate η:

$$\phi^{(t+1)} = \phi^{(t)} - \eta \cdot \nabla_\phi \mathcal{L}_{\text{attack}} \tag{58}$$

where ϕ denotes the full set of transformation parameters.

The optimization is terminated early if the classifier predicts a label different from the original ground truth, indicating a successful adversarial perturbation, as detailed in Algorithm 3.

After applying the group-wise perturbation algorithm to the nuScenes dataset, we evaluated the performance of DB-PointNet++ under subtle geometric disturbances. Fig. 26 illustrates the evolution of training and testing instance accuracy, along with corresponding loss values, over 200 training epochs.

Both the training and test losses exhibit a rapid initial decline and converge after approximately 100 epochs, indicating that the model can fit the perturbed samples and generalize to some extent. However, the converged loss values remain relatively high—around 1.0 for training and 0.9 for testing—suggesting that the model still struggles to confidently discriminate between classes under perturbation.

In terms of classification accuracy, the training instance accuracy plateaus at approximately 46%, while the test accuracy stabilizes slightly higher at around 52%. Given that this is a four-class classification task (with a random-guessing baseline of 25%), the performance is modest. While the results exceed random prediction,

Algorithm 3: Group-wise Geometric Perturbation Attack

Input: Point cloud $\mathbf{P} = \{\mathbf{x}_i\}_{i=1}^{N}$, true label y, classifier $Z(\cdot)$, perturbation budget ε, learning rate η, max iterations T

Output: Perturbed point cloud $\tilde{\mathbf{P}}$

Group Partition: divide $\mathbf{P}$ into G groups $\mathcal{G}_1, \ldots, \mathcal{G}_G$;
Initialize: for all g, set $\theta_g \leftarrow 0$, $s_g \leftarrow 1$, $\delta_{x,g}, \delta_{y,g} \leftarrow 0$;
for $t = 1$ **to** T **do**
  foreach $g = 1$ **to** G **do**
    Compute group centroid: $\mathbf{c}_g \leftarrow \frac{1}{|\mathcal{G}_g|} \sum_{\mathbf{x}_i \in \mathcal{G}_g} \mathbf{x}_i$;
    foreach $\mathbf{x}_i \in \mathcal{G}_g$ **do**
      Apply scaling: $\mathbf{x}_i' \leftarrow s_g \cdot (\mathbf{x}_i - \mathbf{c}_g)$;
      Apply rotation: $\mathbf{R}_g \leftarrow \begin{bmatrix} \cos\theta_g & -\sin\theta_g & 0 \\ \sin\theta_g & \cos\theta_g & 0 \\ 0 & 0 & 1 \end{bmatrix}$;
      $\mathbf{x}_i'' \leftarrow \mathbf{R}_g \cdot \mathbf{x}_i'$;
      Apply translation and restore center: $\tilde{\mathbf{x}}_i \leftarrow \mathbf{x}_i'' + \mathbf{c}_g + [\delta_{x,g}, \delta_{y,g}, 0]^T$;
      Clip displacement: **if** $\|\tilde{\mathbf{x}}_i - \mathbf{x}_i\|_2 > \varepsilon$ **then**
        Scale displacement to enforce ε-limit
  Aggregate perturbed point cloud: $\tilde{\mathbf{P}} \leftarrow \{\tilde{\mathbf{x}}_i\}_{i=1}^{N}$;
  Forward pass: $f = Z(\tilde{\mathbf{P}})$;
  Compute loss: $\mathcal{L}_{\text{attack}} \leftarrow \max(f_y - \max_{j \neq y} f_j, -\kappa) + \lambda_\theta \sum_g \theta_g^2 + \lambda_t \sum_g (\delta_{x,g}^2 + \delta_{y,g}^2) + \lambda_s \sum_g (s_g - 1)^2$;
  Update parameters $\{\theta_g, s_g, \delta_{x,g}, \delta_{y,g}\}$ via Adam optimizer with η;
  if $\arg\max_j f_j \neq y$ **then**
    break;
    ; // Terminate if attack is successful
return $\tilde{\mathbf{P}}$

they fall short of what would be considered acceptable for real-world deployment.

A closer inspection reveals why such minor perturbations can still impose substantial difficulty. The injected geometric noise interferes with the initial assignment of point-level labels during feature abstraction, subtly altering local neighborhoods and occasionally shifting points across decision boundaries in the learned feature space. At the same time, the perturbations slightly modify global density statistics—features explicitly leveraged by the auxiliary loss in DB-PointNet++—thereby weakening the model's ability to exploit density consistency as a discriminative cue. These dual effects compound, resulting in degraded separability across spoofed and genuine classes even when the perturbations appear visually insignificant.

Notably, the absence of a significant accuracy gap between training and testing suggests that the model does not overfit to the perturbed data. Instead, the limited accuracy appears to stem from reduced input discriminability and insufficient robustness of the model to small-scale geometric variations.

In summary, the experiment confirms that the proposed perturbation strategy imposes a substantial challenge to DB-PointNet++, significantly degrading its classification performance. The results highlight the model's limited robustness to fine-grained structural distortions and density fluctuations, underscoring the need for more geometry-aware architectures or stronger regularization mechanisms to improve resilience in real-world conditions.

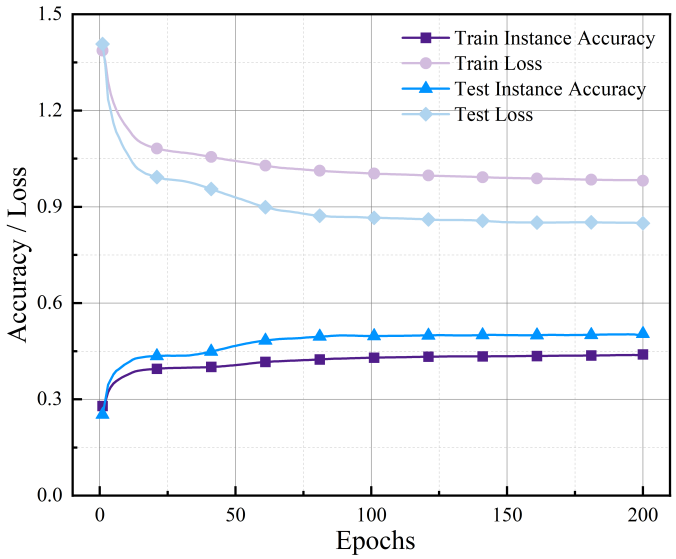


Figure 26: Training and testing performance of the model on the perturbed NuScenes dataset. The purple and light blue curves represent training and test losses, while the dark purple and dark blue curves indicate training and test instance accuracies, respectively. The metrics converge over time, but overall performance remains moderate.

VI. Conclusion

This paper proposed a multi-radar spoofing detection framework that integrates PointNet++ with PnP point cloud registration, addressing the challenges of multi-view alignment and robust feature extraction. On this basis, we developed DB-PointNet++, which incorporates DBSCAN clustering and density-aware feature enhancement to generate more discriminative point cloud representations. The proposed framework effectively captures structural consistency while highlighting spoofing-induced anomalies, providing a novel approach for multi-radar spoofing detection in autonomous driving scenarios.

Experimental results on the simulated dataset demonstrated strong performance. The binary classification task (genuine vs. spoofed) achieved an accuracy of about 92%, with an ROC AUC of 0.836, and reached TPR=88.9% and FPR=34.1% at the optimal operating point. In comparison, the four-class task (genuine, A/B/C spoofed) yielded an overall accuracy of 70.5%, where Genuine and B-spoofed achieved the highest recognition rates (82% and 78%), while A- and C-spoofed remained confusable. These findings indicate that binary classification provides better stability and accuracy, whereas four-class classification offers valuable source attribution.

Further validation on the nuScenes real-world dataset revealed performance degradation under sparse radar conditions. Binary detection accuracy dropped to about 52% with an AUC of 0.680; in the four-class setting, B-spoofed achieved an AUC of 0.751, while A- and C-spoofed only reached 0.672 and 0.701. These results highlight that while the framework shows strong robustness in high-resolution simulated scenarios, real-world conditions pose challenges such as sparsity, noise, and sensor limitations.

In conclusion, the proposed framework demonstrates the feasibility of combining PointNet++ feature extraction with PnP-based registration for multi-radar spoofing detection. Binary detection proves more suitable for real-time, safety-critical deployment, while four-class classification, though less accurate, provides spoof-source attribution useful for downstream defense strategies. Future work will focus on three directions: (1) integrating multi-sensor fusion (e.g., camera

and LiDAR) to alleviate radar sparsity, (2) incorporating temporal consistency constraints to enhance robustness in dynamic scenarios, and (3) leveraging adversarial training and hard-sample mining to improve generalization under real-world spoofing attacks.

REFERENCES

[1] A. Ziebinski, R. Cupek, D. Grzechca, and L. Chruszczyk, "Review of advanced driver assistance systems (adas)," in *AIP Conference Proceedings*, vol. 1906, no. 1. AIP Publishing LLC, 2017, p. 120002.

[2] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley, "Advanced driver-assistance systems: A path toward autonomous vehicles," *IEEE Consumer Electronics Magazine*, vol. 7, no. 5, pp. 18–25, 2018.

[3] J. Piao and M. McDonald, "Advanced driver assistance systems from autonomous to cooperative approach," *Transport reviews*, vol. 28, no. 5, pp. 659–684, 2008.

[4] A. Shaout, D. Colella, and S. Awad, "Advanced driver assistance systems-past, present and future," in *2011 Seventh International Computer Engineering Conference (ICENCO'2011)*. IEEE, 2011, pp. 72–82.

[5] F. Jiménez, J. E. Naranjo, J. J. Anaya, F. García, A. Ponz, and J. M. Armingol, "Advanced driver assistance system for road environments to improve safety and efficiency," *Transportation research procedia*, vol. 14, pp. 2245–2254, 2016.

[6] I. Bilik, "Comparative analysis of radar and lidar technologies for automotive applications," *IEEE Intell. Transp. Syst. Mag.*, vol. 15, no. 1, pp. 244–269, 2023.

[7] A. S. Mohammed, A. Amamou, F. K. Ayevide, S. Kelouwani, K. Agbossou, and N. Zioui, "The perception system of intelligent ground vehicles in all weather conditions: A systematic literature review," *Sensors*, vol. 20, no. 22, p. 6532, 2020.

[8] R. Al Hasnawi and I. Marghescu, "A survey of vehicular vlc methodologies," *Sensors*, vol. 24, no. 2, p. 598, 2024.

[9] I. Bilik, O. Longman, S. Villeval, and J. Tabrikian, "The rise of radar for autonomous vehicles: Signal processing solutions and future research directions," *IEEE Signal Processing Magazine*, vol. 36, no. 5, pp. 20–31, 2019.

[10] I. Bilik, O. Bialer, S. Villeval, H. Sharifi, K. Kona, M. Pan, D. Persechini, M. Musni, and K. Geary, "Automotive MIMO radar for urban environments," in *IEEE Radar Conf.*, 2016.

[11] M. Murad, I. Bilik, M. Friesen, J. Nickolaou, J. Salinger, K. Geary, and J. S. Colburn, "Requirements for next generation automotive radars," in *IEEE Radar Conf.*, 2013.

[12] V. H. Le, J. den Hartog, and N. Zannone, "Security and privacy for innovative automotive applications: A survey," *Computer Communications*, vol. 132, pp. 17–41, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014036641731174X

[13] V. K. Kukkala, S. V. Thiruloga, and S. Pasricha, "Roadmap for cybersecurity in autonomous vehicles," *IEEE Consumer Electronics Magazine*, vol. 11, no. 6, pp. 13–23, 2022.

[14] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2898–2915, 2017.

[15] A. F. Martone, K. I. Ranney, K. Sherbondy, K. A. Gallagher, and S. D. Blunt, "Spectrum allocation for noncooperative radar coexistence," *IEEE Trans. on Aero. and Electr. Sys.*, vol. 54, no. 1, pp. 90–105, 2018.

[16] D. Benvenuti, P. Addabbo, G. Giunta, G. Foglia, and D. Orlando, "Eccm strategies for radar systems against smart noise-like jammers," *IEEE Transactions on Signal Processing*, vol. 72, pp. 3912–3926, 2024.

[17] J. Sun, Y. Yuan, M. Sabrina Greco, and F. Gini, "Coordinated deception jamming power scheduling for multijammer systems against distributed radar systems," *IEEE Transactions on Radar Systems*, vol. 2, pp. 1076–1088, 2024.

[18] J. Sun, Y. Yuan, M. S. Greco, F. Gini, X. Yang, and W. Yi, "Anti-deception jamming resource scheduling for multi-target tracking in distributed radar networks," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–17, 2024.

[19] Z. Yu, H. Gao, X. Cong, N. Wu, and H. H. Song, "A survey on cyber–physical systems security," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 670–21 686, 2023.

[20] A. Pandharipande, C.-H. Cheng, J. Dauwels, S. Z. Gurbuz, J. Ibanez-Guzman, G. Li, A. Piazzoni, P. Wang, and A. Santra, "Sensing and machine learning for automotive perception: A review," *IEEE Sensors Journal*, vol. 23, no. 11, pp. 11 097–11 115, 2023.

[21] S. Z. Gurbuz, *Deep Neural Network Design for Radar Applications*. SciTech Publishing, 2020.

[22] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, p. 100057, 2021.

[23] C. Waldschmidt, J. Hasch, and W. Menzel, "Automotive radar—from first efforts to future systems," *IEEE Journal of Microwaves*, vol. 1, no. 1, pp. 135–148, 2021.

[24] F. Engels, P. Heidenreich, M. Wintermantel, L. Stäcker, M. Al Kadi, and A. M. Zoubir, "Automotive radar signal processing: Research directions and practical challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 865–878, 2021.

[25] B. Kang, J. Kweon, M. Rangaswamy, and V. Monga, "Deep learning for radar waveform design: Retrospectives and the road ahead," in *2023 IEEE International Radar Conference (RADAR)*. IEEE, 2023, pp. 1–6.

[26] T. Diskin, Y. Beer, U. Okun, and A. Wiesel, "Cfarnet: Deep learning for target detection with constant false alarm rate," *Signal Processing*, vol. 223, p. 109543, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168424001622

[27] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.

[28] S. Moosavi-Dezfooli, A. Fawzi *et al.*, "Universal adversarial perturbations," 2017.

[29] J. Wang, X. Liu, J. Hu, D. Wang, S. Wu, T. Jiang, Y. Guo, A. Liu, and J. Zhou, "Adversarial examples in the physical world: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2311.01473

[30] D. Brodeski, I. Bilik, and R. Giryes, "Deep radar detector," in *2019 IEEE Radar Conference (RadarConf)*. IEEE, 2019, pp. 1–6.

[31] J. F. Tilly, S. Haag, O. Schumann, F. Weishaupt, B. Duraisamy, J. Dickmann, and M. Fritzsche, "Detection and tracking on automotive radar data with deep learning," in *Proc. IEEE FUSION*, 2020.

[32] I. Roldan, A. Palffy, J. F. P. Kooij, D. M. Gavrila, F. Fioranelli, and A. Yarovoy, "A deep automotive radar detector using the radelft dataset," 2024. [Online]. Available: https://arxiv.org/abs/2406.04723

[33] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.

[34] J. Fuchs, M. Gardill, M. Lübke, A. Dubey, and F. Lurz, "A machine learning perspective on automotive radar direction of arrival estimation," *IEEE access*, vol. 10, pp. 6775–6797, 2022.

[35] B. K. Chalise, D. M. Wong, M. G. Amin, A. F. Martone, and B. H. Kirk, "Detection, mode selection, and parameter estimation in distributed radar networks: Algorithms and implementation challenges," *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 11, pp. 4–22, 2022.

[36] M. Hassan, F. Fioranelli, A. Yarovoy, and S. Ravindran, "Radar multi object tracking using dnn features," in *2023 IEEE International Radar Conference (RADAR)*, 2023, pp. 1–6.

[37] A. Stroescu, M. Cherniakov, and M. Gashinova, "Classification of high resolution automotive radar imagery for autonomous driving based on deep neural networks," in *2019 20th International Radar Symposium (IRS)*. IEEE, 2019, pp. 1–10.

[38] N. Pandey and S. S. Ram, "Classification of automotive targets using inverse synthetic aperture radar images," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 675–689, 2022.

[39] M. S. Seyfioğlu, A. M. Özbayoğlu, and S. Z. Gürbüz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1709–1723, 2018.

[40] P. Itkin and N. Levanon, "Ambiguity function based radar waveform classification and unsupervised adaptation using deep cnn models," in *2019 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)*, 2019, pp. 1–6.

[41] A. Kloukiniotis, A. Papandreou, A. Lalos, P. Kapsalas, D.-V. Nguyen, and K. Moustakas, "Countering adversarial attacks on autonomous vehicles using denoising techniques: A review," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 61–80, 2022.

[42] Y. Deng, T. Zhang, G. Lou, X. Zheng, J. Jin, and Q.-L. Han, "Deep learning-based autonomous driving systems: A survey of attacks and defenses," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 7897–7912, 2021.

[43] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *Cryptographic Hardware and Embedded Systems–CHES 2017: 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*. Springer, 2017, pp. 445–467.

[44] B. Yang, Z. Jin, Y. Cheng, X. Ji, and W. Xu, "Adversarial robustness analysis of lidar-included models in autonomous driving," *High-Confidence Computing*, p. 100203, 2024.

[45] H. Ren, T. Huang, and H. Yan, "Adversarial examples: attacks and defenses in the physical world," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3325–3336, 2021.

[46] Z. Geng, H. Yan, J. Zhang, and D. Zhu, "Deep-learning for radar: A survey," *IEEE Access*, vol. 9, pp. 141 800–141 818, 2021.

[47] A. Zafar, A. Khan, and S. Younis, "Classical adversarial attack on mm-wave fmcw radar," in *2021 International Conference on Frontiers of Information Technology (FIT)*, 2021, pp. 281–286.

[48] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2014.

[49] J. Choi and S. J. Lee, "Consistency index-based sensor fault detection system for nuclear power plant emergency situations using an lstm network," *Sensors (Basel, Switzerland)*, vol. 20, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:213187025

[50] T. S. Delwar, U. Aras, S. Mukhopadhyay, A. Kumar, U. Kshirsagar, Y. Lee, M. Singh, and J.-Y. Ryu, "The intersection of machine learning and wireless sensor network security for cyber-attack detection: a detailed analysis," *Sensors*, vol. 24, no. 19, p. 6377, 2024.

[51] E. Onyejegbu, A. Lee, A. Ashimbayeva, B. Nakarmi, and I. A. Ukaegbu, "Analysis and mitigation of interference in a multi-radar environment," in *IEEE Electron Devices Technology & Manufacturing Conference*, 2023.

[52] R. Komissarov and A. Wool, "Spoofing attacks against vehicular fmcw radar," in *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security*, ser. ASHES '21, New York, NY, USA, 2021, p. 91–97. [Online]. Available: https://doi.org/10.1145/3474376.3487283

[53] M. A. Vu, W. C. Headley, and K. P. Heaslip, "A comparative overview of automotive radar spoofing countermeasures," in *IEEE International Conference on Cyber Security and Resilience*, 2022, pp. 245–252.

[54] Unity Technologies, "Unity real-time development platform," https://unity.com/, 2025, version 2022.3 LTS, accessed Oct. 11, 2025.

[55] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Red Hook, NY, USA, 2017, p. 5105–5114.

[56] Y. Chen, G. Liu, Y. Xu, P. Pan, and Y. Xing, "Pointnet++ network architecture with individual point level and global features on centroid for als point cloud classification," *Remote Sensing*, vol. 13, no. 3, p. 472, 2021.

[57] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, "Pointnext: Revisiting pointnet++ with improved training and scaling strategies," *Advances in neural information processing systems*, vol. 35, pp. 23 192–23 204, 2022.

[58] M. Hahsler, M. Piekenbrock, and D. Doran, "dbscan: Fast density-based clustering with R," *J. of Statistical Software*, vol. 91, pp. 1–30, 2019.

[59] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "Density-based spatial clustering of applications with noise," in *Int. Conf. knowledge discovery and data mining*, vol. 240, no. 6, 1996.

[60] X. Zhang, L. Wang, J. Chen, C. Fang, G. Yang, Y. Wang, L. Yang, Z. Song, L. Liu, X. Zhang *et al.*, "Dual radar: A multi-modal dataset with dual 4d radar for autononous driving," *Scientific data*, vol. 12, no. 1, p. 439, 2025.

[61] C. Zhao, D. Ding, Z. Du, Y. Shi, G. Su, and S. Yu, "Analysis of perception accuracy of roadside millimeter-wave radar for traffic risk assessment and early warning systems," *International journal of environmental research and public health*, vol. 20, no. 1, p. 879, 2023.

[62] O. Ervan and H. Temeltas, "A fast and precise multi-level global registration method for 3d point clouds," Istanbul Technical University, SSRN Working Paper 4705614, 2024, accessed: Oct. 2025. [Online]. Available: https://ssrn.com/abstract=4705614

[63] M. Bouzeid, P. Bruel, V. Poulain, J. Tachella, J.-Y. Tourneret, and D. Youssefi, "A plug-and-play approach for point cloud registration," in *2025 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2025, pp. 21–25.

[64] S. Qiu, S. Anwar, and N. Barnes, "Pnp-3d: A plug-and-play for 3d point clouds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1312–1319, 2021.

[65] J. Tachella, Y. Altmann, N. Mellado, A. McCarthy, R. Tobin, G. S. Buller, J.-Y. Tourneret, and S. McLaughlin, "Real-time 3d reconstruction from single-photon lidar data using plug-and-play point cloud denoisers," *Nature communications*, vol. 10, no. 1, p. 4984, 2019.

[66] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3450–3466, 2021.

[67] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Computer graphics forum*, vol. 32, no. 5. Wiley Online Library, 2013, pp. 113–123.

[68] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm," *Image and vision computing*, vol. 23, no. 3, pp. 299–309, 2005.

[69] Z. J. Yew and G. H. Lee, "Rpm-net: Robust point matching using learned features," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 824–11 833.

[70] V. Bhandari, T. G. Phillips, and P. R. McAree, "Real-time 6-dof pose estimation of known geometries in point cloud data," *Sensors*, vol. 23, no. 6, p. 3085, 2023.

[71] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Transactions on automatic control*, vol. 25, no. 2, pp. 164–176, 1980.

[72] G. W. Stewart, "On the early history of the singular value decomposition," *SIAM review*, vol. 35, no. 4, pp. 551–566, 1993.

[73] C. C. Paige and M. A. Saunders, "Towards a generalized singular value decomposition," *SIAM Journal on Numerical Analysis*, vol. 18, no. 3, pp. 398–405, 1981.

[74] Y. He and C.-H. Lee, "An improved icp registration algorithm by combining pointnet++ and icp algorithm," in *International Conf. on Control, Automation and Robotics*, 2020, pp. 741–745.

[75] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE transactions on image processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[76] D. Harvey, S. Leybourne, and P. Newbold, "Testing the equality of prediction mean squared errors," *International Journal of forecasting*, vol. 13, no. 2, pp. 281–291, 1997.

[77] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[78] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.

[79] Y. Ren, W. Xu, Y. Guo, Y. Liu, Z. Tian, J. Lv, Z. Guo, and K. Guo, "Mlf-pointnet++: A multifeature-assisted and multilayer fused neural network for lidar-uas point cloud classification in estuarine areas," *Remote Sensing*, vol. 16, no. 17, p. 3131, 2024.

[80] K. T. Wijaya, D.-H. Paek, and S.-H. Kong, "Advanced feature learning on point clouds using multi-resolution features and learnable pooling," *Remote Sensing*, vol. 16, no. 11, p. 1835, 2024.

[81] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[82] R. Bro and A. K. Smilde, "Principal component analysis," *Analytical methods*, vol. 6, no. 9, pp. 2812–2831, 2014.

[83] R. Yacouby and D. Axman, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models," in *Proceedings of the first workshop on evaluation and comparison of NLP systems*, 2020, pp. 79–91.

[84] M. Buckland and F. Gey, "The relationship between recall and precision," *Journal of the American society for information science*, vol. 45, no. 1, pp. 12–19, 1994.

[85] F. Fressin, G. Torres, D. Charbonneau, S. T. Bryson, J. Christiansen, C. D. Dressing, J. M. Jenkins, L. M. Walkowicz, and N. M. Batalha, "The false positive rate of kepler and the occurrence of planets," *The Astrophysical Journal*, vol. 766, no. 2, p. 81, 2013.

[86] L. M. Kucirka, S. A. Lauer, O. Laeyendecker, D. Boon, and J. Lessler, "Variation in false-negative rate of reverse transcriptase polymerase chain reaction–based sars-cov-2 tests by time since exposure," *Annals of internal medicine*, vol. 173, no. 4, pp. 262–267, 2020.

[87] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of thoracic oncology*, vol. 5, no. 9, pp. 1315–1316, 2010.

[88] F. S. Nahm, "Receiver operating characteristic curve: overview and practical use for clinicians," *Korean journal of anesthesiology*, vol. 75, no. 1, pp. 25–36, 2022.

[89] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[90] Y. Hadad, I. Stainvas, and I. Bilik, "Adversarial attacks on modified point-net for radar point cloud classification," in *Proceedings of the IEEE Radar Conference*, 2025.