mmOVD$^+$: Detecting Fully Occluded Vehicles With Millimeter-Wave Signals

Chenming He, Chengzhen Meng, Xiaoran Fan, Dequan Wang, Lu Zhang, *Member, IEEE,* Yubo Yan, *Member, IEEE,* and Yanyong Zhang, *Fellow, IEEE*

***Abstract*—Reliable detection of nearby vehicles is critical for autonomous driving. Problems thus arise when a vehicle is occluded and becomes undetectable, which may lead to accidents. In this paper, we develop *mmOVD*$^+$, a millimeter-wave radar-based system that enables fully occluded vehicle detection by leveraging ground-reflected signals that propagate beneath the blocking vehicle's chassis. To eliminate ghost points caused by frequent multi-path reflections, we devise a set of neighborhood behavior-based point features and integrate them into *GhostNet*, a customized deep network that enables accurate point classification through combined feature learning. In addition, we propose a cumulative clustering algorithm to effectively aggregate the unstable ground-reflected radar points over consecutive frames to derive the bounding boxes of the vehicles.**

We have evaluated *mmOVD*$^+$ in both controlled and real-world environments. In an underground garage and two campus roads, we conducted controlled experiments in 70 scenes with 9 vehicles, including a minibus, a motorcycle, and an electric moped. Our system accurately detects occluded vehicles, with a 92.7% F1 score for occluded vehicle detection and a 100% success rate for occlusion event detection. More importantly, we drove 324 km on crowded roads at a speed up to 70 km/h and show we could achieve a 93.8% occlusion detection success rate and a 2.4% false alarm rate with only 10% of the training data.

***Index Terms*—Millimeter-wave Radar Sensing, Vehicle Detection, Autonomous Driving.**

I. Introduction

DETECTING and continuously tracking nearby vehicles is a fundamental requirement for autonomous and assisted driving systems. However, this process can be severely disrupted by frequent occlusions between vehicles, where the target vehicle becomes invisible to onboard sensors [1], [2]. A common example is when multiple vehicles travel side by side, forming persistent blind zones in the sensing range of typical vehicle-mounted sensors such as cameras and LiDARs, as illustrated in Fig. 1. Such blind zones pose significant safety risks, especially during maneuvers like overtaking, lane changes, or turns, as they hinder both accurate target tracking and timely path planning. According to the U.S. National Highway Traffic Safety Administration (NHTSA), blind zones contributed to over 1,542 fatal crashes in 2022 [3]. These statistics highlight the urgent need for reliable vehicle detection in occlusion scenarios to ensure driving safety.

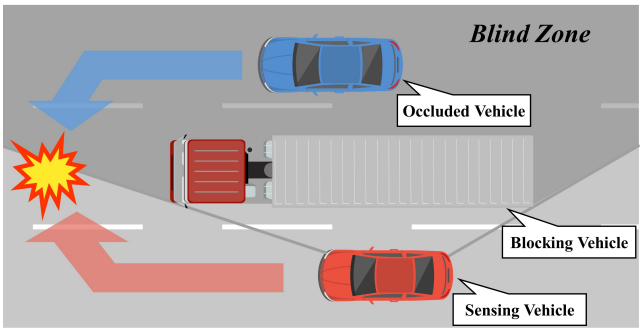


Fig. 1: Illustration of the blind zone caused by the occlusion between vehicles.

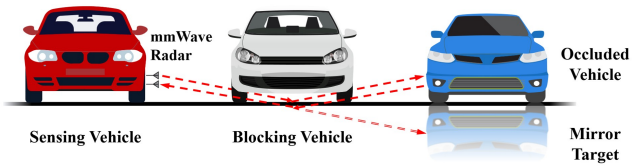


Fig. 2: Occluded vehicle detection through ground-reflected mmWave signals. The red line illustrates the propagation of signals. The reflection of the ground generates a mirror target, from which we can detect the occluded target.

Several prior studies have attempted to address this challenge by leveraging vehicle-mounted cameras or LiDARs. Some methods segment occluded regions of interest and apply deep learning or geometric reasoning to infer the presence of hidden vehicles [2], while others rely on distinctive vehicle features—such as windscreens—to support occlusion detection [4]. However, these methods are typically limited to *partially* occluded scenarios, where a substantial portion of the vehicle remains visible. To date, no existing system can robustly detect *fully* or *mostly* occluded vehicles.

To bridge this gap, we propose to utilize a less explored sensor, millimeter-wave (mmWave) radar [5] for occlusion detection. Compared to traditional sensors, mmWave radars offer distinct advantages, including high ranging accuracy, small form factor, and low power consumption. These benefits have made mmWave radars increasingly popular for localization and perception tasks [6]–[12]. Many public datasets for autonomous driving (e.g., nuScences [13]) have also adopted mmWave radars as one of the supplementary sensors.

A key property of mmWave radar is its sensitivity to multi-

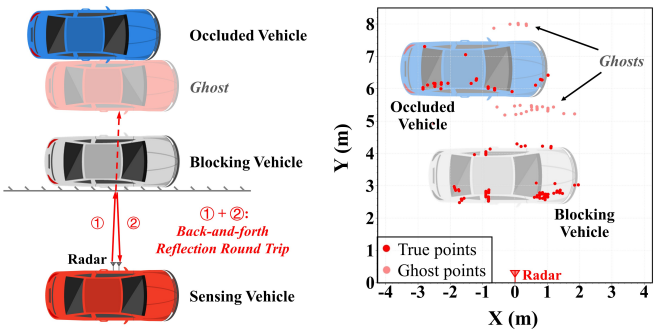
This work was supported by the National Natural Science Foundation of China (No. 62332016) and the Key Research Program of Frontier Science, Chinese Academy of Sciences (No. ZDBS-LY-JSC001). *(Corresponding author: Yanyong Zhang)*

Chenming He, Chengzhen Meng, Dequan Wang, and Yubo Yan are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China (email: hechenming@mail.ustc.edu.cn; czmeng@mail.ustc.edu.cn; wdq15588@mail.ustc.edu.cn; yuboyan@ustc.edu.cn)

Xiaoran Fan is an independent researcher based in Sunnyvale, California, USA (email: gunanjiluzhe@gmail.com)

Lu Zhang is with Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China (email: luzha@ustc.edu.cn)

Yanyong Zhang is with School of Artificial Intelligence and Data Science, University of Science and Technology of China, Hefei 230026, China (email: yanyongz@ustc.edu.cn)



(a) Illustration of ghost phenomenon (b) A frame of radar point cloud

Fig. 3: Ghosts appear in similar locations as occluded vehicles. (a) Signals bounce between the vehicles, creating ghosts. (b) Ghost points appear around true points from the occluded vehicle, making them hard to discriminate.

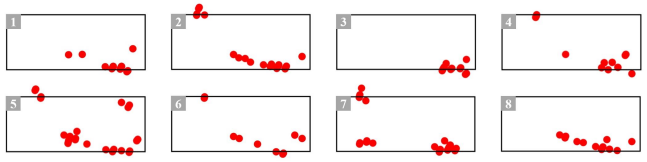


Fig. 4: Radar points from an occluded vehicle in 8 consecutive frames (30 FPS), in which the black boxes represent the vehicle's bounding boxes. On one hand, there are always a cluster of points in similar locations corresponding to the vehicle; on the other hand, the actual sizes and locations of the cluster vary from frame to frame.

path propagation: radio waves can reflect off smooth surfaces such as the ground or nearby vehicles, creating indirect signal paths. Such multi-path effects have been extensively studied in other wireless domains—such as WiFi [14]–[16] and RFID [17]–[19]—where it is commonly regarded as a source of interference. In contrast, radar-based sensing [20]–[22] can exploit these indirect reflections as useful signals.

In our case, the multi-path effect enables non-line-of-sight detection. For instance, radar signals can reflect off the ground [23] beneath a blocking vehicle, generating a *mirror target* that reveals the occluded vehicle (Fig. 2). This capability allows mmWave radars to sense objects within the occlusion zone that are invisible to conventional line-of-sight sensors. However, robust detection remains challenging, as the same multi-path effect also introduces two key issues:

False Alarm Due to Ghost Points: Vehicles are effective reflectors of mmWave signals as they are primarily made of metal and reinforced plastics. When vehicles drive closely together, millimeter waves bounce back and forth between them (Fig. 3(a)). The signals after one back-and-forth reflection generate the true points, while those after two or more reflections create "ghost points", which might trigger false alarms. For instance, several ghost clusters appear (Fig. 3(b)) due to multiple reflections. Ghost points typically exhibit similar signal intensity to true points and appear in locations where occluded vehicles may be present, making the elimination of ghost points an essential challenge.

Inaccurate Detection Due to Unstable Radar Points: In occlusion scenarios, the mmWave radar indirectly detects occluded vehicles through ground reflections, causing unstable points due to changing signal propagation paths as vehicles move. Fig. 4 illustrates the radar points from an occluded vehicle over eight consecutive frames (at 30 FPS), where the black boxes represent the vehicle's bounding boxes. The radar points are rather unstable, appearing at different positions on the vehicle's body, and sometimes even forming several separate point clusters. Consequently, applying typical radar point clustering algorithms (e.g., DBSCAN [24]) can lead to significant positioning errors and even erroneously divide a single vehicle into multiple targets.

This paper presents *mmOVD*$^+$, an occluded vehicle detection system based on mmWave radar. Recognizing the critical impact of sensor setup, *mmOVD*$^+$ first incorporates an optimized radar deployment strategy, meticulously determining the ideal placement of radars on the sensing vehicle to maximize the quality of ground-reflected signals from occluded vehicles while simultaneously minimizing interference from ghost points. Following this foundational setup, raw radar point clouds are generated using the received mmWave signals. However, the raw point clouds contain a large amount of noise, including the challenging ghost points.

To eliminate the ghost points, *mmOVD*$^+$ introduces a refined ghost removal module. Building upon the observation that true and ghost points exhibit drastically different spatial and velocity distributions in their neighborhood, we carefully devise a set of *neighborhood behavior*-based point features. These features extend beyond an individual point's properties (like signal strength or Doppler velocity) to capture the distinct neighbor density, the unique halfway point density, and the nuanced neighbor velocity distribution. Furthermore, to enhance ghost removal performance, especially with sparse mmWave radar points, we integrate *GhostNet*, a novel deep learning network. This architecture, specifically customized from PointNet++, synergistically combines our handcrafted features with learned implicit representations, allowing for accurate point classification even with limited training data.

Further, to combat the challenge caused by the relatively unstable radar points from occluded vehicles, we propose a *conservative clustering, cumulative aggregation* approach. Specifically, we first perform conservative clustering to obtain point clusters that are small and possibly incomplete. We then accumulate these raw clusters from multiple frames and aggregate them into more complete clusters. In this way, we can accurately extract the bounding box of each vehicle and detect whether there is occlusion based on the spatial relationship of these boxes.

In summary, our work makes the following contributions:

- We design and implement an occluded vehicle detection system, *mmOVD*$^+$, which leverages mmWave radars to detect vehicles even in fully occlusion scenarios. To the best of our knowledge, this is the first system that can accurately detect fully occluded vehicles (including motorcycles and electric mopeds).
- We propose a set of neighborhood spatial- and velocity distribution-based point features to discriminate ghost points, and develop *GhostNet*, a novel network that combines these features with deep representations. We have also designed a new detection algorithm that can effectively cope with unstable radar signals reflected from the occluded car by adopting a conservative clustering, cumulative aggregation approach.

- We conduct extensive evaluations in both controlled and real-world environments. Our testbed includes multiple occlusion scenarios and 9 different vehicles, including a minibus, a motorcycle, and an electric moped. The results show that *mmOVD*$^+$ achieves an average accuracy of 95.3% for point classification, an F1 score of 92.7% for occluded vehicle detection, and a success rate of 100% for occlusion event detection with a false alarm rate lower than 5% in our dataset. Additionally, we drove 324 km on crowded roads at a speed up to 70 km/h. The results demonstrate that *mmOVD*$^+$ can achieve an occlusion detection success rate of 93.7% and a low false alarm rate of 2.4% with only 10% training data, demonstrating strong generalization and robustness.

The rest of the paper is organized as follows. Section II discusses the related work. Section III elaborates on our design. The evaluation of our system with controlled experiments and real-world experiments are presented in Sections IV and V, respectively. We conclude this work in Section VII.

II. Related Work

A. Partial Occlusion Detection

Detecting vehicles under partial occlusion has been extensively studied in autonomous driving research [1], [2], [25], [26]. Most existing methods rely on vehicle-mounted sensors such as cameras and LiDARs, and propose various occlusion handling techniques to improve perception performance. For example, Pham and Lee [4] leveraged the distinctive appearance of vehicle windshields to identify partially occluded vehicles. Ohn-Bar et al. [27] combined monocular and stereo vision with motion and depth cues to enhance detection performance. These approaches typically apply appearance modeling, region segmentation, and part-based reasoning to infer the presence of occluded objects. However, their effectiveness is limited to partially-occluded scenarios, where a significant portion of the target remains visible. Detection performance in partially occluded cases typically exceeds that in heavily occluded ones by 10%–40% [28]. This performance gap highlights the challenge of detecting vehicles under heavy or full occlusion, where traditional vision-based methods often fail due to the lack of direct line-of-sight observations.

To address this challenge, our system *mmOVD*$^+$ takes a fundamentally different approach by leveraging mmWave radar's multipath reflections from the ground to detect vehicles even under full occlusion. By utilizing these indirect signals, our system extends detection capability beyond the visible field of view, enabling robust detection in scenarios where vision-based methods fail entirely.

B. Radar-based NLOS Detection

In the field of wireless sensing, many researchers have focused on achieving non-line-of-sight (NLOS) detection based on mmWave radar, such as detecting objects behind corners [29]–[32]. For example, Guo et al. [33] proposed a NLOS location algorithm based on ray-tracing model using a mmWave radar. Yue et al. [34] addressed the limitations of prior RF-based ray-tracing models by proposing CornerRadar, which combines radar signals with a novel reflection encoding method and leverages deep learning to achieve accurate around-corner localization. Additionally, Scheiner et al. [35] leveraged Doppler radar and temporal signal analysis to detect and track hidden objects in outdoor environments. While these methods demonstrate the feasibility of radar-based NLOS perception, they primarily focus on around-the-corner detection in static or low-speed scenes. Furthermore, these methods typically require other sensors (e.g., LiDARs) to provide accurate modeling of reflective surfaces, which increases the difficulty of deploying them in real-world environments.

In contrast, *mmOVD*$^+$ is designed for the more dynamic and challenging context of autonomous driving. By leveraging ground reflections rather than relying on detailed environmental models, our system eliminates the need for auxiliary sensors and complex scene reconstruction. Furthermore, we address key challenges such as unstable reflection paths and the presence of dense ghost points, enabling robust detection of fully occluded vehicles in complex, real-world scenarios.

C. Ghost Point Identification

Identifying ghost points remains a critical challenge in mmWave radar sensing. Some researchers adopt feature engineering [36] or rule-based approaches [37], [38] to detect and eliminate ghost points. For instance, Wang et al. [38] proposed a ghost detection algorithm based on velocity change patterns, achieving accurate ghost removal in indoor environments. These methods are highly interpretable and require only a small amount of labeled data for reliable operation. However, they often struggle to capture the full spatial-temporal information of ghost points, which limits their effectiveness in complex or dynamic scenarios. Other researchers have explored deep learning-based approaches [39]–[41], which learn point features in a data-driven manner to effectively suppress ghost points. Zeller et al. [41] developed a transformer-based framework that enriches the information of individual points to accurately remove both ghost and background points. While these methods typically achieve strong performance on benchmark datasets, they are computationally intensive and require large amounts of labeled training data.

To overcome these limitations, we propose a hybrid strategy that combines handcrafted spatial-temporal features—such as neighbor density, halfway point density, and neighbor velocity distributions—with a customized deep learning architecture, *GhostNet*. This design enables accurate ghost point removal with enhanced interpretability and significantly reduced dependency on large-scale labeled data.

III. System Design

A. Overview

Fig. 5 depicts the overview of our *mmOVD*$^+$ system, which consists of the following components:

(1) **Radar Deployment Optimization:** Before data processing, we investigate the optimal placement of radars on the sensing vehicle to reduce the signal intensity of ghost points while enhancing the reflected signals from occluded vehicles. Through comprehensive empirical studies, we determine an optimized radar deployment strategy.

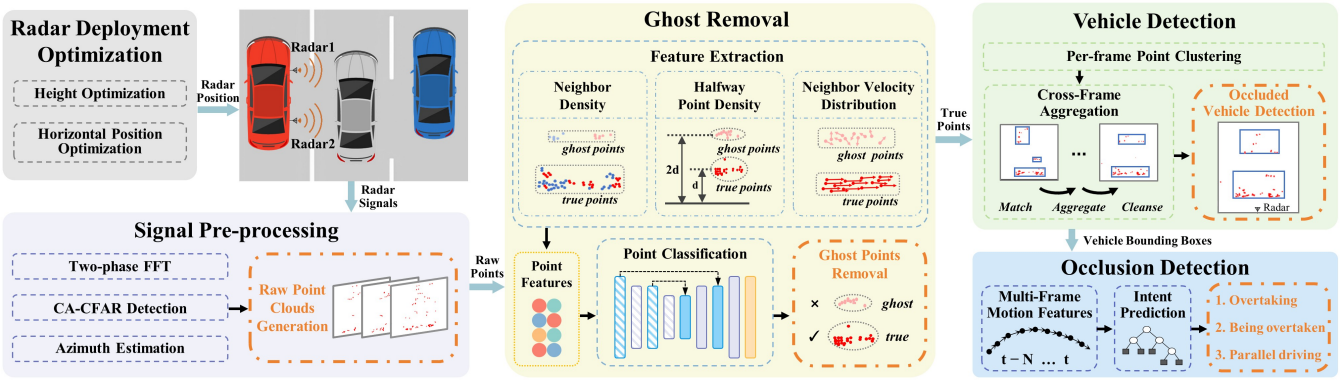


Fig. 5: *mmOVD*$^+$ involves five main steps: (1) radar deployment optimization, (2) signal pre-processing, (3) ghost removal, (4) vehicle detection, and (5) occlusion detection and intent prediction.

(2) **Radar Signal Pre-processing:** The received mmWave signals first go through the range measurement and velocity measurement processes. We then use a constant false alarm rate (CFAR) algorithm to generate point clouds. These points consist of true points (i.e., those from reflections from actual targets) and noise points (i.e., those include ghost points and background noises).

(3) **Neighborhood Spatial/Velocity-Based Ghost Removal:** In the raw point clouds, ghost points are the dominant noises and the most challenging to eliminate. By thoroughly examining the characteristics of radar points, we propose neighborhood spatial features and velocity distribution features to effectively eliminate ghost points.

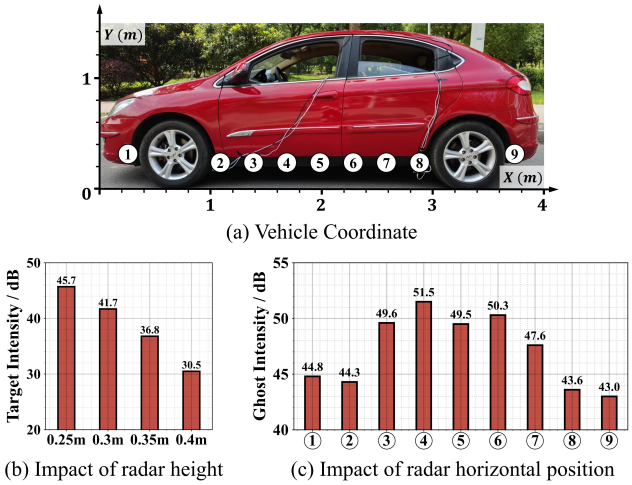
(4) **Cross-Frame Cumulative Aggregation-Based Vehicle Detection:** After removing ghost points, we need to detect vehicles from the point clouds. We first perform conservative clustering to get raw clusters of vehicles. We then propose a cumulative aggregation approach, which accumulate the incomplete clusters from multiple frames to achieve accurate detection for occluded vehicles.

(5) **Occlusion Detection and Intent Prediction:** Based on the detected vehicle bounding boxes, we further identify occlusion events by examining spatial relationships across consecutive frames. Once an occlusion is confirmed, we leverage short-term trajectory sequences to infer the motion intent of occluded vehicles, enabling higher-level understanding of their behaviors.

B. Radar Deployment Optimization

In this section, we explore the optimal deployment positions of the radars through experiments. Our two main goals are: (i) to enhance the signal reflections from occluded vehicles, and (ii) to suppress the signal intensity of ghost points. To facilitate quantitative analysis, we define a vehicle coordinate system, using the projection point of the vehicle head on the ground as the origin, the horizontal direction as the x-axis, and the vertical direction as the y-axis (see Fig. 6(a)).

We first examine the impact of radar height, i.e., the y-axis position of radars. Theoretically, the radars should be placed as low as possible so that more waves can propagate under the blocking vehicle, allowing for clearer detection of occluded vehicles. We test radar heights within the range of $y = [0.25, 0.4]$ m, with a sampling interval of 0.05 m. Fig. 6(b)



(a) Vehicle Coordinate

(b) Impact of radar height

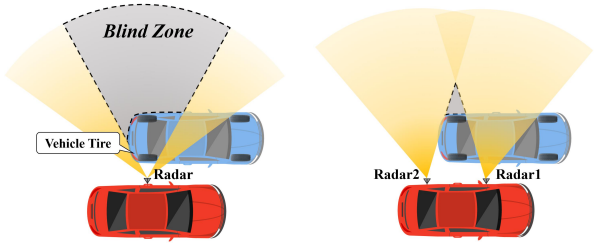
(c) Impact of radar horizontal position

Fig. 6: (a) Vehicle coordinate system. (b) Average signal intensity of true points from the occluded vehicle with different radar heights. (c) Average signal intensity of ghost points with different radar horizontal positions.

shows the result: as the radar height increases, the signal intensity from occluded vehicles drops significantly, validating our hypothesis. Therefore, we choose a height of 0.25 m—the chassis height of our vehicle and the lowest position that does not interfere with driving.

Next, we examine the horizontal position along the x-axis by testing 9 different placements of the radar, as illustrated by the circled numbers ①∼⑨ in Fig. 6(a). The corresponding results are shown in Fig. 6(c). It can be seen that when the radar is placed next to the tires, the intensities of the ghost points are the weakest. The reason is that the tires are less reflective than flat doors, leading to fewer, weaker, and sparser ghost points. Based on this observation, we have four candidate positions ①, ②, ⑧ and ⑨—near the front and rear tires on each side of the vehicle—for radar deployment.

Our final consideration is whether to use a single radar or multiple radars on each side of the sensing vehicle. In practice, we find that when relying on a single radar for detection, the propagation of radar signals may be significantly obstructed by the tires of the blocking vehicle, resulting in a substantial blind zone. An example is illustrated in Fig. 7(a). By installing two (or more) radars on each side of the sensing vehicle, their fields of view can complement each other, as shown in Fig. 7(b). This configuration not only reduces blind zones but also enables cross-validation between overlapping sensing



(a) Single radar (b) Two radars

Fig. 7: (a) A single radar may lead to a large blind zone since the mmWave signals can be significantly obstructed by the tires of the blocking vehicle. (b) Having two or more radars can greatly alleviate the blind zone problem due to their complementary coverage.

regions, as will be discussed in Sec. III-D1. Our experiments indicate that installing two radars on each side of the vehicle and merging their data can significantly increase detection accuracy by over 20% compared to using only one radar. However, using three radars does not produce a significant gain, with detection accuracy only improving by 1.2% over using two radars. Therefore, in the design of *mmOVD*$^+$, we assume two radars are mounted on each side of the vehicle. Specifically, to maximize coverage and ensure overlapping fields of view, we place the radars at positions ② and ⑧, with each radar mounted at a height of 0.25 m.

C. Radar Signal Pre-processing

Once the radar signals are received, we perform signal pre-processing to extract radar point clouds. This involves a standard two-dimensional Fast Fourier Transform (2D-FFT) pipeline to obtain range and velocity information. Specifically, we first apply a 1D FFT along the time domain within each chirp to obtain the range spectrum. This step resolves the time delay of reflected signals, thereby measuring the distance between the radar and the reflecting objects.

Subsequently, we perform a second FFT across multiple chirps to estimate the Doppler frequency shift, which is used to calculate the relative radial velocity of each object with respect to the radar. The combination of these two steps yields a Range-Doppler Matrix (RDM), where each element represents the reflected signal intensity corresponding to a particular range and Doppler velocity bin.

To identify potential targets, we apply a Cell Averaging Constant False Alarm Rate (CA-CFAR) detector on the RDM. This algorithm adaptively selects a detection threshold based on the noise level in neighboring cells, allowing for robust peak detection under varying noise conditions. Detected peaks in the RDM are then considered candidate targets.

For each detected peak, we estimate the azimuth angle using a third FFT applied across the antenna array (angle FFT), which leverages the phase differences among the receiving antennas to resolve the angular position of the target.

The final output of this pre-processing pipeline is a radar point cloud $\mathcal{P}$, where each point p represents a detected reflection characterized by five physical parameters:

$$\mathcal{P} = \{p\},\ p = (r, v_d, \theta, SI, SNR), \tag{1}$$

where r denotes the range, v_d the Doppler velocity, θ the azimuth angle, SI the signal intensity, and SNR the signal-to-noise ratio.

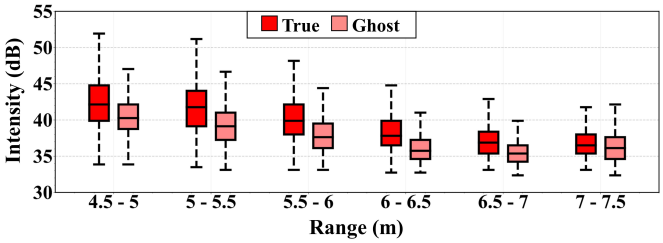


Fig. 8: Intensity levels of true and ghost points are hard to be differentiated at all range intervals.

D. Ghost Removal through Neighborhood Spatial/Velocity Feature Extraction

After the above pre-processing steps, we obtain the raw point clouds which consist of true points, ghost points, and background noise points. The background noise points, sparse and random, can be easily removed through the density-based spatial clustering of applications with noise algorithm (DBSCAN) [24], which is adopted in our system (detailed in Section III-E). However, distinguishing between true and ghost points poses a dire challenge to the underlying design, as ghost points often exhibit similar signal intensity as the true points. Our experiments indicate a significant overlap in signal intensity between ghost points and true points at similar locations, as shown in Fig. 8. Moreover, ghost points often appear in locations where occluded vehicles are located (illustrated in Fig. 3), further complicating this task. In this section, we provide a detailed explanation of our ghost removal approach, which aims to accurately discriminate ghost points from true points. To clearly explain our design, we only focus on the sensing vehicle (with two radars on each side) and the blocking vehicle below.

To distinguish true points from ghost points, two distinct approaches are usually available. One approach is to use deep-learning networks to perform point classification [39]. The other approach is feature engineering, which involves extracting handcrafted features for each radar point and using these features to classify the points [36]. Considering the amount of data and computational resources required by the learning-based approaches, we adopt the second approach by thoroughly investigating the characteristics of radar points and deriving a suitable feature set for classification purposes.

1) Neighbor Density Features: We first examine the point density within a true/ghost point's local neighborhood. Our first observation is that true points tend to have more neighboring points than ghost points, as shown by the CDF results in Fig. 9(a). This is because the mmWave signal energy fades quickly with a longer reflection path, resulting in many fewer ghost points which are also much more scattered.

We define our test point as $p_{test}^{r,f}$ with r denoting the radar ID that generates the point and f the frame ID. Then we define the *neighbor density* feature F_{nbr} for $p_{test}^{r,f}$ as the number of neighboring points within the distance threshold:

$$F_{nbr}^{r,f} = \left|\{p \in \mathcal{P} \mid dist(p, p_{test}^{r,f}) < T_{nbr}\}\right|, \tag{2}$$

where $\mathcal{P}$ is the radar point cloud, and T_{nbr} is the neighbor distance threshold. Next, we enhance this feature by incorporating additional temporal and spatial dimensions, generating

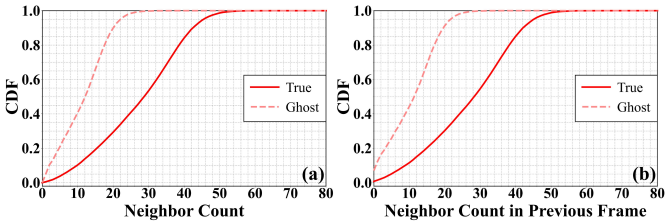


Fig. 9: (a) CDF of a point's neighbor counts within a radius of 0.8m; (b) CDF of a point's neighbor counts within a radius of 0.8m in the previous frame.

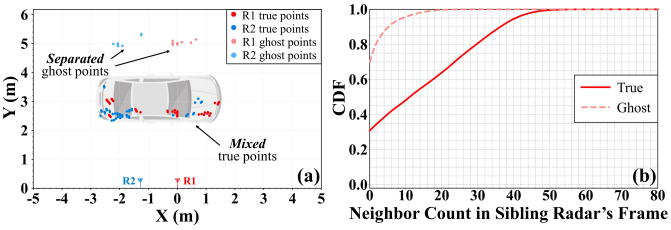


Fig. 10: (a) Aggregated point clouds from the two sibling radars, in which the true point clusters are commonly mixed, while the ghost point clusters are separated. (b) CDF of a point's neighbor count in the sibling radar's corresponding frames.

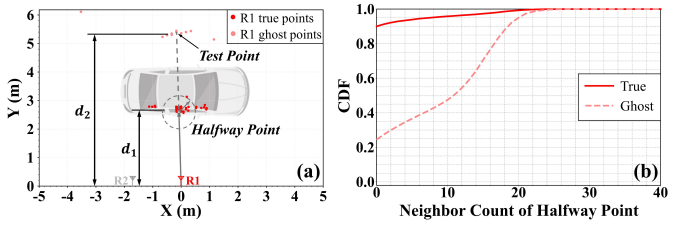


Fig. 11: (a) A cluster of points commonly appears halfway between the radar and ghost points ($d_2 \approx 2d_1$). (b) CDF of halfway point's neighbor counts within a radius of 0.8m.

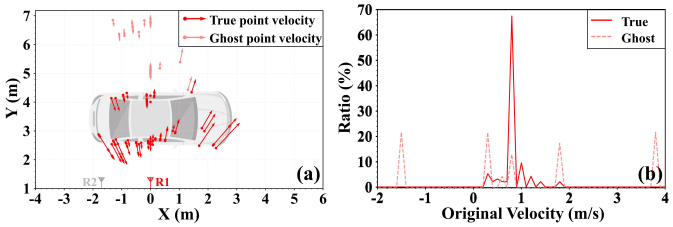


Fig. 12: (a) Doppler velocities of radar points, with arrows representing direction and magnitude. (b) Original velocity distribution of the points from a real vehicle (red solid line) and a ghost (pink dashed line).

a suite of neighbor density features. Its complete neighbor density features include:

- The neighbor density feature $F_{nbr}^{r,f}$. The first feature is the test point's neighbor density feature value in the current frame f from the same radar r;
- The history neighbor density feature $F_{nbr}^{r,f-1}$. The second feature calculates the neighbor density around the test point's location in the previous frame (with frame ID $f-1$). The rationale here is that a real vehicle appears in similar positions in two consecutive frames which generate dense points in both frames. Thus, if a test point has dense neighbors in consecutive frames, its likelihood of being a true point increases. Fig. 9(b) shows our experimental evidence.
- The sibling sensor's neighbor density feature $F_{nbr}^{\widetilde{r},f}$. The third feature calculates the neighbor density around the test point's location in frame f from the sibling radar (we assume there are two radars on each side of the sensing car) with radar ID $\widetilde{r}$. The rationale here is that if a test point has dense neighbors in both radars' frames, its likelihood of being a true point increases. As shown in Fig. 10(a), if we merge the frames from both radars (with proper spatial calibration), we find that the true points from the two radars tend to cluster together, while their ghost points are further apart. As a result, the difference between the neighbor densities in the sibling radar's frames is even more pronounced, as shown in Fig. 10(b).

2) Halfway Point Density Feature: We next examine the point distributions in areas beyond the ghost point's local neighborhood and find that there is usually a cluster of points halfway between the radar and the ghost point. That is, a ghost point's halfway point density is higher than a true point's halfway point density. We illustrate this pattern in Fig. 11(a). In our case, the halfway location between the radar and the ghost point coincides with the blocking car's near-radar side. Besides, the CDF results depicted in Fig. 11(b) also support this observation.Below we explain this observation.

In our setting, the ghost points result from signal reflections between the sensing car and the blocking car (as shown in Fig. 3). When the radar receives the reflected signal after a single round trip between the radar and the blocking car's near radar side (the reflecting surface), it generates the true points. When the radar receives reflected signals after two or more round trips, it generates the ghost points. When we focus on those ghost points from two round-trip reflections, we find their signal propagation distance is twice of the aforementioned true points. We can find a cluster of true points around halfway between the radar and the ghost points, hence the halfway point density feature. Here, we ignore those ghost points generated after more than two round trips because their propagation paths are too complex to analyze.

We define the halfway point density feature F_{half} as:

$$F_{half} = \left|\{p \in \mathcal{P} \mid dist(p, p_{half} < T_{nbr}\}\right|, \tag{3}$$

where the p_{half} represents the halfway point between the test point and the radar.

3) Neighbor Velocity Distribution Feature: After exploring the point density distribution as the discriminating factor, we next explore the dynamic aspect of points: velocity. To be more precise, we leverage the Doppler velocity of radar points [21] and differentiate between true points and ghost points by exploring differences in their velocity variation patterns. The mmWave radar detects an object's Doppler velocity, which represents the radial velocity (as shown in Fig. 12(a)), by calculating the phase differences $\Delta\Phi$ of adjacent chirps in the reflected signals [42]. Given the wavelength λ of mmWave signals and the time interval T_c between chirps, the Doppler velocity v_d can be calculated as: $v_d = \frac{4\pi T_c}{\lambda \Delta\Phi}$. Notably, Doppler velocities of radar points can't be directly used for point discrimination. Instead, in the case of vehicles driving side by side, we can accurately reconstruct the original velocity based on the Doppler velocity v_d and azimuth angle θ of the radar points: $v = v_d / sin\theta$.

Among the points generated by the radar, the true points reflected by the same car exhibit the same velocity, i.e., the

reflecting car's velocity. Conversely, ghost points, affected by accumulated phase errors from multiple reflections [43], are anticipated to exhibit different velocities. To illustrate, Fig. 12(b) shows an example of the original velocity distributions of true points and ghost points, in which the x axis represents the velocity, and the y axis represents the ratio of radar points moving at different velocities. We observe that true points predominantly exhibit the same velocity, forming a main peak, while the ghost velocities are more chaotic and scattered, even with different directions.

We thus propose a feature that characterizes the test point's neighbors' velocity distribution, denoted as F_{vel}. We consider a velocity range of $(-10, 10)$ m/s, with an interval of 0.1m/s. The neighbor velocity distribution vector $\boldsymbol{R_{vel}} = (r_1, ..., r_M)$ thus records the ratio of neighboring points whose velocity falls within the M ($M = 200$ in our implementation) velocity intervals. Notably, this velocity range refers to the range of velocity differences between the sensing vehicle and neighboring vehicles. We believe such a velocity difference range is reasonable for real-world driving conditions, and our experiments in Sec. V have confirmed this.

In this raw form, such velocity distribution vectors are usually sparse—on average, 43.8% of the vector elements are zeros. We then employ principal component analysis (PCA) to record the velocity ratios in a reduced 3-dimensional vector. The resulting feature $\boldsymbol{F_{vel}}$ is calculated as:

$$\boldsymbol{F_{vel}} = \boldsymbol{R_{vel}}\boldsymbol{W}, \quad \boldsymbol{W} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}), \tag{4}$$

where $\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}$ represent the top three principal components, each being a column vector of length M.

4) mmOVD$^+$ *Point Classification:* We summarize our proposed radar point classification feature set in Table I. Within the feature set, the test point is denoted as $p_{test}^{r,f}$. We set the neighbor distance threshold, denoted as T_{nbr}, to 0.8m. This value is chosen as vehicles typically maintain a lateral gap of over 1m for safety. Our experiments demonstrate that over 95% of points whose distances from each other are smaller than this threshold belong to the same vehicle or ghost.

Next, we feed the raw radar points along with their features into a deep learning network, which further extracts implicit representations and performs point-wise classification. In this way, our approach combines both handcrafted features and deep learning techniques, enabling more accurate classification with less training data.

TABLE I: Our handcrafted point feature set

Feature	***Description***
r, v_d, θ, SI, SNR	Basic features: range, azimuth, doppler velocity, signal intensity, and signal-to-noise ratio of $p_{test}^{r,f}$
$F_{nbr}^{r,f}$	Number of points within the distance threshold T_{nbr} to $p_{test}^{r,f}$
$F_{nbr}^{r,f-1}$	Number of points within the distance threshold T_{nbr} to $p_{test}^{r,f}$ in the previous frame
$F_{nbr}^{\widetilde{r},f}$	Number of points within the distance threshold T_{nbr} to $p_{test}^{r,f}$ in the frame of the sibling radar
F_{half}	Number of points within the distance threshold T_{nbr} to the halfway point
F_{vel}	Reduced 3-dimensional velocity distribution of the points within the distance threshold T_{nbr} to $p_{test}^{r,f}$

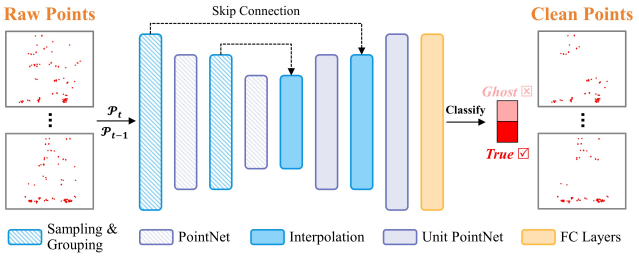


Fig. 13: Architecture of our *GhostNet*, which extracts multi-scale features from raw radar points and classifies each point as ghost or true.

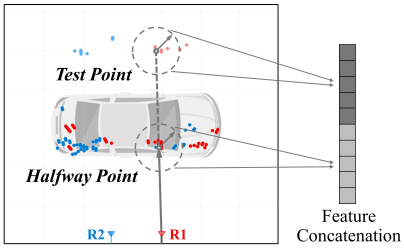


Fig. 14: Illustration of the halfway grouping strategy. Besides the standard grouping around the test point, we additionally gather neighboring points around the halfway point to enhance spatial feature representation.

The architecture of our network, *GhostNet*, is illustrated in Fig. 13. It is built upon PointNet++ [44], a classical and efficient model for point cloud processing. PointNet++ adopts a hierarchical architecture that progressively downsamples the input point cloud to extract local features and then upsamples to propagate global context back to individual points. Each downsampling layer consists of three steps: (1) sampling, which selects a subset of representative points; (2) grouping, which defines local neighborhoods for each sampled point; and (3) feature extraction, which aggregates features within each local region using a PointNet [45]. During upsampling, the model interpolates the coarse-level features back to the original points, followed by another unit PointNet to refine the representations. The final per-point features obtained through this hierarchical process are then fed into fully connected layers for point-wise classification.

However, the original PointNet++ is designed for LiDAR point clouds, which differ significantly from mmWave radar point clouds in terms of sparsity, noise characteristics, and distribution. To address these differences, we design *GhostNet* as a customized network tailored specifically for mmWave radar point classification. Its architecture is guided by the insights gained from our handcrafted feature design and incorporates several targeted modifications:

- First, the historical neighbor density feature $F_{nbr}^{r,f-1}$ demonstrates that incorporating the previous frame of points can benefit classification. Therefore, we input both the current frame $\mathcal{P}_t$ and the previous frame $\mathcal{P}_{t-1}$ into the network. Notably, only the points from $\mathcal{P}_t$ are used for sampling and classification, while the points from $\mathcal{P}_{t-1}$ are utilized solely for feature extraction.
- Second, the neighbor density features indicate that nearby points provide crucial context for accurate point classification. Inspired by this, we design the encoder of *GhostNet* with multiple grouping layers that search for neighboring points within a specified radius and aggre-

gate them into local region features using PointNet layers. While this follows the general structure of PointNet++, we tailor the radius settings to the sparse nature of radar point clouds. Instead of the typical 0.1 m initial radius used for dense indoor LiDAR data, we set the first radius to 0.8 m—matching the neighbor threshold T_{nbr}—and double it in subsequent layers. This customized radius scaling better captures meaningful local structures in sparse radar data, while maintaining conceptual consistency with the handcrafted features.

- Third, motivated by the halfway point density feature F_{half}, we introduce a halfway grouping step into the grouping layers, as illustrated in Fig. 14. Specifically, in addition to the standard circular grouping around the test point, we also group neighboring points within the same radius centered at the halfway point. Features from both groups are extracted using PointNet layers and then concatenated to form the feature vector. To the best of our knowledge, this halfway grouping approach has not been explored previously, and is specifically introduced to capture the spatial reflection patterns of ghost points.

E. Occluded Vehicle Detection Through Aggregation Across Frames

After identifying the ghost points and removing them, we next cluster the clean points and estimate the bounding boxes of the detected vehicles. Then from the spatial relationship between the bounding boxes, we infer their relative positions and detect the occluded cars.

A typical approach is to first apply clustering algorithms (e.g., DBSCAN [24]) on each frame to obtain the cars' bounding boxes, and then perform object tracking to associate the bounding boxes to refine their locations/sizes. However, we cannot directly apply this approach in our case because the radar points reflected from occluded vehicles are unstable (illustrated in Fig. 4), rendering the per-frame clusters less reliable than desired. To make matters worse, these clusters also contain ghost points that are hard to remove.

To combat the challenge caused by the unstable nature of the reflected signals, we propose a *conservative clustering, cumulative aggregation* approach, which consists of two stages. First, we perform conservative clustering by employing a relatively small distance threshold, generating small and possibly incomplete clusters while avoiding aggressively grouping points from different objects into a cluster. Second, we accumulate the incomplete clusters for the same object from consecutive frames and then carefully aggregate them into a more complete cluster of each car. In this way, we can effectively cope with the unstable points from reflected signals and accurately extract the bounding box of an occluded car. Fig. 15 shows the pipeline of our algorithm.

1) Per-Frame Point Clustering: We employ the DBSCAN algorithm [24], which is often used for radar point clustering under the assumption that a real target consists of points that are near each other. DBSCAN determines the physical distance between each pair of points, and points within a preset threshold are assigned to the same cluster. This threshold, denoted by $T_{cluster}$, is a critical parameter for the algorithm.

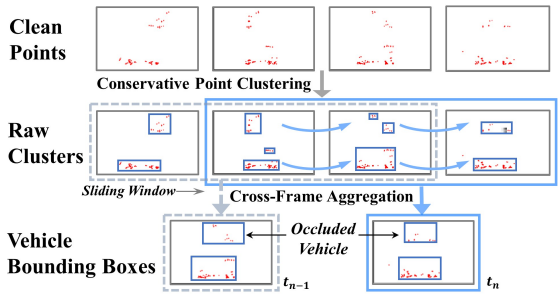


Fig. 15: Illustration of the cross-frame cumulative aggregation approach.

We adopt a relatively small $T_{cluster}$ value, i.e., 0.5 m in our current implementation. This decision considers the fact that vehicles typically maintain a lateral gap of over 1 m for safety. Consequently, we use half of this shortest lateral distance as the distance threshold to ensure that point clouds from two vehicles are not clustered into one target and allows for some positioning error in the radar point cloud.

2) Cross-Frame Cumulative Cluster Aggregation: We establish a sliding window of length W, which continuously holds the clustering results of the latest W frames. We then perform cumulative cluster aggregation as follows:

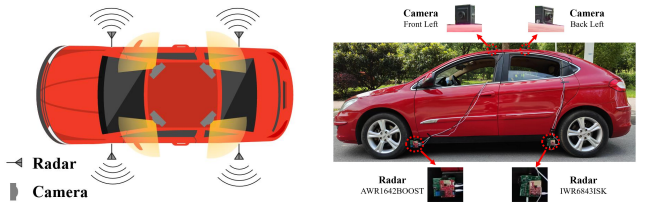
- *Cluster Aggregation*. We sequentially fetch frames in the sliding window and match clusters across frames based on the center distances. The algorithm works by selecting the pair of clusters with the closest center distance and terminates when the closest center distance exceeds the threshold of $2T_{cluster}$ (i.e., 1 m). This choice considers the fact that vehicles typically have a center distance of over 2 m from each other. Each matched pair of clusters is considered to belong to the same object and is merged into one. This matching and merging operation is iterative by nature as we may have more than one cluster in a frame due to our conservative clustering approach. We continue to perform this action until all the frames in the aggregation window are processed.
- *Cluster Cleansing*. We then eliminate those clusters that have had no match in the previous step, assuming they are false targets. Finally, we check through the bounding boxes of all the remaining clusters. If two boxes intersect, we merge them into a single cluster.

F. Occlusion Detection and Intent Prediction

We further investigate how the proposed detection results can support downstream decision-making in autonomous driving scenarios. Specifically, two functionalities are considered: occlusion alerting and intent prediction for occluded vehicles.

First, we analyze the spatial relationships among detected bounding boxes to determine whether a target is occluded by another object. Fig. 15 shows representative examples. To improve robustness against occasional false positives, a temporal filtering strategy is employed: an occlusion alert is triggered only when an occluded vehicle is consistently detected over N consecutive frames. The parameter N is empirically set to 10, as discussed in Section IV-D.

Once an occlusion is confirmed, intent prediction is performed based on the same N-frame temporal window. For each detected vehicle, a short trajectory is constructed from the



(a) Hardware setup (b) Prototype *mmOVD*$^+$

Fig. 16: (a) Illustration of hardware setup of *mmOVD*$^+$. (b) Photo of the left side of our sensing vehicle.

sequence of detection outputs, represented as $\{(x_t, y_t, v_t)\}_{t=1}^{N}$, where (x_t, y_t) and v_t denote the estimated position and velocity at frame t, computed by averaging the point cloud measurements associated with the same target cluster.

Based on this trajectory, a compact set of motion features is extracted to characterize the vehicle's dynamics. First-order motion is captured by frame-wise differences $\Delta x_t = x_t - x_{t-1}$ and $\Delta y_t = y_t - y_{t-1}$. The mean and standard deviation of Δx_t and Δy_t are computed to reflect the average motion trend and temporal variability. In addition, global displacement features are included as $x_N - x_1$ and $y_N - y_1$. Finally, velocity statistics, including the mean and standard deviation of $\{v_t\}_{t=1}^{N}$, are incorporated to capture motion magnitude and stability.

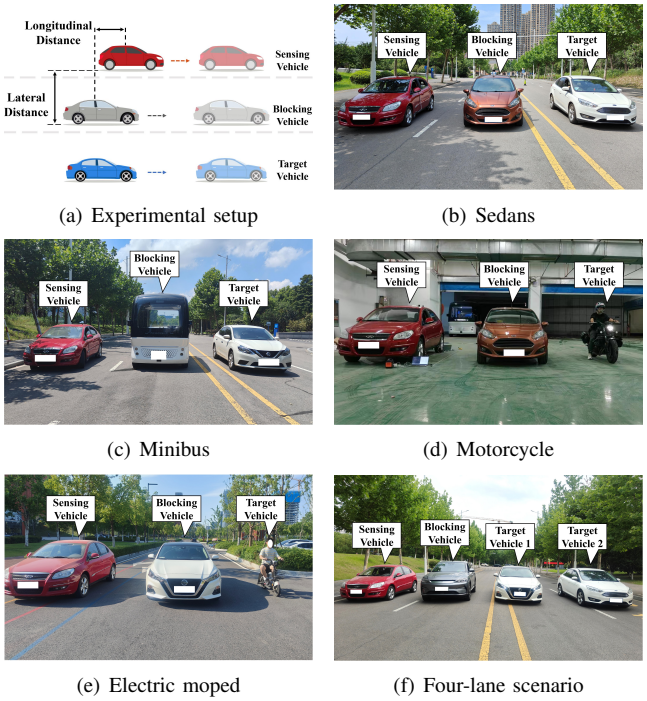
These features are concatenated into a fixed-length representation and fed into a lightweight classifier (a random forest in our implementation) for intent prediction. In this work, the intent is defined as a trajectory-level motion pattern inferred from short-term observations, and is categorized into three types: *overtaking*, *being overtaken*, and *parallel driving*.

IV. Evaluation through Controlled Experiments

A. Experimental Setup

1) Prototype mmOVD$^+$: We have implemented the prototype *mmOVD*$^+$ system using two different types of mmWave radar: the TI AWR1642BOOST [46], operating in the frequency band of 77-81 GHz, and the TI IWR6843ISK [47], operating in 60-64 GHz. This choice ensures that they do not interfere with each other. To achieve an optimal field of view, we position the radars near the four tires on the side of the vehicle. The radars are configured to achieve a theoretical range resolution of 9.75 cm, a velocity resolution of 0.41 m/s, and an azimuth resolution of 15°. Each radar is connected to a TI DCA1000EVM data capture card [48] for achieving high-speed data transmission. Additionally, to emulate a typical vehicle detection system, we install 4 cameras on the experiment vehicle's roof, inspired by nuScenes [13]. Please note that these cameras are not part of our system but are solely used for performance comparison and verification of our system. The hardware setup is depicted in Fig. 16.

2) Network Implementation: We implement *GhostNet* using the PyTorch framework, based on the PointNet++ architecture with multi-scale grouping. The model is trained on an NVIDIA RTX 3060 GPU with a batch size of 8. We adopt the Adam optimizer with an initial learning rate of 1×10^{-3}. The training stage is conducted over a maximum of 50 epochs. A cross-entropy loss is used to supervise the point-wise binary classification.



(a) Experimental setup (b) Sedans

(c) Minibus (d) Motorcycle

(e) Electric moped (f) Four-lane scenario

Fig. 17: (a) Illustration of the experimental setup. (b) - (f) Pictures of five controlled experiment setups.

3) Experiment Settings: To address potential occlusion scenarios in real driving situations, we examine several important parameters of an occlusion scenario: (1) The longitudinal distance between vehicles, (2) the lateral distance between vehicles, (3) the types of blocking/occluded vehicles, and (4) the driving patterns of vehicles. Varying these parameters, our evaluation involves experiments conducted across 70 diverse scenes, as illustrated in Fig. 17. These experiments were conducted in an underground garage and on two campus roads, and can be categorized into three types of settings:

- *Setting I: Semi-Static Occlusion Scenarios.* In this setting, we construct various occlusion situations under different parameter settings. We keep the sensing and blocking vehicles stationary while moving the target vehicle. Initially, we position the sensing vehicle and the blocking vehicle side by side in two adjacent lanes. The longitudinal distance between them varies from -2.5 meters to 2.5 meters, while the lateral distance ranges from 1 meter to 2.5 meters. Subsequently, we drive the target vehicle along the third lane, capturing the entire process as it enters and exits the blind zone. Our aim is to precisely detect the target vehicle within the blind zone. In this set of experiments, we recorded a total of 90 minutes of data across 30 distinct scenes. The vehicles we used include 4 sedans, 2 SUVs, and 1 minibus.
- *Setting II: Dynamic Driving Scenarios.* In this setting, we drive the sensing vehicle, blocking vehicle, and target vehicle on campus roads in the following four scenarios: (1) Three vehicles drive side by side at similar speeds; (2) the sensing vehicle and blocking vehicle drive at similar speeds, while the target vehicle overtakes from behind; (3) the sensing vehicle and target vehicle drive at similar speeds, while the blocking vehicle overtakes from behind; (4) the blocking vehicle and target vehicle drive at similar speeds, while the sensing vehicle

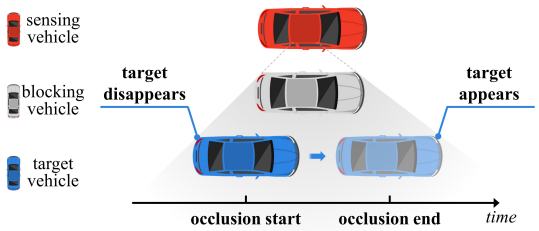


Fig. 18: Illustration of an occlusion event. We get the start and end of an event by monitoring when the target vehicle disappears from cameras and when it reappears.

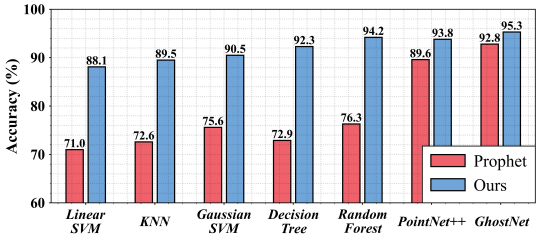


Fig. 19: Comparison of point classification accuracies using our feature set vs. the baseline feature set [36] across different classifiers.

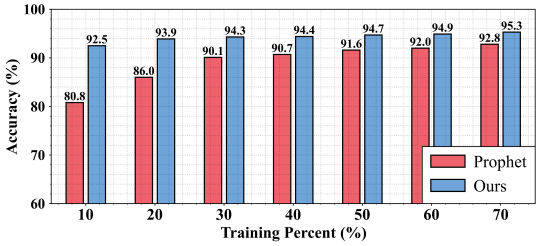


Fig. 20: Comparison of point classification accuracies using our feature set vs. the baseline feature set [36] under different training percent of data.

overtakes from behind; and (5) only the sensing vehicle and the blocking vehicle drive at similar speeds, with no occluded vehicles within the blind zone. Our aim is to assess the system's performance in different driving patterns and to verify the effectiveness of our ghost removal scheme. We have recorded 125 minutes of data across 25 different scenes, encompassing various vehicle types, road environments, and driving patterns. Out of the 125 minutes of data, 80 minutes involve occlusion conditions. The vehicles we used include 3 sedans and 2 SUVs, with speeds ranging from 20 to 40 km/h.

• *Setting III: Challenging Occlusion Scenarios.* To further evaluate the robustness of our system, we design several challenging occlusion scenes based on the semi-static setting. These include (1) using a motorcycle and (2) an electric moped as the target vehicles, both of which have smaller radar cross-sections and are harder to detect; and (3) simulating a four-lane scenario where two target vehicles are simultaneously occluded by a single blocking vehicle. In all cases, the sensing and blocking vehicles remain stationary while the target vehicles move through the blind zone. These scenarios test the system's capability in handling small and multiple occluded objects. We recorded a total of 45 minutes of data across 15 distinct scenes. The vehicles we used include 3 sedans, 1 SUV, 1 motorcycle, and 1 electric moped.

4) Experiment Data Preparation: To summarize, we conducted our controlled experiments in an underground garage and on two campus roads, generating a total of over 200 minutes of data. We deployed a 32-line LiDAR on the roof of the blocking vehicle to obtain bounding boxes of target vehicles. The results obtained from the LiDAR are considered ground truth, which will be used for labeling the radar points.

The labeled radar data is then split into two distinct sets: a training set comprising 180 minutes of data, and a testing set comprising 80 minutes of data. The scenarios of the training set and the testing set do *not* overlap. The training set is used to train the point classifier and the test set is used for evaluation.

5) Performance Metrics: We adopt the following metrics to evaluate the proposed *mmOVD*$^+$ system:

• *Accuracy of Point Classification*: The points generated by real vehicles are labeled as *positive*, while the ghost points and noises are labeled as *negative*. Therefore, the correctly classified points are counted as true positive (TP) and true negative (TN), while the incorrect ones are false positive (FP) and false negative (FN). The accuracy is defined as $\frac{TP+TN}{TP+TN+FP+FN}$.

• *Accuracy of Occluded Vehicle Detection*: As shown in Section III-E, we extract the bounding box of the target vehicle from point clouds. If the detected bounding box is inside the ground-truth bounding box (which is obtained by the LiDAR), it is counted as a true positive (TP). Otherwise, the detections and ground-truth objects are counted as false positives (FP) and false negatives (FN), respectively. We mainly use the F1 score to assess the accuracy, which is the harmonic mean of precision (P) and recall (R), i.e., $F1 = \frac{2*P*R}{P+R}$.

• *Success Rate of Occlusion Event Detection*: We first define an *occlusion event* as the entire process of the target vehicle entering and exiting the blind zone, as shown in Fig. 18. By monitoring when the target vehicles disappear from cameras' fields of view and subsequently reappear, we can determine the ground-truth start and end times of each occlusion event. *mmOVD*$^+$ triggers an alarm when it detects an occluded vehicle in N consecutive frames (where N is a preset alarm threshold). If *mmOVD*$^+$ raises an alarm for a true event, we consider the event as successfully detected. The success rate is calculated as the ratio between the number of detected occlusion events and the total number of occlusion events. To find the appropriate threshold N for achieving a high success rate while mitigating false alarms, we perform statistical analysis on false alarms in no occlusion scenarios.

B. Radar Point Classification

Firstly, we evaluate the accuracy of point classification with our feature set under different classifiers and compare the results with the feature set proposed by Prophet et al. [36]. The comparison results are shown in Fig. 19, which shows that our feature set outperforms the baseline by a large margin, particularly when used with classical machine learning models. For example, our feature set, when combined with the decision tree model, can improve the accuracy from 72.9% to 92.3%, accounting for a 19.4% improvement. In contrast, the performance gap narrows when deep learning models such as PointNet++ and *GhostNet* are employed. This is because their sampling and grouping layers can effectively capture both local and global spatial structures, enabling high classification accuracy even without handcrafted features. Nevertheless, incorporating our feature set still yields noticeable

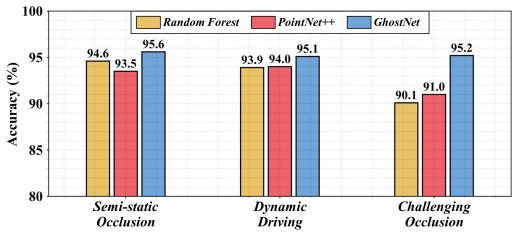


Fig. 21: Comparison of point classification accuracy of *GhostNet* and baseline classifiers across three experimental settings.

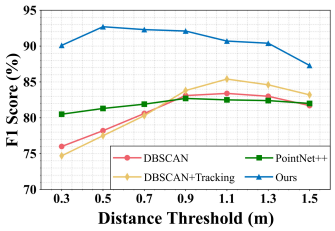


Fig. 22: Comparison of *mmOVD*$^+$ with the baseline schemes in terms of vehicle detection accuracy.

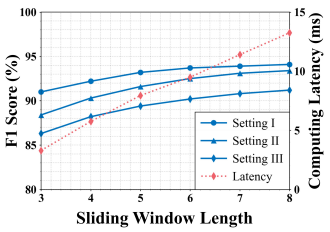


Fig. 23: F1 score and computing latency of detection under different sliding window lengths.

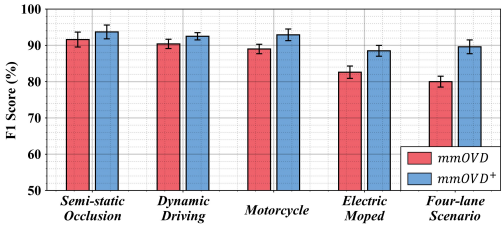


Fig. 24: Comparison of vehicle detection F1 scores using *mmOVD*$^+$ vs. *mmOVD* [51] across different scenarios. *mmOVD*$^+$ achieves notable improvements, especially in the last three challenging scenarios.

improvements, demonstrating its complementary benefit even on top of strong deep-learning backbones.

To further demonstrate the robustness of our feature set, we evaluate classification accuracy under varying training data proportions, as shown in Fig. 20, with *GhostNet* as the classifier. The results indicate that our feature set provides greater benefits when training data is limited. Notably, even with only 10% of the data used for training, the system achieves an accuracy of 92.5% using our feature set. This highlights its ability to extract discriminative point features that accelerate model convergence and improve generalization.

The above analysis demonstrates the effectiveness and robustness of the proposed feature set. We next examine the contribution of *GhostNet*. Specifically, we compare it with the two best-performing baseline models—Random Forest and PointNet++—across the three experimental settings, as shown in Fig. 21. In the relatively simple Settings I and II, all methods achieve similarly high accuracy, and *GhostNet* provides only marginal improvements (1%–2.1%). This is because the task is less challenging and the performance is already close to saturation when using the proposed feature set. In contrast, in the most challenging Setting III, which includes small targets and complex multi-vehicle scenarios, *GhostNet* achieves a clear performance gain (4.2%–5.1%). More importantly, its accuracy remains comparable to that in other settings, whereas the baseline methods degrade noticeably. These results demonstrate that *GhostNet* significantly improves robustness under challenging conditions and effectively narrows the performance gap across different scenarios, thereby extending the applicability of the system beyond relatively simple settings.

C. Occluded Vehicle Detection

We then evaluate the overall performance of our cross-frame aggregation detection algorithm and two baseline schemes, illustrated in Fig. 22. The blue line represents the performance of solely utilizing DBSCAN for vehicle detection under different distance thresholds, showing the optimal F1 score achieved is 83.4%. We also apply a typical tracking algorithm, AB3DMOT [49], on top of DBSCAN. The results are represented by the yellow line, with an optimal F1 score of 85.4%. In addition, we evaluated the full version of PointNet++ (with both classification and segmentation) for vehicle detection, by varying the distance thresholds in its grouping module. The results indicate that PointNet++ performs poorly in challenging occlusion scenarios with sparse radar points, achieving a best F1 score of only 82.7%. Meanwhile, our method outperforms the baselines significantly, with an optimal F1 score of 92.7%.

In addition, we also evaluate the performance of our cross-frame aggregation-based vehicle detection algorithm under different sliding window sizes. We report the F1 score and average computing latency in Fig. 23. The results demonstrate that the performance of the *mmOVD*$^+$ becomes better as the length of the sliding window increases. Specifically, when the length of the sliding window reaches 6, the F1 score of all three settings exceed 90%. Notably, the F1 score of Setting II is always slightly lower than Setting I, while the gap decreases as the window length increases. This difference arises because Setting II involves a rather challenging occlusion scenario, where three vehicles are driving side by side at similar speeds for 20 minutes. However, we note that vehicles driving side by side for an extended period of time are not common in the real world as it is considered a bad driving habit [50]. Besides, Setting III yields the lowest F1 scores, as it includes small and weakly reflective targets (e.g., motorcycles and electric mopeds) and multi-object occlusions. Despite this, our system still maintains strong performance, achieving an F1 scores of 90.2% under these challenging conditions.

Furthermore, we observe a nearly linear increase in the computing latency with the increase in sliding window length. This is expected, as the computational load of our detection algorithm scales proportionally with the window size. Given that the radar operates at 30 frames per second, it is essential to keep the per-frame latency below 30 ms. Empirical measurements indicate that point feature extraction and network inference require 10.7 ms and 5.6 ms per frame, respectively. Based on these results, we select a sliding window length of 6 frames, which yields an average per-frame latency of 9.49 ms and a total system runtime of 25.8 ms per frame. This allows the system to operate in real time at a 30 Hz radar frame rate. All experiments were conducted on a desktop with an 11th Gen Intel Core i5-11500 CPU (2.70 GHz) and an NVIDIA GeForce RTX 3060 GPU. The GPU is used solely for training *GhostNet*, whereas all module inferences during runtime are performed on the CPU, demonstrating that the system achieves real-time performance without specialized hardware.

In Setting I, we construct various occlusion scenarios under three important parameters: longitudinal distance, lateral dis-

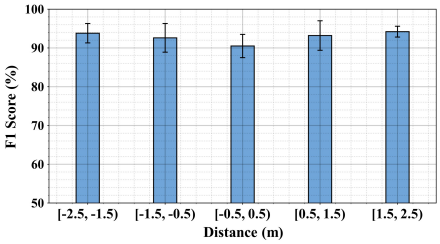


Fig. 25: F1 score of vehicle detection under different longitudinal distances between sensing and blocking vehicles.

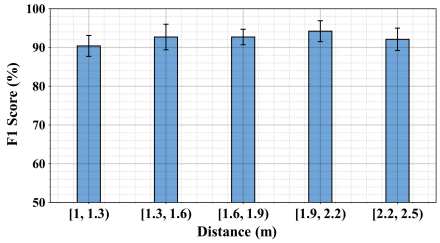


Fig. 26: F1 score of vehicle detection under different lateral distances between sensing and blocking vehicles.

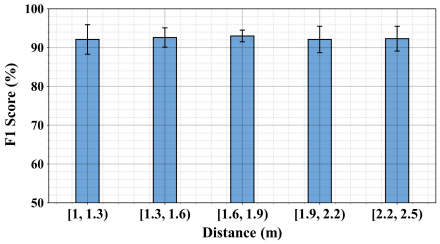


Fig. 27: F1 score of vehicle detection under different lateral distances between blocking and target vehicles.

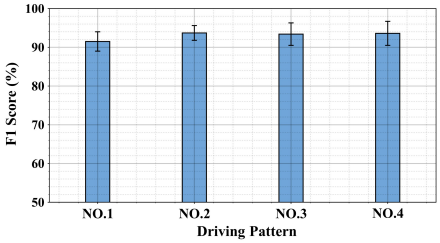


Fig. 28: F1 score of vehicle detection under different driving patterns.

tance, and vehicle type. In Setting II, we conduct experiments in different driving patterns. In Setting III, we conduct experiments in three challenging scenarios. We then quantify the influences of these parameters on occluded vehicle detection.

1) Detection Accuracy under Different Scenario Types: Fig. 24 shows the vehicle detection F1 scores across different scenarios. In both *Semi-static Occlusion* and *Dynamic Driving* scenarios, *mmOVD*$^+$ consistently achieves F1 scores above 92%. Moreover, when detecting smaller occluded targets such as a motorcycle or an electric moped, it maintains high performance with F1 scores of 92.9% and 88.5%, respectively. Even in the complex *Four-lane Scenario*, where two target vehicles are simultaneously occluded by a single blocker, *mmOVD*$^+$ still achieves an F1 score of 89.6%, demonstrating strong robustness under diverse and challenging conditions.

In addition, we compare *mmOVD*$^+$ with our previous approach *mmOVD* [51], as shown in Fig. 24. Across all five scenarios, *mmOVD*$^+$ achieves consistent performance improvements. In particular, the gains are more pronounced in challenging cases: the F1 score increases by 6.0% in the *Electric Moped* scenario and by 9.6% in the *Four-lane Scenario*. This improvement stems from the integration of handcrafted features with deep learning models in *mmOVD*$^+$, which enables more comprehensive extraction of spatial features from radar point clouds. As a result, the system remains effective even when the target vehicle exhibits weak reflections and only sparse true points are present.

2) Impact of Longitudinal Distance Between Vehicles: The longitudinal distance between the sensing and blocking vehicles plays a crucial role in determining the extent of the blind zone. Specifically, we define the longitudinal distance as negative when the blocking vehicle is positioned behind sensing vehicle, and positive when it is ahead. Fig. 25 shows the system's performance at different longitudinal distances. As the longitudinal distance approaches 0 (i.e., two vehicles are side by side), the F1 score of vehicle detection decreases by 3.3%, 2.1%, 2.8%, and 3.7%, respectively, compared to other four distance settings. The reason is that when the two vehicles are completely parallel, the blind zone becomes the largest, leading to the occlusion of the target vehicle for a longer duration. However, we note that this situation is not common in the real world [50].

3) Impact of Lateral Distance Between Vehicles: The lateral distance between the sensing and blocking vehicles is another important parameter to consider. As we focus on the occlusion scenarios, we assume that all three vehicles stay within their respective lanes and maintain a safe lateral distance of at least 1 meter between them. Fig. 26 shows the system's performance under different lateral distances between the sensing vehicle and the blocking vehicle. It can be seen that when the distance is smaller than 1.3 meters, the F1 score is 1.7% to 3.8% lower compared to other situations. This decline is caused by the stronger multi-path effect and more severe occlusion when the blocking vehicle gets too close. However, vehicles driving in adjacent lanes usually do not maintain such a short distance in the real world. Additionally, Fig. 27 presents the system's performance under different lateral distances between the blocking vehicle and the target vehicle. The F1 scores consistently exceed 92%, and there are no notable fluctuations across different situations. Collectively, the results demonstrate that *mmOVD*$^+$ is rather robust against varying lateral distances.

4) Impact of Driving Patterns: In Setting II, we emulate real driving scenarios on campus roads using six distinct experimental setups. In the first three setups, we employ an SUV as the blocking vehicle and a sedan as the target vehicle. Each setup features different driving patterns: (1) Three vehicles drive side by side at similar speeds; (2) the sensing vehicle and blocking vehicle drive at similar speeds, while the target vehicle overtakes from behind; (3) the sensing vehicle and target vehicle drive at similar speeds, while the blocking vehicle overtakes from behind; and (4) the blocking vehicle and target vehicle drive at similar speeds, while the sensing vehicle overtakes from behind. Subsequently, we exchange the two vehicles and repeat the experiments. The performance of different driving patterns is shown in Fig. 28. Notably, in driving pattern 1, in which three vehicles are driving side by side, the target vehicle remains continuously

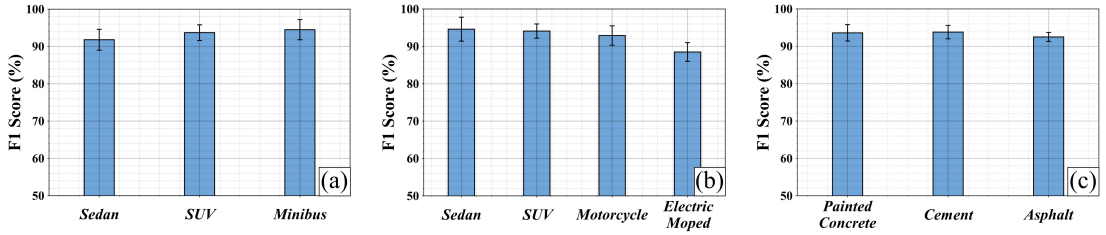


Fig. 29: F1 score of vehicle detection under different (a) types of blocking vehicles, (b) types of target vehicles, and (c) types of roads.

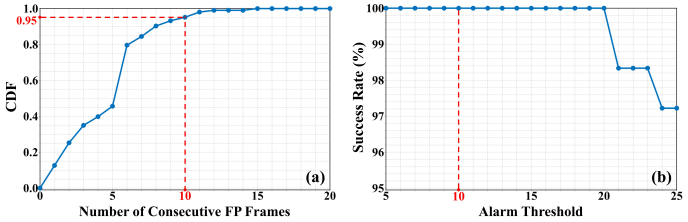


Fig. 30: A 100% success rate and a 5% false alarm rate are achieved with the alarm threshold set to 10. (a) CDF of consecutive FP detection number when there is no occlusion. (b) Success rate of occlusion event detection under different alarm threshold values.

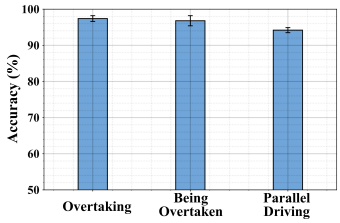


Fig. 31: Intent prediction accuracy for three motion patterns: overtaking, being overtaken, and parallel driving.

occluded. Consequently, the F1 score of driving pattern 1 is 1.9% to 2.2% lower than in the other patterns but still exceeds 91%, showcasing the ability of *mmOVD*$^+$ to handle long-term occlusion scenarios effectively.

5) Impact of Vehicle Type: The type of vehicles can also influence the detection of occluded vehicles, due to their varying shapes and sizes. Therefore, we conduct experiments using various vehicle types. In addition to common sedans and SUVs, we include a minibus as the blocking vehicle and a motorcycle and an electric moped as target vehicles.

The results presented in Fig. 29(a) indicate that when the blocking vehicle is a sedan, the F1 score is 1.1% to 1.6% lower compared to other types. This is attributed to sedans typically having lower ground clearance (around 15 cm), compared to SUVs and the minibus (around 20 cm), which limits the propagation space for radar signals beneath the vehicle and leads to weaker ground reflections. In addition, a lower chassis constrains the effective range of incident angles at which the radar signals can reach and be reflected by the ground, making the detection of occluded vehicles more challenging. Nevertheless, the system still achieves an F1 score of around 92% when sedans act as blocking vehicles, demonstrating its effectiveness under such conditions.

Fig. 29(b) shows that the F1 score drops when the occluded target is a motorcycle or an electric moped. This performance degradation is mainly due to the smaller size and weaker reflectivity of these vehicles, which lead to fewer and less distinctive radar points. Nevertheless, our system still maintains reasonable detection accuracy in these cases, demonstrating its robustness in handling challenging small-object scenarios.

6) Impact of Road Type: The type of road surface can also affect occluded vehicle detection, due to differences in surface roughness and reflectivity. In our experimental sites, the underground garage had a smooth painted concrete floor, while the two campus roads included both cement and asphalt surfaces. To eliminate the influence of scene difficulty, this evaluation only uses data from Setting I and Setting II, excluding the more challenging scenarios. The results in Fig. 29(c) show that the average F1 scores are 93.6% on the painted concrete floor, 93.8% on cement roads, and 92.5% on asphalt roads. These results indicate that our system can adapt to a variety of typical road surfaces. The slightly lower performance on asphalt roads may be attributed to its rougher surface, which leads to greater signal attenuation during signal reflection.

D. Occlusion Event Detection

As *mmOVD*$^+$ triggers an alarm when it detects an occluded vehicle in N consecutive frames, the choice of the alarm threshold N is crucial. On one hand, if N is too small, it may result in a high false alarm rate. On the other hand, if N is too large, it may lead to missed alarms and delayed responses. In this work, we attempt to find the smallest N that gives a low enough (say, 5%) false alarm rate. In order to find the threshold value that gives a certain false alarm rate, we collect additional data from Setting II when there is no occlusion. Here, we drive the sensing vehicle and the blocking vehicle side by side on the campus road, ensuring that there are no occluded vehicles within the blind zone. In this set of experiments, we recorded 45 minutes of data across 15 distinct scenes.

Specifically, we calculate the likelihood of having N consecutive false positive (FP) detections when there is no occlusion. Then we obtain the CDF and present the results in Fig. 30(a). We observe that when we choose $N = 10$, the corresponding CDF is 95%. That is, in an unknown situation, when we detect the presence of a hidden vehicle from 10 or more frames consecutively, the likelihood of having no occlusion is lower than 5%, thus the false alarm rate of 5%. As a result, we can choose the alarm threshold $N = 10$ in our system—*mmOVD*$^+$ triggers an alarm when it detects an occluded vehicle in 10 consecutive frames. Notably, although increasing the value of N could further reduce the false alarm rate while keeping a 100% of success rate, it would also delay the response time. We chose N to be 10 to ensure that *mmOVD*$^+$ maintains a short response latency.

Furthermore, we also plot the system's success rate of occlusion event detection with different alarm threshold values, shown in Fig. 30(b). With the alarm threshold set to 10, *mmOVD*$^+$ achieves a 100% success rate of occlusion event

detection. As a result, we conclude $N = 10$ is a suitable threshold for issuing occlusion alarms for it achieves a 100% success rate while mitigating the false alarm rate to be lower than 5% in our dataset.

In addition to occlusion event detection, we further evaluate the effectiveness of the proposed method for intent prediction. Specifically, all detected occlusion events are randomly split into training and testing sets with a ratio of 7:3. To ensure generalization, the testing set consists of continuous temporal segments rather than randomly sampled frames. The results are shown in Fig. 31. The overall prediction accuracy reaches 96.1%, demonstrating that the proposed system can reliably infer vehicle intent from occluded observations. This performance is enabled by the complementary use of spatial and velocity information. In addition to estimating the position of occluded vehicles, the system leverages Doppler measurements from millimeter-wave radar to obtain velocity, which provides critical cues for motion analysis. By jointly modeling the temporal evolution of spatial positions and velocities, the system can effectively distinguish different motion patterns.

Among the three categories, the accuracy of parallel driving is slightly lower (by approximately 3%) than the other two. This is mainly because vehicles in parallel driving scenarios do not always maintain perfectly stable velocities, and occasional speed fluctuations may lead to ambiguous motion patterns and misclassifications. In practical autonomous driving systems, incorporating longer temporal context or higher-level reasoning may further improve the robustness of intent prediction.

V. Occlusion Detection in the Real World

After conducting extensive controlled experiments, we next assess the performance of *mmOVD*$^+$ while we drive the sensing vehicle in the real world.

A. Driving Scenarios and Data Acquisition

We drove multiple trips from our campus to a nearby town with a sensing car equipped with the *mmOVD*$^+$ prototype. The route included parkways, tunnels, freeways, overpasses, and bridges, most with three or four lanes (as shown in Fig. 32). We drove a total distance of *324 kilometers* over *580 minutes*, reaching a maximum speed of approximately *70 km/h*. We acquired 153,000 frames of valid data. We manually annotated the occlusion events by examining the camera images frame by frame. Our dataset comprises 226 occlusion events, each lasting between 0.7 and 4 seconds.

B. Occlusion Event Detection while Driving

1) Overall Performance: To evaluate our system's overall performance with different training data amounts, we vary the training data volume from 10% to 70% of the total data and report the detection success rate and false alarm rate in Fig. 33. For each data point, we conducted 10 experiments, each experiment using randomly sampled data. We discover that *even with just 10% of the total data (i.e., 15,300 frames) used for training, our system achieved a detection success rate of 93.8%*. Furthermore, the false alarm rate of the system remained consistently low ($< 4\%$), showcasing the robustness of our system when operating in real-world environments.

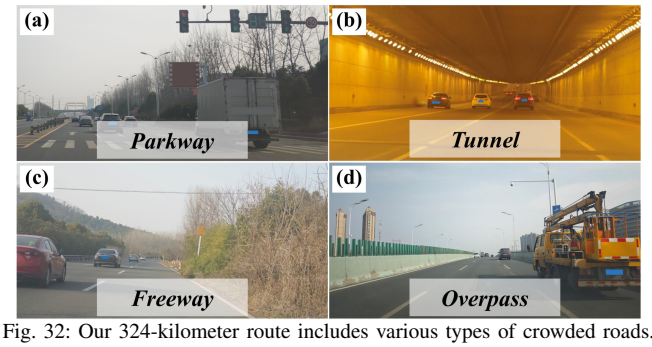


Fig. 32: Our 324-kilometer route includes various types of crowded roads.

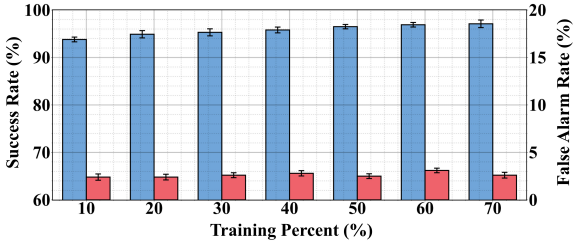


Fig. 33: Success rate (blue) and false alarm rate (pink) of real-world experiments under different training percent of data.

Additionally, each occlusion commonly lasts 2-4 seconds in real-world scenarios, and *mmOVD*$^+$ triggers a correct alarm on average within 0.5 seconds after the occlusion starts.

To further evaluate the effectiveness of intent prediction in real-world driving, we conduct additional experiments on the collected dataset. For safety considerations, the sensing vehicle was driven in the rightmost lane, and thus the recorded occlusion events are overwhelmingly *overtaking* scenarios. We randomly select 10% of continuous temporal segments as training data to fine-tune the intent prediction model trained in controlled experiments. Evaluation on the remaining data shows that the prediction accuracy reaches 95.1%, demonstrating the effectiveness of the proposed method in the real world.

2) Real-World Occlusion Event Analysis: In Fig. 34(a), we plot the number of occlusion events detected within a minute over a 30-minute period, collected while the vehicle was driving on a busy overpass. The yellow bars represent the ground-truth number of occlusion events during each time interval, while the brown and blue lines represent the average number of successfully detected occlusion events and falsely alarmed occlusion events during that interval, respectively. This figure demonstrates that our system exhibits excellent performance even in crowded scenarios. As for the occasional errors, we found that they were mainly caused by situations not considered in the controlled experiments. For example, we observed several trucks being occluded that our system failed to detect. The reason is that their chassis are much higher than common vehicles, and are beyond the radars' detection range. Fortunately, these trucks are typically tall enough for cameras to detect even when occluded by other vehicles.

In Fig. 34(b), we show the fine-grained per-frame occluded vehicle detection result over a 600-frame (20 seconds) period, containing 4 occlusion events. Due to the complexity of crowded road environments, occasional per-frame false positives may occur, which is expected. However, our event-level detection mechanism can effectively accommodate these occasional per-frame detection errors.

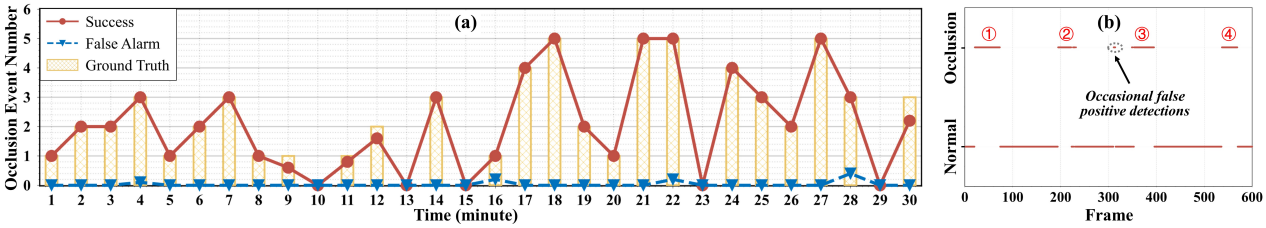


Fig. 34: (a) The occlusion event detection results over a 30-minute period. (b) The occluded vehicle detection results over a 600-frame period.

3) Performance on Wet Roads: During real-world driving data collection, one day occurred shortly after rainfall. On that day, the roads were wet and contained multiple puddles. The occlusion event detection success rate was 96.9%, with a false alarm rate of 2.6%, using 70% training data. This performance is comparable to that observed under dry conditions. These results demonstrate that *mmOVD*$^+$ can operate effectively on moderately wet surfaces. Note that our radars lack waterproofing, so data under heavy rain or snow could not be collected. Evaluating system performance under such extreme weather conditions remains a direction for future research.

VI. Discussion

A. Generalization

As shown in Sec. IV-B and Sec. V-B, our system maintains high accuracy even with a limited amount of training data. This indicates strong generalization capability, as the learned features are largely environment-independent, enabling robust performance in unseen scenarios. Moreover, our ghost point removal method is designed based on the physical causes and behavioral patterns of multi-path ghosts, making it potentially applicable to a broad range of mmWave radar sensing tasks beyond the specific use cases evaluated in this paper.

In addition, our controlled experiments correspond to relatively sparse traffic conditions, where only a few vehicles contribute to reflections. In contrast, real-world driving experiments on busy urban roads involve dense traffic, where multiple surrounding vehicles introduce complex reflections. Using the same 70% training data, the occlusion event detection success rate only slightly decreased from 100% in controlled experiments to 97.1% in real-world scenarios. This demonstrates that *mmOVD*$^+$ is robust across different traffic conditions, further confirming its generalization capability.

From a geometric perspective, the effectiveness of the proposed method relies on the existence of a valid underbody propagation and reflection path between the radar, the blocking vehicle, and the occluded vehicle. As long as such a path exists, the radar signals can propagate beneath the blocking vehicle and generate detectable reflections from the occluded target. This underlying mechanism enables the system to maintain consistent performance across different traffic scenarios.

B. Deep Learning Model

The ghost point removal task can be formulated as a point-wise binary classification problem over sparse radar point clouds. This fundamentally requires effective point cloud feature extraction and classification. In recent years, attention-based models such as transformers have gained popularity in detection and classification tasks due to their strong global modeling capabilities. However, these models often incur significant computational overhead, especially when applied to sparse, noisy data like radar point clouds.

In our system, we adopt PointNet++ as the backbone of *GhostNet* to balance computational efficiency and classification accuracy. Its hierarchical structure is well-suited for sparse, unordered point clouds, effectively capturing both local and global features. Compared to attention-based models, PointNet++ offers lower latency and a lighter architecture, making it more suitable for our real-time detection task.

C. Limitations

While *mmOVD*$^+$ has achieved high performance in many scenarios, it is important to acknowledge its limitations.

First, although *mmOVD*$^+$ can detect small vehicles such as motorcycles and electric mopeds, its ability in identifying even smaller and less reflective objects—such as bicycles and pedestrians—remains limited. This is partly due to the use of commercial off-the-shelf radars with standard signal processing pipelines, which constrain the quality and density of the generated point clouds. Enhancing point cloud resolution or developing customized signal processing techniques may help improve detection of such weak targets in the future.

Second, the current system is validated on a desktop computer with ample computational resources. However, real-world deployment of *mmOVD*$^+$ may require running the full pipeline on resource-constrained edge devices installed on vehicles. This poses challenges for maintaining real-time performance, especially given the computation involved in point-wise classification and multi-frame clustering. Further optimization of the algorithms and potential model compression will be necessary to enable efficient and reliable deployment in practical edge environments.

Third, *mmOVD*$^+$ is primarily evaluated in occlusion scenarios where vehicles move in the same direction in adjacent lanes, a common form of inter-vehicle occlusion. Other occlusion cases, such as vehicles emerging from behind a blocking car at intersections, also frequently occur in practice. In these scenarios, vehicles approaching from lateral directions can still generate detectable reflections. However, due to the rapid antenna gain attenuation at larger azimuth angles of the current radar [52], the resulting point clouds are sparse, making reliable detection more challenging. Extending the system to such scenarios with higher-performance automotive radars is an important direction for future work.

VII. Conclusion

We have presented *mmOVD*$^+$, a novel system for detecting fully occluded vehicles using mmWave radars. *mmOVD*$^+$ takes

advantage of the multipath in the environment to see through vehicles, while addressing the challenges due to ghost points and unstable reflections. For the first time, we can accurately detect and localize an "unseeable" vehicle. Extensive experiments in both controlled and real-world scenarios validate its effectiveness. Specifically, *mmOVD*$^+$ achieves an average point classification accuracy of 95.3% and an F1 score of 92.7% for occluded vehicle detection, outperforming baseline methods by 7.3%–10%. It also attains a 100% success rate for occlusion event detection in controlled experiments and maintains a 93.7% success rate in real-world driving with only 10% training data, demonstrating strong robustness and generalization. Future work will focus on improving detection of weak targets, enabling real-time deployment on edge devices, and extending to more diverse occlusion scenarios. We believe that the design, implementation, and evaluation of our results present important practical contributions toward safer autonomous driving.

REFERENCES

[1] J. Li, H.-C. Wong, S.-L. Lo, and Y. Xin, "Multiple object detection by a deformable part-based model and an r-cnn," *IEEE SPL*, vol. 25, no. 2, pp. 288–292, 2018.

[2] W. Zhang, Q. J. Wu, X. Yang, and X. Fang, "Multilevel framework to detect and handle vehicle occlusion," *IEEE T-ITS*, vol. 9, no. 1, pp. 161–174, 2008.

[3] NHTSA, "Traffic safety facts 2022," 2025, https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813656.

[4] H. Van Pham and B.-R. Lee, "Front-view car detection and counting with occlusion in dense traffic flow," *Int. J. Control Autom. Syst.*, vol. 13, no. 5, pp. 1150–1160, 2015.

[5] A. Shastri, N. Valecha, E. Bashirov, H. Tataria, M. Lentmaier, F. Tufvesson, M. Rossi, and P. Casari, "A review of millimeter wave device-based localization and device-free sensing technologies and applications," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1708–1749, 2022.

[6] F. Zhang, Z. Zhang, L. Kang, A. Zhou, and H. Ma, "mmtaa: A contact-less thoracoabdominal asynchrony measurement system based on mmwave sensing," *IEEE Transactions on Mobile Computing*, vol. 24, no. 2, pp. 627–641, 2024.

[7] C. Meng, Y. Duan, C. He, D. Wang, X. Fan, and Y. Zhang, "mmplace: Robust place recognition with intermediate frequency signal of low-cost single-chip millimeter wave radar," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 4878–4885, 2024.

[8] Z. Hao, J. Xiao, Y. Wang, G. Wang, and F. Li, "mm-arnet: Exploring millimeter wave radar point clouds for human action recognition," *IEEE Transactions on Mobile Computing*, 2025.

[9] E. Sie, Z. Liu, and D. Vasisht, "Batmobility: Towards flying without seeing for autonomous drones," in *Proceedings of ACM MobiCom*, 2023, pp. 1–16.

[10] K. Cui, Q. Yang, L. Shen, Y. Zheng, F. Xiao, and J. Han, "Towards isac-empowered mmwave radars by capturing modulated vibrations," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 13 787–13 803, 2024.

[11] C. Meng, C. He, D. Wang, Y. Xiao, L. Wang, X. Fan, L. Zhang, and Y. Zhang, "Gr-fall: A fall detection system with gait recognition for indoor environments using siso mmwave radar," *Proc. ACM IMWUT*, vol. 9, no. 3, pp. 1–26, 2025.

[12] X. Liu, H. Liu, Y. Han, X. Xie, X. Tong, and K. Li, "Mhtrack: mmwave-based mobile hand tracking," *IEEE Transactions on Mobile Computing*, 2025.

[13] 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," in *Proc. IEEE/CVF CVPR*, 2020.

[14] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding wifi localization," in *Proc. ACM MobiSys*, 2013, pp. 249–262.

[15] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse, "Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver," in *Proc. ACM MobiSys*, 2018, pp. 376–388.

[16] K. Jiokeng, G. Jakllari, A. Tchana, and A.-L. Beylot, "When ftm discovered music: Accurate wifi-based ranging in the presence of multipath," in *Proc. IEEE INFOCOM*, 2020, pp. 1857–1866.

[17] J. Wang and D. Katabi, "Dude, where's my card? rfid positioning that works with multipath and non-line of sight," in *Proc. ACM SIGCOMM*, 2013, pp. 51–62.

[18] Z. Wang, M. Xu, N. Ye, R. Wang, and H. Huang, "Rf-focus: Computer vision-assisted region-of-interest rfid tag recognition and localization in multipath-prevalent environments," *Proc. ACM IMWUT*, vol. 3, no. 1, p. 29, 2019.

[19] G. Wang, C. Qian, K. Cui, X. Shi, H. Ding, W. Xi, J. Zhao, and J. Han, "A universal method to combat multipaths for rfid sensing," in *Proc. IEEE INFOCOM*, 2020, pp. 277–286.

[20] G. L. Turin, F. D. Clapp, T. L. Johnston, S. B. Fine, and D. Lavry, "A statistical model of urban multipath propagation," *IEEE Trans. Veh. Technol.*, vol. 21, no. 1, pp. 1–9, 1972.

[21] H. Liu, Y. Wang, A. Zhou, H. He, W. Wang, K. Wang, P. Pan, Y. Lu, L. Liu, and H. Ma, "Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 4, p. 140, 2020.

[22] C. He, R. Xia, C. Meng, X. Fan, D. Wang, H. Ren, J. Ji, and Y. Zhang, "Ghost points matter: Far-range vehicle detection with a single mmwave radar in tunnel," in *Proceedings of ACM MobiCom*, 2025, pp. 650–666.

[23] R. Schneider, D. Didascalou, and W. Wiesbeck, "Impact of road surfaces on millimeter-wave propagation," *IEEE Trans. Veh. Technol.*, vol. 49, no. 4, pp. 1314–1320, 2000.

[24] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, "Dbscan: Past, present and future," in *Proc. ICADIWT*, 2014, pp. 232–238.

[25] W. Min, M. Fan, X. Guo, and Q. Han, "A new approach to track multiple vehicles with the combination of robust detection and two classifiers," *IEEE T-ITS*, vol. 19, no. 1, pp. 174–186, 2017.

[26] A. Palffy, J. F. Kooij, and D. M. Gavrila, "Occlusion aware sensor fusion for early crossing pedestrian detection," in *Proc. IEEE IV*, 2019, pp. 1768–1774.

[27] E. Ohn-Bar, S. Sivaraman, and M. Trivedi, "Partially occluded vehicle recognition and tracking in 3d," in *Proc. IEEE IV*, 2013, pp. 1350–1355.

[28] S. Gilroy, E. Jones, and M. Glavin, "Overcoming occlusion in the automotive environment—a review," *IEEE T-ITS*, vol. 22, no. 1, pp. 23–35, 2019.

[29] T. Woodford, X. Zhang, E. Chai, and K. Sundaresan, "Mosaic: Leveraging diverse reflector geometries for omnidirectional around-corner automotive radar," in *Proc. ACM MobiSys*, 2022, pp. 155–167.

[30] L. Dodds, H. Shanbhag, J. Guan, S. Gupta, and H. Hassanieh, "Around the corner mmwave imaging in practical environments," in *Proc. ACM MobiCom*, 2024, pp. 953–967.

[31] T. Woodford, K. Qian, and X. Zhang, "Metasight: High-resolution nlos radar with efficient metasurface encoding," in *Proc. ACM SenSys*, 2023, pp. 308–321.

[32] N. Mehrotra, D. Pandey, A. Prabhakara, Y. Liu, S. Kumar, and A. Sabharwal, "Hydra: Exploiting multi-bounce scattering for beyond-field-of-view mmwave radar," in *Proc. ACM MobiCom*, 2024, pp. 1545–1559.

[33] S. Guo, Q. Zhao, G. Cui, S. Li, L. Kong, and X. Yang, "Behind corner targets location using small aperture millimeter wave radar in nlos urban environment," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 460–470, 2020.

[34] S. Yue, H. He, P. Cao, K. Zha, M. Koizumi, and D. Katabi, "Cornerradar: Rf-based indoor localization around corners," *Proc. ACM IMWUT*, vol. 6, no. 1, pp. 1–24, 2022.

[35] N. Scheiner, F. Kraus, F. Wei, B. Phan, F. Mannan, N. Appenrodt, W. Ritter, J. Dickmann, K. Dietmayer, B. Sick *et al.*, "Seeing around street corners: Non-line-of-sight detection and tracking in-the-wild using doppler radar," in *Proc. IEEE/CVF CVPR*, 2020, pp. 2068–2077.

[36] R. Prophet, J. Martinez, J.-C. F. Michel, R. Ebelt, I. Weber, and M. Vossiek, "Instantaneous ghost detection identification in automotive scenarios," in *Proc. IEEE Radar Conf.*, 2019, pp. 1–6.

[37] J. Kopp, D. Kellner, A. Piroli, and K. Dietmayer, "Fast rule-based clutter detection in automotive radar data," in *Proc. IEEE ITSC*, 2021, pp. 3010–3017.

[38] D. Wang, X. Zhang, K. Wang, L. Wang, X. Fan, and Y. Zhang, "Rdgait: A mmwave based gait user recognition system for complex indoor environments using single-chip radar," *Proc. ACM IMWUT*, vol. 8, no. 3, pp. 1–31, 2024.

[39] F. Kraus, N. Scheiner, W. Ritter, and K. Dietmayer, "Using machine learning to detect ghost images in automotive radar," in *Proc. IEEE ITSC*, 2020, pp. 1–7.

[40] T. Griebel, D. Authaler, M. Horn, M. Henning, M. Buchholz, and K. Dietmayer, "Anomaly detection in radar data using pointnets," in *Proc. IEEE ITSC*, 2021, pp. 2667–2673.

[41] M. Zeller, V. S. Sandhu, B. Mersch, J. Behley, M. Heidingsfeld, and C. Stachniss, "Radar instance transformer: Reliable moving instance segmentation in sparse radar point clouds," *IEEE Trans. Robot.*, vol. 40, pp. 2357–2372, 2023.

[42] C. Lovescu and S. Rao, "The fundamentals of millimeter wave radar sensors," *TI White Paper*, 2015.

[43] Z. Xu, C. Fan, and X. Huang, "Mimo radar waveform design for multipath exploitation," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5359–5371, 2021.

[44] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *NeurIPS*, vol. 30, 2017.

[45] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of CVPR*, 2017, pp. 652–660.

[46] TI, "Awr1642 single-chip automotive radar sensor evaluation module," 2020.

[47] ——, "Iwr6843 intelligent mmwave sensor standard antenna plug-in module," 2020.

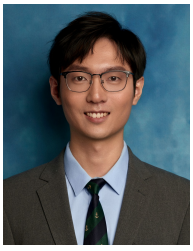
[48] ——, "Real-time data-capture adapter for radar sensing evaluation module," 2020.

[49] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *Proc. IEEE IROS*, 2020, pp. 10 359–10 366.

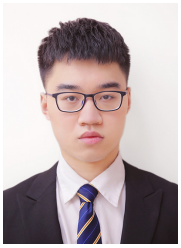
[50] H. Xiong and J. Hu, "Study on safe driving spacing control of co-directional vehicles on adjacent lanes in urban roads," *Advances in Transportation Studies*, 2019.

[51] C. He, C. Meng, C. He, X. Fan, B. Wang, Y. Yan, and Y. Zhang, "See through vehicles: Fully occluded vehicle detection with millimeter wave radar," in *Proceedings of ACM MobiCom*, 2024, pp. 740–754.

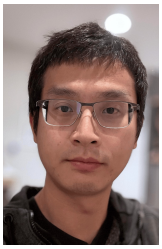
[52] Texas Instruments, "Programming chirp parameters in ti radar devices," Texas Instruments, Tech. Rep. SWRA553A, 2018. [Online]. Available: https://www.ti.com/lit/an/swra553a/swra553a.pdf



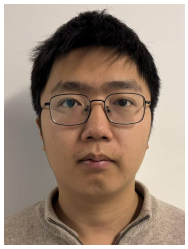
Chenming He received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2021, where he is currently pursuing the Ph.D. degree. His research interests include wireless sensing, autonomous driving, smart city, and embodied intelligence.



Chengzhen Meng received the B.S. degree from the School of Computer Science, Northwestern Polytechnical University. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, University of Science and Technology of China. His research interests include mmWave sensing.



Xiaoran Fan received the Ph.D. degree from Rutgers University in 2020. He is currently a Research Scientist at Google. He leads the Multimodal Sensing Foundation Research team. This team develops unified foundational models by integrating diverse inputs (like audio, radar, inertial, and video) to solve complex tasks in health, safety, home automation, and user identity. He is an experimentalist and system builder. He is enthusiastic about real-world-experiment-driven research.



Dequan Wang received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2021, where he is currently pursuing the Ph.D. degree. His research interests include wireless sensing, human–computer interaction and embodied intelligence.

Lu Zhang (Member, IEEE) received the Ph.D. degree from University of Science and Technology of China in 2022. She was a research assistant with Institute of Artificial Intelligence, Hefei Comprehensive National Science Center in 2022. Her current research interests include intelligent unmanned systems technology, multi-seneor calibration, and collaborative perception.

Yubo Yan (Member, IEEE) received the BS, MS, and PhD degrees in communication and information system from the PLA University of Science and Technology, China, in 2006, 2011 and 2017 respectively. He is now an associate professor with the University of Science and Technology of China. His current research interests include Internet of Things, wireless networks, intelligent sensing, and mobile computing.

Yanyong Zhang (Fellow, IEEE) received the B.S. degree from University of Science and Technology of China (USTC), Hefei, China, in 1997, and her Ph.D. degree from Penn State University, PA, USA, in 2002, both in Computer Science. In July 2002, she joined the Electrical and Computer Engineering Department and Winlab at Rutgers University as an Assistant Professor. She was promoted to an Associate Professor with tensure in 2008, and a Professor in 2015. Since July 2018, she has been with the School of Computer Science at USTC. Her current research interests include sensing, embodied intelligence, and cyber-physical systems.