Malicious Attacks against Multi-Sensor Fusion in Autonomous Driving

Yi Zhu[1], Chenglin Miao[2], Hongfei Xue[3], Yunnan Yu[1], Lu Su[4], Chunming Qiao[1]
[1] State University of New York at Buffalo, USA [2] Iowa State University, USA
[3] University of North Carolina at Charlotte, USA [4] Purdue University, USA
Email: [1] {yzhu39, yunnanyu, qiao}@buffalo.edu, [2] cmiao@iastate.edu, [3] hongfei.xue@charlotte.edu, [4] lusu@purdue.edu

ABSTRACT

Multi-sensor fusion has been widely used by autonomous vehicles (AVs) to integrate the perception results from different sensing modalities including LiDAR, camera and radar. Despite the rapid development of multi-sensor fusion systems in autonomous driving, their vulnerability to malicious attacks have not been well studied. Although some prior works have studied the attacks against the perception systems of AVs, they only consider a single sensing modality or a camera-LiDAR fusion system, which can not attack the sensor fusion system based on LiDAR, camera, and radar. To fill this research gap, in this paper, we present the first study on the vulnerability of multi-sensor fusion systems that employ LiDAR, camera, and radar. Specifically, we propose a novel attack method that can simultaneously attack all three types of sensing modalities using a single type of adversarial object. The adversarial object can be easily fabricated at low cost, and the proposed attack can be easily performed with high stealthiness and flexibility in practice. Extensive experiments based on a real-world AV testbed show that the proposed attack can continuously hide a target vehicle from the perception system of a victim AV using only two small adversarial objects.

CCS CONCEPTS

• **Security and privacy → Domain-specific security and privacy architectures**; • **Computer systems organization → Embedded and cyber-physical systems**.

ACM MobiCom '24, November 18-22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/09…$15.00
https://doi.org/10.1145/3636534.3649372

KEYWORDS

Autonomous driving; multi-sensor fusion; adversarial attack

ACM Reference Format:
Yi Zhu, Chenglin Miao, Hongfei Xue, Yunnan Yu, Lu Su, and Chunming Qiao. 2024. Malicious Attacks against Multi-Sensor Fusion in Autonomous Driving. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), September 30-October 4, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3636534.3649372

1 INTRODUCTION

Recent years have witnessed the proliferation of autonomous vehicles (AVs). One of the most fundamental functions of AVs is perception, which aims to understand surrounding driving environment using the equipped sensors. The perception systems in existing AVs typically employ sensor fusion, which integrates the perception results from different sensing modalities including LiDAR, camera and radar. Despite the rapid development of sensor fusion based perception systems, their vulnerability to malicious attacks have not been well studied.

Thus far, some studies have proposed to individually attack camera, LiDAR, and radar perception in AVs. For camera perception, the attacker can use some stickers [64] or paintings [77] to change the input image pixel values to fool the camera perception models. For LiDAR perception, a recent study [87] proposed to place some arbitrary objects (e.g., drones) at some specific locations to manipulate the LiDAR point cloud and fool the LiDAR perception models. For radar perception, some attack methods [15, 35, 47, 66] are proposed to actively transmit radar signals into the victim radar using some special devices. However, none of the above methods can attack all three sensor types simultaneously, which limits their effectiveness on the autonomous vehicles employing multi-sensor fusion systems. A straightforward attack against multi-sensor fusion systems is to simultaneously launch existing attacks on the camera, LiDAR, and radar sensors. However, this solution is not only practically challenging, but also inefficient in terms of both cost and stealthiness. For example, to simultaneously launch the existing camera attack [64], LiDAR attack [87] and radar

attack [47], the attacker needs to put some stickers on the target vehicle, and at the same time, control multiple drones to hover around some specific locations, as well as employ some special devices to transmit radar signals to the victim AV. Such a naively combined attack would suffer from not only significant attack effort and cost but also high risk of detection. More seriously, the window for targeting each sensing modality is limited, so finding the perfect opportunity to launch simultaneous attacks on all three sensors and synchronizing attack activities for long enough time to compromise the fusion system is almost mission impossible.

To fill this research gap, in this paper, we aim to investigate the possibility of using a single type of compositive adversarial objects to attack all three sensors through passive reflection. Developing such type of object, however, is not easy. It is possible to combine the existing attacks on camera [64] and LiDAR [87] by placing some objects with special color patterns around some specific locations in the driving environment. However, existing attacks on radar perception, which rely on special devices to actively transmit signals, can not be incorporated with the above attacks on camera and LiDAR, since such active attacks suffer from various practical challenges. For example, they require sub-nanosecond-level synchronization between the attacker's transmitter and the victim radar [15, 35], or require the transmitter to be placed at a fixed angle/distance to the victim radar [47, 66]. So in their experiments, the attacker has to synchronize his transmitter and the victim radar by connecting them using a wired link, or keep the radar and the transmitter stationary during the attack. Obviously, these active attacks on radar lack practicality and flexibility and thus can not be incorporated with the attacks on camera and LiDAR.

To address above challenges, we design a new attack method on radar perception by leveraging the characteristics of mmWave reflection on a smooth metal surface. By placing a smooth metal surface between the radar and a target vehicle with a specific orientation, the transmitted mmWave signals can be deflected from radar receiver, leading to a reduction in the energy of echo signals from the vehicle. When the energy becomes lowered than a threshold, the target vehicle will be hidden from radar perception. Following this idea, we design a compositive adversarial object by integrating the proposed radar attack with existing attacks on camera and LiDAR. As shown in Figure 1, the adversarial object is a piece of cardboard attached by a color patch and a metal foil. As discussed, the metal surface on the object can attack radar perception. Its color patch can manipulate the input image pixel values to attack camera perception. The object can reflect lasers to attack LiDAR perception. By choosing appropriate number, size, color pattern of the objects and placing them at specific locations with specific orientations, the attacker can simultaneously attack all three types of sensors, so that the final perception results of the sensor fusion systems are changed.

Figure 1: An attack example.

To launch the attack, the attacker can employ various object carriers such as drones or car advertisements. Figure 1 shows an example of using drones to launch the proposed attack. The victim AV drives on a road with a vehicle in front of it. The attacker aims to hide the front vehicle (referred to as **target vehicle**) from the sensor fusion based perception system of the victim AV. The attacker first generates the adversarial objects and their locations and orientations in an offline manner before the attack. To launch the attack, the attacker uses drones to carry the derived objects and makes them hover at some specific locations with specific orientations. This type of attack may cause collisions between the victim AV and the target vehicle. Since the objects do not require special materials or 3D printing techniques, they can be easily fabricated at low cost. Launching the attack is easy in practice, since the attacker only need to control the locations and orientations of the drones. In addition, since the drones only hover for a few seconds during the attack and can fly away from the victim AV immediately after the attack, the attack can be performed with high stealthiness and flexibility.

To achieve the attack goal using these objects, we need to maximize the attack effectiveness on hiding the target vehicle from the sensor fusion system after placing these objects in the driving environment. Intuitively, more and larger objects would benefit the attack in terms of its effectiveness. However, to reduce the attack cost and improve its stealthiness, we need to minimize the objects number and sizes. To achieve a trade-off between these conflicting goals, we characterize the objects with some parameters including the number, sizes, color pattern, orientations and locations of the objects, and formulate an optimization problem to optimize these parameters by considering both attack effectiveness as well as cost and stealthiness. However, directly solving this problem through gradient descent is challenging due to the non-differentiable objective function and constraints. To address these challenges, we propose an attack framework

which solves the optimization problem into two steps based on a series of novel heuristics.

We evaluate the proposed attack on both real-world autonomous vehicle testbed and simulators. Our experimental results show that the attacker can continuously attack the sensor fusion systems by using only two small objects. Our evaluations on Baidu Apollo demonstrate that the attack can cause potential collisions of the victim AV.

In addition, we propose a vulnerability analysis framework that can estimate how important role a sensor play in the fusion system. The proposed framework can not only identify vulnerable fusion systems that rely on a subset (or only one) of sensors but also provide guidance for designing more robust sensor fusion systems.

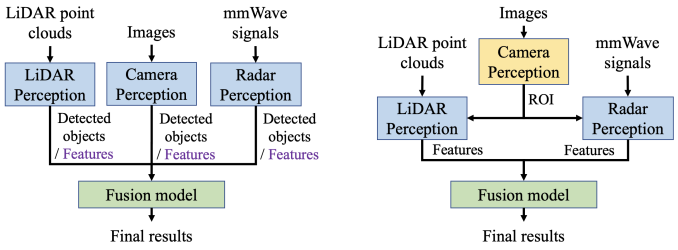
In summary, this paper has the following contributions:

- We propose the first study on the vulnerability of AV's multi-sensor fusion systems that employ LiDAR, camera, and radar. A novel compositive adversarial object is proposed to simultaneously attack all three sensors.
- We propose a novel passive-reflection-based attack on radar perception by leveraging the characteristics of mmWave reflection on a metal surface, which can be integrated with existing attacks on camera and LiDAR.
- We propose a vulnerability analysis framework that can not only identify vulnerable fusion systems that rely on a subset (or only one) of sensors but also provide guidance for the design of robust sensor fusion systems.

2 PRELIMINARIES

Single sensor perception. State-of-the-art perception systems of autonomous vehicles are equipped with LiDAR, camera, and radar. Cameras provide images which can be directly processed by deep neural networks (DNN) to generate bounding boxes of the detected objects [56]. LiDARs generate 3D point cloud that contains the 3D coordinates of the reflected points through laser scanning. In existing LiDAR perception, the point cloud is first divided into voxels/pillars [37, 45]. A feature map is then generated based on the features on each voxel/pillar, and DNNs are used to process the feature map.

Compared with camera and LiDAR, radar is more robust to severe weather and lighting conditions [26, 27, 51, 57, 61, 63]. In AVs, radars are often operated in millimeter Wave (mmWave) band, and they transmit the Frequency Modulated Continuous Wave (FMCW) signal, which is a kind of continuous wave whose frequency increases uniformly with time. Fast Fourier Transform (FFT) is used to process the obtained intermediate frequency (IF) signals. After applying FFT on the IF signals along the time domain, the output is $f(d) \approx A_r \delta(d - d^*)$, where $\delta()$ is the Dirac delta function and A_r is determined by the energy of the received signal [50].



(a) Decision-level / Feature-level **(b) Cascaded Fusion**

Figure 2: Pipeline of sensor fusion.

Thus, the detected object can be obtained by finding the peak in $f(d)$, and the peak value is determined by the energy of the echo signal from the object. Due to the noise, the detection results may contain many false positives. To eliminate those false positives, the peak whose values of $f()$ is smaller than a threshold will be removed. The threshold can not be too large otherwise the object would be hard to be detected, and it can not be too small otherwise it would cause too many false alarms. Thus, the threshold is often calculated by estimating the background noise power based on the Constant False Alarm Rate (CFAR) detection [60].

Sensor fusion based perception. Sensor fusion has been proved to be able to improve the perception accuracy and reliability in many applications [19, 22, 38, 40, 55, 55, 58, 59, 67–69, 80, 80], and has been widely adopted in the task of object detection in AVs [10, 41, 52–54]. Existing sensor fusion systems in AV for object detection can be categorized as decision-level, feature-level, and cascaded fusion. The general pipelines of these types of fusion systems in AV are shown in Figure 2. For decision-level fusion, the data from each sensor are processed by individual perception models to generate the perception results, i.e., detected objects. A fusion model is then used to aggregate the detected objects from each perception model and generate the final detection results. For feature-level fusion, the perception model of each sensor generates features instead of detected objects. For camera and LiDAR, the features are intermediate features learned by DNNs. For radar, the features are generated by projecting the detected objects (points) into BEV or image plane [49, 79]. The fusion model fuses the features and generate the final detection results using DNNs [34, 43, 49, 79]. For cascaded fusion, the data of one sensor is first processed by its perception model to generate the detected objects [46, 74], based on which Regions of Interest (ROI) are generated. Then, the perception models of other sensors process the sensory data and extract the features within the ROI. The DNNs in the fusion model process the features and generate the fused detection results. In existing perception models, the detected objects are filtered in the post-processing step by removing

the output objects whose detection confidences are smaller than a threshold.

3 ATTACK GOAL AND THREAT MODEL

Attack goal. This paper focuses on the scenario where the AV is equipped with a multi-sensor fusion system as its object detection system. Specifically, we assume that the victim AV drives on a road and there is a vehicle (the target vehicle) in front of it . The goal of the attack is to continuously hide the target vehicle from the multi-sensor fusion system, i.e., make the victim AV not able to detect the target vehicle in its collected sensory data frames as it drives towards the target vehicle. The attacker can select the target scenes (roads) to launch the attack. The target vehicle could be any random vehicle on the road or one owned by the attacker. For example, the attacker can intentionally park a car on a selected road and launch the attack to let the victim AV collides with it. The attacker may have many types of motivations to launch this attack, such as causing traffic accidents for insurance frauds, unfair competition between autonomous driving companies, or hurting the drivers and passengers in the victim AV or the target vehicle.

Threat model. We consider a practical and challenging setting where the attacker can not obtain the original sensory data collected by the victim AV. But he can obtain surrogate sensory data on the selected scene by using simulators or collecting them through the same models of sensors. Besides, we consider a white-box setting and assume that the attacker has the full knowledge of the victim sensor fusion system, which is widely adopted in existing attack methods [14, 70, 77]. This assumption is reasonable because some autonomous driving companies launch open-source autonomous driving platforms [2, 3]. The attacker can also purchase the same model of AV as the victim AV and obtain such information through reverse engineering. In addition, we assume the attacker can intentionally select a specific scenario such as a particular road segment, background environment, weather as well as target vehicle. As long as the attack succeeds in one selected scenario within a short time period, the attack goal can be achieved (e.g., causing traffic accidents to raise safety concerns to a specific model of autonomous vehicles, in order to defame the autonomous driving company for unfair competition).

4 ATTACK DESIGN

While some methods have been developed to individually attack camera, LiDAR, and radar sensors, none of them can attack all three sensor types simultaneously to attack multi-sensor fusion systems in AVs. A straightforward solution is to simultaneously launch existing attacks on the camera, LiDAR, and radar. However, such a naively combined attack would suffer from not only significant attack effort and cost but also high risk of detection. More seriously, finding the perfect opportunity to launch simultaneous attacks and synchronizing attack activities for long enough time is almost mission impossible.

In this paper, we aim to investigate the possibility of using a single type of compositive adversarial objects to attack all three sensors through passive reflection, which can reduce the attack cost and effort as well as improve the attack stealthiness. Developing such type of object is not easy. It is possible to combine the attacks on camera [64] and LiDAR [87] by placing some objects with special color patterns around some specific locations in the environment. However, existing attacks on radar, which use some special devices to actively transmit signals, can not be incorporated with the above attacks. This is because they suffer from various practical challenges, such as requiring sub-nanosecond-level synchronization between the attacker's devices and the victim radar [15, 35], or requiring the devices to be placed at a fixed angle/distance to the victim radar [47, 66]. So in their experiments, the attacker needs to connect their devices to the victim radar using a wired link, or keep the radar and the devices stationary.

To address this challenge, we leverage the characteristics of mmWave reflection on metal surfaces: *(1) metal surfaces are strong reflectors and the mmWave signals can barely penetrate them; (2) the reflection on a smooth metal surface is almost specular.* If we place a metal surface between the radar and the target vehicle, and change the orientation of the surface, the energy of the echo signals from the vehicle can be reduced and thus affect the radar perception results. As shown in Figure 3, without the metal surface, the transmitted signals are reflected by the vehicle and then received by the radar, which makes the radar detect the vehicle. After placing the metal surface between the radar and the vehicle at some specific locations, part of the transmitted signals are blocked by the surface since the mmWave signals can not penetrate it. And by adjusting the orientation of the metal surface, the signals blocked by the surface are reflected away, so that the energy of signals received by the radar are reduced. According to Section 2, the small energy of the echo signals result in a small peak value in $f(d)$. By adjusting the location, size, and orientation of the surface, the attack can make the peak value smaller than the CFAR threshold to cause a missing radar detection of the vehicle. Please note that completely blocking the whole vehicle is not necessary for attacking radar perception. As long as the reflected signal strength of the vehicle is smaller than the detection threshold, the vehicle can be hidden. Moreover, the detection threshold cannot be too small. This is because the real-world driving environments are not empty spaces, and there could be noise reflections generated by the surrounding objects

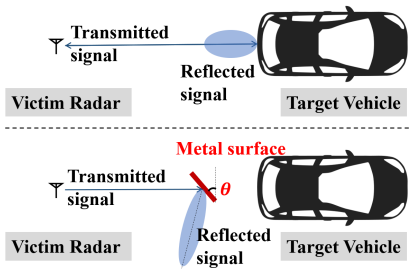


Figure 3: Attack radar using a metal surface.

such as trees, buildings, and lamp posts. Using smaller detection thresholds would result in many false alarms, causing vehicle freezing or frequent braking behaviors.

Adversarial object design. We combine the above attack vectors and design a new type of *compositive adversarial objects*. As shown in Figure 1, the adversarial object is a piece of cardboard covered by a metal foil and a color patch. As discussed above, the metal foil on the object can be used to attack radar perception. In addition, the adversarial object can reflect laser signals to create some malicious point clusters or block existing points, which can change the input LiDAR point cloud and fool the DNNs in LiDAR perception model. Furthermore, the specially designed color patch on the object can change the pixel values of camera images to fool the DNNs in camera perception models. By strategically designing the sizes, orientations, and color pattern of the objects, and placing appropriate number of these objects in some specific physical locations, the attacker may simultaneously attack all three types of sensors, and change the final perception results of the multi-sensor fusion system.

Attack pipeline. To achieve the attack goal using the proposed adversarial object, the attack pipeline works as follows: Before the attack, the attacker first selects a target scene where he intends to launch the attack, and then simulates various possible driving conditions (e.g., relative positions of the victim AV and target vehicles as the AV drives towards the target) in the selected scene. To hide the target vehicle from the multi-sensor fusion system, the attacker generate a specific number of adversarial objects with specific sizes, color pattern, locations and orientations in an offline manner based on the selected scene and simulated driving conditions. To launch the attack, the attacker can use drones to carry the derived objects, and make them hover around the derived locations with the derived orientations in the selected scene.

5 ATTACK METHODOLOGY

To achieve the attack goal, we need to maximize the attack effectiveness on hiding the target vehicle from the sensor fusion system after placing these objects in the driving environment. Intuitively, more and larger objects would benefit the attack in terms of its effectiveness. However, to reduce the attack cost and improve its stealthiness, we need to minimize the objects number and sizes. To achieve a trade-off between these conflicting goals, we characterize the objects with some parameters including the number, sizes, color pattern, orientations and locations of the objects, and formulate an optimization problem to optimize these parameters by considering both attack effectiveness as well as cost and stealthiness.

5.1 Problem Formulation

We use $P_s = \{P_{s,n} | n = 1, 2, ..., N\}$ to denote the sizes of the objects, where $P_{s,n} = \{w_{s,n}, h_{s,n}\}$ denotes the width $w_{s,n}$ and height $h_{s,n}$ of the n-th adversarial object. We use $P_l = \{P_{l,n} | n = 1, 2, ..., N\}$ to denote the locations of the adversarial objects. $P_{l,n} = \{x_n, y_n, z_n\}$ denotes the xyz-coordinates of the n-th object. The orientations of the adversarial objects are denoted as $P_o = \{P_{o,n} | n = 1, 2, ..., N\}$, where $P_{o,n}$ is the yaw angle (θ in Figure 3) of the n-th object. The color pattern of the adversarial objects are denoted as $P_c = \{P_{c,n} | n = 1, 2, ..., N\}$, where $P_{c,n}$ is the set that contains the RGB values of the cover (patch) on the n-th object. Then we formulate the problem of deriving the adversarial objects as the following optimization problem:

$$\begin{aligned} \min_{N,P} \quad & M(X_{camera}, X_{lidar}, X_{radar}) + \alpha N + \beta L_{area} \\ \text{s.t.} \quad & X_{camera} = T_{camera}(N, P_s, P_l, P_o, P_c), \\ & X_{lidar} = T_{lidar}(N, P_s, P_l, P_o), \\ & X_{radar} = T_{radar}(N, P_s, P_l, P_o), \end{aligned} \tag{1}$$

where $P = \{P_s, P_l, P_o, P_c\}$ and $M(X_{camera}, X_{lidar}, X_{radar})$ represents the attack utility, which is modeled as the output detection confidence of the sensor fusion model M for the target vehicle given the input image X_{camera}, LiDAR point cloud X_{lidar}, and mmWave signals X_{radar}. T_{camera}, T_{lidar}, and T_{radar} are the functions that model the camera's image data, LiDAR point cloud, and radar signals, respectively, given the parameters of the adversarial objects, i.e., $\{N, P_s, P_l, P_o, P_c\}$. L_{area} is the total area of the N adversarial objects. α and β are used to adjust the trade-off between the three terms in the objective function, which considers both the attack effectiveness and attack stealthiness.

To approximate $T_{camera}()$ and $T_{lidar}()$, we use the rendering function proposed in [33] and the ray-casting method in [1] to obtain the image pixel values and point clusters generated by the adversarial objects, respectively. These image pixel values and point clusters are injected into the surrogate camera images and LiDAR point cloud without the attacks, which can be collected by the attacker in the selected scenes or can be obtained through simulations [21]. For $T_{radar}()$, we adopt the signal simulation method in [78]. It divides the surfaces of an object into many small triangles. The mmWave signals reflected from each small triangle are modeled using

the method in [36]. The final received IF signals can be obtained by summing up the mmWave signals reflected from each triangle. The mesh of the target object can be obtained through open-sourced 3D mesh library, mesh generation models [31, 76, 83], or manual modeling.

To better illustrate the attack framework, here we first consider the decision-level sensor fusion system, since it is the same type of fusion used in existing autonomous driving platforms such as Autoware [2] and Baidu Apollo [3]. **In Section 5.5, we will discuss how to extend our proposed framework to other types of fusion systems, including feature-level and cascaded fusion.** In decision-level fusion, the algorithm of fusing different detection results is not directly differentiable. Thus, we decompose the fusion function $M(X_{camera}, X_{lidar}, X_{radar})$ and propose to simultaneously minimize the detection confidences outputted by the camera, LiDAR and radar perception models, i.e., minimizing $M_{camera}(X_{camera}) + M_{lidar}(X_{lidar}) + M_{radar}(X_{radar})$. The intuition is that, according to Figure 2, if the object does not appear in the detection results of any individual perception models, it will obviously not appear in the outputs of the fusion model. For radar perception, its detection confidence can be measured by the difference between the CFAR threshold f_t and the value of the FFT output $f()$ at the object's groundtruth location d^*, i.e., $f(d^*) - f_t$. We then normalize it using the function $\frac{1}{1+exp(f_t - f(d^*))}$.

Solving the above optimization problem is challenging. The constraints in Eq (1) are non-differentiable since N is discrete. The objective function is also non-differentiable, since the point cloud pre-processing step (dividing point cloud into voxels or pillars) in $M_{lidar}(X_{lidar})$ is non-differentiable [37]. The non-differentiable constraints and objective function make it difficult to directly solve this problem using gradient based methods.

Solution overview. To address the above challenges and solve the optimization problem, we propose an attack framework involving a series of novel heuristics, based on our studies on the characteristics of the above attack parameters. In our proposed attack framework, we decompose the optimization problem to first determine the object locations P_l and then update the other parameters. The intuition behind the idea is that, **according to our investigations on the impact of attack parameters, which are detailed in Section 5.2**, *the LiDAR perception model M_{lidar} is mainly affected by P_l and is barely affected by other parameters.* Thus, if we can determine the values of P_l to minimize $M_{lidar}(X_{lidar})$ in the first step of the optimization framework, we can remove the non-differentiable function $M_{lidar}(X_{lidar})$ when updating other parameters. Besides LiDAR perception $M_{lidar}(X_{lidar})$, we have to consider camera and radar perception when determining P_l. For camera and radar perception models, we find that *there are some location sets (some values of P_l), where manipulating the sizes P_s, orientations P_o and color pattern P_c of the objects can have a high probability of affecting the camera and radar perception results.* These location sets are referred to as *vulnerable location sets*. In contrast, the objects at other locations always have little impact on the perception results no matter how we change their values of P_s, P_o and P_c. Thus, if we can find the vulnerable location set in the first step, it would be easier to minimize $M_{radar}(X_{radar}) + M_{camera}(X_{camera})$ when updating P_s, P_o and P_c in the following steps. Based on the above observations, we aim to find the optimal location for each adversarial object P_l^* in the first step of our solution that not only minimizes $M_{lidar}(X_{lidar})$ but also belongs to the vulnerable location sets (having a high probability of affecting camera and radar perception results after changing P_s, P_o and P_c). This step is referred to as Location Probing. **Details on how to find P_l^* will be discussed in Section 5.3. After the location probing step, in Section 5.4, we propose to alternatively update the remaining parameters $\{P_o, P_c, N, P_s\}$, until some convergence criterion is satisfied.** Specifically, we first fix $\{N, P_s\}$ and update $\{P_c, P_o\}$. Since $M_{lidar}(X_{lidar})$ can be removed according to the above discussion and N is fixed, both the objective function and constraints are differentiable. Thus, gradient descent algorithm can be used to update $\{P_c, P_o\}$. When updating $\{N, P_s\}$ and fixing $\{P_o, P_c\}$, we propose to remove the redundant objects or parts of the objects that have little contribution to the success of the attack. This can reduce the value of $\alpha N + \beta L_{area}$ without hurting the perception results significantly.

5.2 Characteristics of Attack Parameters

In this section, we study the characteristics of different attack parameters, i.e., the impact of different attack parameters on different sensing modalities.

We select YOLO-v3 [56] as the camera perception model and PointPillars [37] as the LiDAR perception model, which are state-of-the-art models used in Autoware [2] and Baidu Apollo [3]. The radar perception model is the same as that in Section 2, which is commonly adopted by existing AVs. To evaluate the impact of object locations P_l, we propose to study the changes of detection confidences after manipulating the value of P_l while fixing the values of other parameters. Specifically, given a set of $\{N, P_s, P_o\}$, we randomly change the object locations P_l multiple times and optimize the color pattern of the objects to minimize the detection confidence of camera perception model. For each sensing modality (camera, LiDAR and radar), we record the difference between the maximum and minimum values of the detection confidence under different values of P_l, referred to as I_{camera}, I_{lidar}, and

I_{radar}, respectively. $\{I_{camera}, I_{lidar}, I_{radar}\}$ can be used to measure how much impact changing the object locations P_l can cause to the outputs of camera, LiDAR and radar perception, respectively. To obtain more reliable measurement, we select 100 scenes from the KITTI dataset, and repeat the above procedure multiple times with random values of $\{N, P_s, P_o\}$. The object number N ranges between 1 and 4. The object size P_s ranges between $0.3m$ and $1.0m$. The object orientation P_o ranges between 0 and $\pi/2$. After multiple runs, we calculate the average values of $\{I_{camera}, I_{lidar}, I_{radar}\}$ for each sensing modality, referred to as $\{\bar{I}_{camera}, \bar{I}_{lidar}, \bar{I}_{radar}\}$. We also use similar methods to evaluate the impact of object size P_s, and orientation P_o. Table 1 summarizes the impact ($\{\bar{I}_{camera}, \bar{I}_{lidar}, \bar{I}_{radar}\}$) of different parameters on the three sensing modalities. We do not calculate $\bar{I}_{camera}$ for color pattern P_c since it has been proved to have large impact on camera perception [77] and have no effect on the LiDAR point cloud and radar signals.

Table 1: Impact of different attack parameters

Parameters	Location P_l	Size P_s	Orientation P_o	Color pattern P_c
LiDAR	**High** (0.81)	Low (0.22)	Low (0.15)	None
Camera	**High** (0.94)	Mid (0.48)	Low (0.12)	**High**
Radar	**High** (0.93)	Low (0.29)	**High** (0.65)	None

From Table 1, we can see that object locations P_l have large impact on the detection confidences for all the sensing modalities. The LiDAR perception is mainly affected by P_l, and it is barely affected by other parameters. This is because existing LiDAR perception models learn the features based on the locations of LiDAR points. Changing the size/orientation of an object has much smaller effect on the locations of its LiDAR points, compared with changing the location of that object. For radar and camera perception, besides P_l, the orientations P_o and the color pattern P_c also have large impact on the outputs, respectively. In addition, for some sets of locations P_l, it is highly possible to change the outputs of camera and radar perception by manipulating the values of object size P_s, orientation P_o, and color pattern P_c. In contrast, for other values of P_l, the perception results can not be significantly affected no matter how we change the values of P_s, P_o and P_c.

5.3 Location Probing

Based on the investigation in the previous section, we aims to find the object locations P_l^* that belong to the vulnerable location sets and can minimize $M_{lidar}(X_{lidar})$. To help find vulnerable location sets, for both camera and radar perception, we define the adversarial score on a given location to measure how likely an object at this location can cause significant impact on the outputs of camera and radar perception models, after manipulating the object sizes, orientations and color patterns.

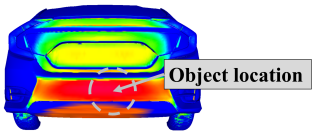


Figure 4: Radar energy heatmap.

Adversarial scores. As discussed in Section 4, the radar perception can be attacked because the radar echo signals are blocked by the metal surface. Thus, to calculate adversarial scores of a given location for radar perception, we first generate the radar heatmap through the simulation method in [78], by calculating the energy of the echo signals from different parts of a vehicle. Figure 4 shows an example of radar energy heatmap, where red color indicates high energy of echo signals from this part. We can see that different parts of the target vehicle generate different energy levels of signals due to their different geometric shape and materials. And some parts of the vehicle significantly contribute to the total received energy. Blocking these areas can significantly reduce the received energy of radar signals, making the radar perception model fail to detect the target vehicle. Given this intuition, we define the adversarial score $S_r(l)$ of a given location l as the summation of pixel values around this location in the heatmap, i.e., $S_r(l) = \sum_{k \in K} H(k)$, where $H(k)$ is the value of pixel k in the heatmap and K is the set of pixels that are within a predefined range around location l, as shown in Figure 4. Larger value of S_r means the object at this location can block larger energy of echo signals, thus has larger potential to affect radar perception results. For camera perception, the gradient of a pixel can be used to measure the potential impact of changing this pixel value on the model's output. According to [9], larger gradient of a pixel means changing its value can have larger impact on the camera perception results. Thus, we define adversarial score $S_c(l)$ of a given location l for camera perception as the summation of gradients of the pixels within a predefined range around this location in image plane, i.e., $S_c(l) = \sum_{k \in K} |G(k)|$, where K is the set of pixels that are within a predefined range around location l in the input image and $G(k)$ is the gradient of pixel k.

Location update. Based on the above definitions, larger adversarial scores S_c and S_r of each object location in P_l means P_l is more likely to belong to vulnerable location sets. Thus, to find the optimal object locations P_l^*, we update P_l to simultaneously minimize the output confidence of LiDAR perception model $M_{lidar}(X_{lidar})$, and maximize the summation of adversarial scores S_c and S_r of each location $P_{l,n}$:

$$\begin{aligned} \min_{P_l} \quad & M_{lidar}(X_{lidar}) - \eta_c \sum_{P_{l,n} \in P_l} S_c(P_{l,n}) - \eta_r \sum_{P_{l,n} \in P_l} S_r(P_{l,n}) \\ \text{s.t.} \quad & X_{lidar} = T_{lidar}(N^0, P_s^0, P_l, P_o^0), \end{aligned} \tag{2}$$

where η_c and η_r are used to balance the later two terms. The larger value of and $\{N^0, P_s^0, P_o^0\}$ are the initial values of the objects number, sizes and orientations, respectively. Evolutionary algorithm is used to solve this problem.

5.4 Parameters Updating

After identifying the optimal object location P_l^*, we next aim to update the remaining parameters, so that we can derive the adversarial objects to achieve the attack goal. To solve this problem, we propose to alternatively update the remaining parameters $\{P_o, P_c, P_s, N\}$ until the convergence criterion is satisfied. When updating $\{P_o, P_c\}$ and fixing $\{N, P_s\}$, we can remove the $M_{lidar}(X_{lidar})$ and the corresponding constraint since $\{P_o, P_c\}$ has little impact on LiDAR perception according to Table 1, so that the objective function becomes differentiable. In addition, since N is fixed, the constraints are also differentiable. Thus, gradient descent algorithm can be used to update $\{P_o, P_c\}$.

The initial object number N^0 and size P_s^0 of the objects are usually predefined as relatively large values, which can be larger than needed, i.e., some objects or parts of the objects may not be necessary for achieving the attack goal. Thus, to minimize $\alpha N + \beta L_{area}$, we propose to remove these redundant objects or parts of the objects, which have little contribution to the success of the attack. This can minimize $\alpha N + \beta L_{area}$ without hurting the value of $M_{lidar}(X_{lidar}) + M_{radar}(X_{radar}) + M_{camera}(X_{camera})$ significantly. We first divide each object into $0.025m * 0.025m$ grids, and define a set of importance scores $\{w_d^l, w_d^c, w_d^r\}$ for each grid d to measure its contribution on attacking LiDAR, camera, and radar, respectively. For camera and radar perception, the importance score of each grid can be measured using the a similar method as that in Section 5.3. Specifically, the importance score w_d^c of grid d for camera can be measured by the summation of gradients of the pixels within the grid in image plane. The importance score w_d^r of grid d for radar can be measured by the summation of values of pixels that are within the grid d in the radar heatmap. For LiDAR perception, to measure the importance score of each grid, we adopt the idea in [87] and randomly remove some grids to calculate the change of output detection confidence in LiDAR perception model for L iterations. The importance score $\bar{w}_d^l$ of grid d is defined as the average value of $\{M_{lidar}(X_{lidar}) - M_{lidar}(X_{lidar} - X(D^i)) | d \in D^i, i = 1, 2, ..., L\}$, where $X_{lidar} - X(D^i)$ is the LiDAR point clouds after removing the points $X(D^i)$ generate by the grids D^i. In addition, we also remove each object n and calculate $\hat{w}_d^l = M_{lidar}(X_{lidar}) - M_{lidar}(X_{lidar} - X(n))$, where $X_{lidar} - X(n)$ is the LiDAR point clouds after removing the object n and d belongs to object n. The importance score of each grid on LiDAR d is defined as $w_d^l = \bar{w}_d^l * \hat{w}_d^l$. The final importance score w_d of each grid d is the summation of the scores on each sensor after applying Softmax operations. The grids that have smaller value of w_d have smaller contribution on the attack. Based on the calculated w_d values, we remove the grids whose w_d values are smaller than a threshold. The remaining grids are clustered using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to generate the new adversarial objects. The number of the updated objects after this step is the number of the remaining clusters. The sizes of the updated adversarial objects are the sizes (height and width) of the remaining clusters.

5.5 Generalizability and Robustness

Attack other types of fusion. Besides decision-level fusion, the proposed attack framework can be easily adapted to attack other types of sensor fusion systems, including feature-level and cascaded fusion. According to Figure 2, in feature-level and cascaded fusion, the outputs of individual perception models do not contain the detection confidence, and only the fusion models can output detection confidences. Thus, to generate the adversarial objects that can be used to attack these types of fusion systems, we replace detection confidences $M_{lidar}()$, $M_{camera}()$, and $M_{radar}()$ of individual perception models in Eq (1) and Eq (2) with the final detection confidence $M()$ of the fusion model.

Continuous and robust attack. To achieve continuous and robust attack when the victim AV drives towards the target vehicle, the attacker first simulates various possible conditions in the selected scene/road such as various approaching distances between the two vehicles. The attacker then generates the parameters of the adversarial objects by summing the objective values in Eq. (1) for all the simulated conditions. To improve the attack robustness against the imprecise placement of adversarial objects caused by positioning errors of drones, random perturbations on the locations and orientations of the adversarial objects are added during the optimization process. In this way, the target vehicle can be continuously hidden under various driving conditions, even when the objects are not placed with high precision.

6 REAL-WORLD EXPERIMENTS

6.1 Experimental Setup

Real-world testbed. To evaluate the proposed attacks in the physical world, we use a Lincoln MKZ (shown in Figure 5) as the autonomous vehicle testbed. A Velodyne VLP-32C LiDAR, an Allied Vision Mako G-319 camera, and a TI AWR1843 radar are mounted on the AV testbed. The radar is operated in 77GHz, which is the same as that in existing AVs [3]. As shown in Figure 5, the victim AV drives towards the target vehicle (black Honda sedan) when we evaluate the proposed attacks.

Figure 5: The real-world testbed.

Model and attack settings. We consider a victim multi-sensor fusion system that is equipped with three types of sensors (LiDAR, camera, and radar) and adopts the decision-level fusion, which is commonly used in existing autonomous driving platforms such as Autoware [2] and Baidu Apollo [3]. Specifically, the camera perception model is YOLO-v3, and the LiDAR perception model is PointPillars, which are state-of-the-art perception models used in Autoware and Baidu Apollo. The radar perception model is the same as that described in Section 2, which is commonly adopted in existing autonomous driving systems. The goal of the attacker is to hide the target vehicle from the multi-sensor fusion system as the victim AV drives forwards. Specifically, the attacker aims to simultaneously hide the target vehicle from the perception models of LiDAR, camera and radar, so that the fusion system can not detect the target vehicle no matter what fusion algorithm it uses.

According to the attack pipeline discussed in Section 4, we simulate various driving conditions, based on which we obtain the surrogate LiDAR and camera data and simulate the 3D mesh of the target vehicle. The function $T_{lidar}()$, $T_{camera}()$ and $T_{radar}()$ are built upon these surrogate data and 3D meshes. For $T_{camera}()$, we also consider the color distortion of the printers and cameras in the physical world using the method in [77]. In addition, the initial object number N^0 is set to 3, and the initial size P_s^0 of each object is set to $0.5m$ (both initial width and height of the object are set to $0.5m$). Please note that N^0 and P_s^0 are hyper parameters for the proposed attack algorithm.

Evaluation metrics. We measure the detection *Recall* of LiDAR, camera, and radar perception models (i.e., the percentage of the sensory data frames where the target vehicle is successfully detected by the individual sensor) before and after attacks. We also measure the final detection Recall after fusion. Specifically, we consider a strict criteria where the vehicle is detected by the sensor fusion system if it is detected by any perception models. We define the *Attack Success Rate* (ASR) as the percentage of the sensory data frames where the target vehicle is not detected by any perception models. Here, the sensory data frames contain the LiDAR point cloud, camera image, and mmWave signals collected at each timestamp. Finally, we evaluate the consequences of the attack on a state-of-the-art autonomous driving platform to demonstrate the safety threats caused by the proposed attacks.

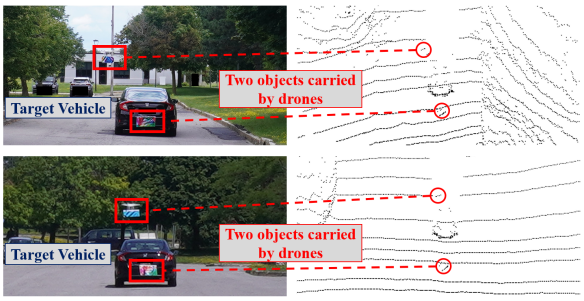


Figure 6: Attacks in the physical world.

(a) Without attack **(b) With attack**

Figure 7: Attack consequence.

6.2 Attack Performance and Consequence

Attack performance. In our experiments, we consider two scenes, which are shown in the first column of Figure 6. Before the attack, we drive the victim AV towards the target vehicle from $30m$ to $5m$ and collect a sequence of data frames. Then we repeat this process multiple times in each scene. Overall, we collected 343 frames, and the detection Recalls for LiDAR, camera, and radar perception are 0.97, 0.99, and 1.00, respectively. We generate the adversarial objects using the proposed attack algorithm. The total area of the two objects in the first scene is $0.25m^2$. The $\{w, h\}$ (width and height) of the two objects is $\{0.5m, 0.3m\}$ and $\{0.4m, 0.25m\}$, respectively. The total area of objects in the second scene is $0.26m^2$. The {width, height} of two objects are $\{0.5m, 0.3m\}$ and $\{0.4m, 0.275m\}$, respectively. Two drones (a DJI Phantom 4 Pro and a DJI Mavic Pro) are used to carry the two objects to hover around the derived locations with the derived orientations. The second column of Figure 6 shows the locations of the derived adversarial objects in BEV of LiDAR point clouds. During the attack, we collect 351 frames, and the detection Recalls for LiDAR, camera and radar perception is 0.04, 0.07 and 0, respectively. The final detection Recall after fusion is 0.1, and the ASR is 90%. We can see that the two adversarial objects are enough to hide the target vehicle from the sensor fusion system in the two scenes.

Attack consequence. We also evaluate the attack consequences on an open-sourced autonomous driving platform, Baidu Apollo, to demonstrate the potential safety threat that may caused by the attack. We select the attack scenes shown

Table 2: Performance w.r.t. distance

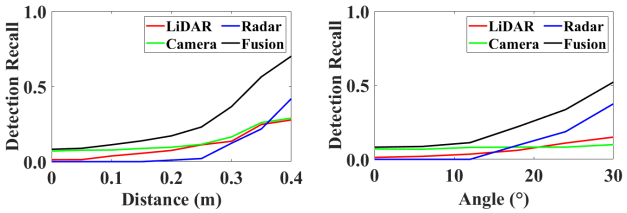
Distance (m)	30-25	25-20	20-15	15-10	10-5
Recall-LiDAR	0.00	0.00	0.00	0.05	0.09
Recall-camera	0.00	0.00	0.00	0.00	0.16
Recall-radar	0.00	0.00	0.00	0.00	0.0
Recall-fusion	0.00	0.00	0.00	0.05	0.24

in Figure 6 and evaluate the driving behavior of the victim AV during the attack through Baidu Apollo. Specifically, we feed the generated perception results into Baidu Apollo and evaluate the output of its planning module. As shown in Figure 7, without the attack, the victim AV successfully stops at $8m$ before the target vehicle. After the attack, the victim AV collides with the target vehicle in both of the two scenes.

The reasons for collisions can be described as follows. On one hand, our investigation shows that, as the victim AV drives towards the target vehicle, the target vehicle can be detected only when their distance are close enough. To demonstrate this point, we divide the collected data frames into different groups according to the distance between the victim AV and the target. Table 2 summarizes the average detection Recall in different groups. We can see that the target vehicle is completely hidden by the attack when their distance is larger than $15m$ and have high probability (95%) of being hidden when the distance is larger than $10m$. In real driving scenarios, $10m$ is close enough to cause collisions because it can be shorter than the victim AV's minimum braking distance [30]. On the other hand, even when the victim AV drives close to the target vehicle, the target vehicle is detected sporadically only in a few frames. This sporadic detection may be ignored as false alarms by the tracking modules in the victim AV. According to existing tracking algorithms, a valid detection requires an object to be detected multiple times in a given time window (adopted in Baidu Apollo and Autoware [2, 3]). In existing autonomous driving systems [2, 3], the time window is usually set to a small value to help eliminate false alarms and to avoid vehicle freezing or frequent braking behaviors. Thus, the sporadic detection in only a few frames will not be considered valid, which makes the succeeding planning module not able to avoid the collision.

6.3 Attack Robustness

Stability of drones. When using drones to carry the objects, precisely placing the objects at the derived locations with the derived orientation could be challenging. To study the effect of imprecise placement of the objects, we perturb the locations and orientations of the objects to different values in the physical world. We randomly perturb the locations to different values ranging from 0.05m to 0.5m. Also, we randomly perturb the orientations to different values ranging from 0° to 30°. Figure 8 shows the attack performance with respect



(a) Location error **(b) Orientation error**

Figure 8: Impact of imprecise placement.

to different perturbation values. We can find that the detection Recall is still around 0.1 (ASR is 90%) when the location error is $0.15m$ and orientation error is 12°. This is because we consider these errors in our attack framework as discussed in Section 5.5. With control algorithms and various sensors such as IMU, camera and ultrasound sensors, the drones in our experiments can stably hover at the desired positions with an average location error of $0.1m$ and orientation error of 5°. Moreover, empowered by Real-time kinematic (RTK) positioning systems or other advanced drone localization techniques [7, 23, 25, 44, 48, 72, 75, 84], the industrial-level drones can be controlled with centimeter-level accuracy (e.g., the control accuracy of DJI Matrice 300 RTK [5] equipped with D-RTK 2 mobile station [4] can be up to $1cm$). However, using industrial-level drones to launch the attack is very expensive (e.g., the DJI Matrice 300 RTK and D-RTK 2 mobile station costs $17,300). And using these expensive drones is not necessary. The control accuracy of most recreational drones is enough to achieve the attack goal, due to the robustness of the attack to location and orientation errors.

Driving direction of the victim AV. In practice, the victim AV may not drive exactly behind the target vehicle. For example, the victim AV may drive on the left/right side behind the target vehicle. To evaluate this effect, we drive the victim AV on the left side, right side, and exactly behind the target vehicle. The average Recall after the attack is 0.12, 0.09 and 0.09, respectively. We can find that the objects can achieve similar performance due to the consideration of various driving conditions in the attack framework (Section 5.5).

Passing-by vehicles. When launching the attack, there might be some other vehicles passing by the target. To study this effect, we perform the same attacks in Figure 6 when there is another vehicle passing by. We found that the average detection recall after the attack is still 0.11. This shows that the passing-by vehicles has little effect on the attacks.

Speed of the victim AV. The victim AV may drive towards the target vehicle in different speed. To study the effect of different speed, we evaluate attack performance when the victim AV's speed is 5mph, 10mph, 15mph, and 20mph. The average Recalls after the attack are 0.10, 0.08, 0.11, and 0.07, respectively. Thus, the speed of the vehicle has little impact on the attack. This is because the perception system collects

the data and detects the objects at each timestamp independently. The speed has no impact on the collected data and the detection results.

6.4 Alternative Object Carriers

What we proposed in this paper is a general attack framework which can optimize the number, locations, orientations, sizes and color patterns of the adversarial objects. The proposed attack framework does not rely on specific object carriers such as drones. Here we explore alternative carriers for the adversarial objects. Specifically, our proposed adversarial objects can be camouflaged as car advertisements and mounted on the target vehicle. To demonstrate the feasibility of this object carrier, we perform the attack in the scenario shown in Figure 9. We use the proposed attack framework to optimize the number, sizes, locations, orientations, and color patterns of the adversarial objects. To ensure that the objects adhere properly to the target vehicle, their locations are restricted to the vehicle's surface during the optimization process. Figure 9 shows the generated adversarial objects. The lower object is camouflaged as an advertisement poster stuck on the target vehicle. The upper object is camouflaged as a part of the car roof sign, which is mounted on the target vehicle using a roof rack. We evaluate the attack performance using our Lincoln MKZ AV testbed. Before the attack, we drive the victim AV towards the target vehicle from 30m to 5m and collect a sequence of data frames, which is the same evaluation procedure as that of the drone-assisted attacks (Section 6.2). Overall, the victim AV collect 161 frames, and the detection recalls for LiDAR, camera and radar are 0.92, 1.00 and 1.00, respectively. After the attack, the victim AV collect 138 frames, and the detection recalls for LiDAR, camera and radar are reduced to 0.06, 0.01 and 0.00, respectively. The results demonstrate the effectiveness of using this object carrier to launch the attack.

7 EXPERIMENTS ON A PUBLIC DATASET

7.1 Experimental Setup

To further demonstrate the effectiveness of the proposed attacks, we also evaluate them on the public KITTI dataset [24]. In our experiments, we randomly sample 100 scenes from this dataset. In each sampled scene, we randomly select one vehicle in front of the victim AV as the target vehicle. The radar signals are generated by converting the point cloud into meshes and using the simulation method in [78]. We then generate the adversarial objects for each sampled scene using the proposed attack framework. The model and attack settings as well as evaluation metrics are the same as that in real-world experiments.

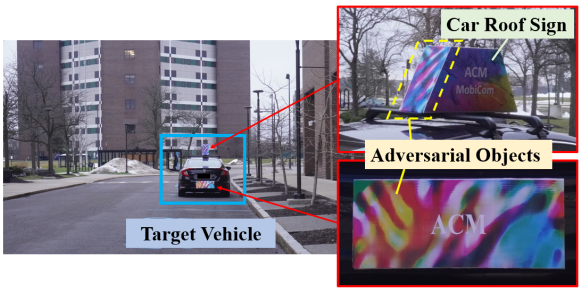


Figure 9: Attack using car signs.

7.2 Comparison to Baselines

We compare the proposed attack with a baseline developed in [70] whose goal is to hide the target vehicle from the camera-LiDAR fusion system. This baseline method proposes to place a 3D printed object on the roof top of a vehicle. We denote this baseline as PhyObj. Table 3 shows the attack performance of our proposed attack **AdvBoard** and PhyObj. We can see that PhyObj achieves the best performance on attacking camera perception. However, the performance of our proposed attack on attacking LiDAR performance is better than that of PhyObj. This is because PhyObj fixes the location of the adversarial object and only optimizes the object's shape, while our attack mainly optimizes the locations of the objects. For LiDAR perception, the locations of the injected objects have much larger impact on its perception results, compared with the shapes of objects. This shows that manipulating the locations of adversarial objects is a stronger attack vector than manipulating their shapes. In addition, PhyObj cannot affect radar perception. Thus, it cannot be used to attack the fusion system that involves radar.

Table 3: Comparison to baselines

Method	Detection recall				ASR	Avg N	L_{area} (m^2)
	L	C	R	F			
PhyObj	0.51	**0.01**	1.00	1.00	0%	-	-
AdvBoard-GA	0.33	0.21	0.35	0.66	34%	2.27	0.18
AdvBoard-RandLoc	0.93	0.45	0.88	0.99	1%	2.95	0.22
AdvBoard	**0.02**	0.03	**0.01**	**0.06**	**94%**	2.41	0.21

To demonstrate the effectiveness of the proposed optimization framework, we also use different methods to generate the parameters of the proposed adversarial objects and evaluate their performance. Genetic algorithm has been proved to be effective in generating adversarial examples when gradient information is not available [8, 71]. Thus, we use the genetic algorithm [8] to directly optimize the parameters in Eq (1), which is referred to as AdvBoard-GA. In addition, to demonstrate the effectiveness of the proposed two-step optimization design and the importance of the location probing step, we randomly generate the object locations P_l^* and alternatively update other parameters using the same method in Section 5.4, which is referred to as AdvBoard-RandLoc.

Table 4: Performance on different fusion systems

Model	Sensors	Type	Recall	Avg N	L_{area} (m^2)
BEVFusion	C+L	feature	1.00 / 0.08	2.21	0.21
CRFNet	C+R	feature	1.00 / 0.04	2.65	0.22
Radarnet	L+R	feature	1.00 / 0.02	2.35	0.20
LFusion	C+R	feature	1.00 / 0.00	2.77	0.24
RRPN	C+R	cascaded	1.00 / 0.00	2.58	0.21
HD-FPNet	C+L+R	cascaded	1.00 / 0.10	2.62	0.22

Table 3 summarizes the attack performance of these baseline methods. We can see that AdvBoard achieves better performance than AdvBoard-GA and AdvBoard-RandLoc, which demonstrates the advantage of our proposed framework.

7.3 Attacks on different sensor fusion systems

As discussed in Section 5.5, besides decision-level fusion, the proposed attack framework can also be applied to feature-level and cascaded fusion. In this section, we evaluate the attack performance on more state-of-the-art sensor fusion systems. For feature-level fusion, we consider BEVFusion [43], CRFNet [49], LFusion [34], and Radarnet [79]. For cascaded fusion, we consider RRPN [46] and HD-FPNet [74]. The attack scenes are the same as that in Table 3. N^0 and P_s^0 is set to 3 and $0.5m$, respectively. Table 4 summarizes the adopted sensors (C, L, and R represents camera, LiDAR and radar, respectively), as well as the attack performance on these fusion systems. The detection Recalls without attack (the left part in the Recall column) are 1.0 for all fusion systems. After the attack, their detection Recalls (the right part in the Recall column) are significantly reduced, which demonstrates the generalizability of the proposed framework.

8 VULNERABILITY ANALYSIS OF SENSOR FUSION SYSTEMS

In this section, we propose a vulnerability analysis framework that can estimate how important role a sensor play in the fusion system. According to Table 1 and the proposed attack framework in Section 5, updating the color pattern P_c mainly changes the outputs of camera perception, updating the object orientations P_o mainly changes the outputs of radar perception, and updating the object locations P_l mainly changes the outputs of LiDAR perception. By individually updating each parameter of $\{P_l, P_o, P_c\}$ in the attack framework, we can separately manipulate the outputs of the LiDAR, radar and camera perception models, respectively, and study its impact on sensor fusion by evaluating the changes of the final detection confidence after fusion. The analysis framework works as follows: we first randomly select some scenes and target vehicles from public datasets or our collected real-world data. Then we perform the proposed attack framework to update the locations P_l, orientations P_o, and color pattern P_c. For each selected scene, we record the maximum changes of the final detection confidences after fusion, i.e., V_{lidar}, V_{radar} and V_{camera}, when updating P_l, P_o and P_c, respectively. The final values of $\{\tilde{V}_{lidar}, \tilde{V}_{radar}, \tilde{V}_{camera}\}$ are obtained by performing a standard normalization on $\{V_{lidar}, V_{radar}, V_{camera}\}$ for each scene and then taking the average values for all the scenes. $\{\tilde{V}_{lidar}, \tilde{V}_{radar}, \tilde{V}_{camera}\}$ measures the impact of attacking different sensing modalities on the fusion results, which can be used to estimate the importance of different sensors on the sensor fusion system.

Table 5: Vulnerability analysis on sensor fusion

Model	Type	$\tilde{V}_{lidar}$	$\tilde{V}_{cam}$	$\tilde{V}_{radar}$
WBFusion	decision	0.37	0.35	0.28
BEVFusion	feature	**0.81**	0.19	0.00
CRFNet	feature	0.0	**0.77**	0.23
Radarnet	feature	**0.90**	0.00	0.10
LFusion	feature	0.00	0.31	**0.69**
RRPN	cascaded	0.00	0.01	**0.99**
HD-FPNet	cascaded	0.19	**0.72**	0.09

Table 5 summarizes the values of $\tilde{V}_{lidar}$, $\tilde{V}_{radar}$, and $\tilde{V}_{camera}$ for various sensor fusion systems including those in Table 4 and Weighted Box Fusion (WBFusion) [62], which is a widely adopted decision-level fusion model. We can find that, for feature-level and cascaded fusion systems, the fusion results can be significantly affected by attacking a specific sensing modality. This shows that these sensor fusion systems tend to rely on a single type of sensor and treat other sensors as auxiliaries. For the decision-level fusion method WBFusion, the three sensors are equally important to the sensor fusion system. Thus, it is more robust than the feature-level and cascaded fusion systems. Attacking a subset of sensors may not be able to affect its fusion results. But we can still attack WBFusion by simultaneously attacking all three sensors using the proposed framework in Section 5.

9 DISCUSSION

Potential defense. According to Section 5.3, the attack goal is achieved by first determining some important locations P_l^* and then placing some adversarial objects around them. To mitigate the attack, a possible solution is to degrade the impact of P_l^* on the output of sensor fusion and make it difficult for the attacker to derive those important locations. To achieve this, we propose to modify the training process of the sensor fusion system. Specifically, we combine the original training loss of the perception model with a new loss, i.e., $L_{def} = \sum_{P_l \in H} S(P_l)$, where $S(P_l)$ measures the impact of adding objects to locations P_l on the perception result. In practice, we can take the objective function in optimization problem (2) as $S(P_l)$ because that objective function is also used to measure the impact of locations P_l. H is a location set that contains the potential important locations required by

the proposed attack. To obtain H, we can randomly sample some target vehicles from the training data and randomly probe many sets of locations around each target vehicle. H is the union of the probed location sets whose values of $S()$ are larger than a threshold. The workflow of this defense strategy can be described as follows. We first train a perception model using the original training loss. We randomly sample 50% of the training data in KITTI dataset and select all the vehicles in these training samples to obtain the location set H. In total, we select 6851 vehicles. Then, we retrain the model by considering the new loss L_{def}. We apply the proposed defense method to train the same sensor fusion system described in Section 6.1 and evaluate the attack performance. Without the attack, the detection Recall after fusion is 1.00. After the attack, the detection Recall is 0.50 and the ASR is 50%. Compared with the results in Table 3, the perception model trained with the defense strategy are less vulnerable to the attack, but the ASR is still high (around 50%).

Stealthiness of drone-assisted attack. In our experiments, we use drones to launch the attacks. In practice, it is challenging to detect the drones because they only need to hover for a few seconds and fly away immediately after the attack. For example, in our experiments, the drones only need to hover for 2.8s when the speed of the victim AV is 20mph. In addition, even if the drones are noticed, it is difficult to identify if the drones are controlled by attackers or benign users such as recreational flyers. For example, some recreational flyers may use drones to hover around their cars to record videos. Some tethered drones may hover around the patrol vehicle for surveillance. Moreover, with the deployment of drone delivery services, the attacker can camouflage their drones as those used by the delivery company.

10 RELATED WORK

Prior works have studied the safety and vulnerability of vehicular systems [11, 16, 20, 32, 39, 42, 81], especially the perception systems of AVs. To attack camera perception in AVs, some works propose to use stickers or painting in special color [64, 77]. To attack LiDAR perception, [12, 14, 29, 65] propose to spoof LiDAR by transmitting laser signals to inject/remove points, and [71, 82, 85, 87] propose to place some physical objects on/around the target vehicle to hide the vehicle from the LiDAR perception. To attack radar perception, most existing studies propose to use some spoofing devices to actively transmit special signals to the victim radar [15, 35, 47, 66, 73]. However, these attack methods target on only a single type of sensor, and would not be able to fool a multi-sensor fusion based AV perception system.

In addition, existing active attacks on radar suffer some practical challenges when being performed in the real world. These attacks require sub-nanosecond-level synchronization between the spoofing devices and the victim radar [15, 35], or require the devices to be placed at a fixed angle/distance to the victim radar [47, 66]. So in their experiments, they normally used a wired link to connect their devices to the victim radar, or kept the radar and the devices stationary during the attack. Compared with these attacks, our proposed adversarial objects rely on passive reflection, which can achieve the attack goal by simply placing the objects in the driving environment. Although [17] proposed to use some special radar absorbing materials to attack radar perception, the adopted material can only work within a specific radar frequency range, i.e., 18-40GHz. Today's mmWave radar perception systems in AVs usually operate in a much higher frequency, e.g., the radar used in Baidu Apollo operates in 77 GHz [3]. Compared with this attack, our proposed objects does not rely on any special material and can work on any mmWave frequency. This is because the objects leverage the specular reflection on a metal surface, which is barely affected by the mmWave frequency. [86] proposes to use some 3D printed objects to attack deep learning based radar perception models. However, the radar perception models in existing AVs' multi-sensor fusion systems do not rely on deep learning. In contrast to [86], this paper proposes to attack the conventional radar perception models that have been widely adopted in AVs' sensor fusion systems.

Although there are a few studies on attacking camera-LiDAR fusion systems [6, 13, 18, 28, 70], they can not be used to attack the sensor fusion systems that involve radar, which is widely adopted in existing AVs [2, 3]. In contrast, our proposed objects can attack sensor fusion systems that contain all three types of sensors.

11 CONCLUSION

This paper presents the first study on the vulnerability of multi-sensor fusion systems incorporating LiDAR, camera, and radar in autonomous driving. We propose a new type of adversarial object that can be simultaneously attack the above three types of sensors. By placing some adversarial objects at some specific locations and orientations, we can continuously hide a target vehicle from the victim AV's perception system. Real world experiments demonstrate the the attack can be easily achieved by using only two small objects. We also propose a general framework that can not only identify vulnerable fusion systems that rely on a subset (or only one) of sensors but also provide guidance for the design of robust sensor fusion systems.

12 ACKNOWLEDGEMENTS

We thank our anonymous reviewers for their insightful comments and suggestions on this paper. This work was supported in part by the US National Science Foundation under grant PFI-2044670, CNS-2120369, and CNS-2154059.

REFERENCES

[1] 2012. Intro to Rendering, Ray Casting. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-837-computer-graphics-fall-2012/lecture-notes/MIT6837F12Lec11.pdf.

[2] 2015. Autoware Foundation. https://www.autoware.org/

[3] 2020. Baidu Apollo. https://developer.apollo.auto/

[4] 2020. D-RTK 2 Mobile Station. https://www.dji.com/d-rtk-2

[5] 2021. DJI Matrice 300. https://enterprise.dji.com/matrice-300

[6] Mazen Abdelfattah, Kaiwen Yuan, Z Jane Wang, and Rabab Ward. 2021. Towards Universal Physical Attacks On Cascaded Camera-Lidar 3D Object Detection Models. *arXiv preprint arXiv:2101.10747* (2021).

[7] Tasnim Azad Abir, Endrowednes Kuantama, Richard Han, Judith Dawes, Rich Mildren, and Phuc Nguyen. 2023. Towards Robust Lidar-based 3D Detection and Tracking of UAVs. In *Proceedings of the Ninth Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. 1–7.

[8] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B Srivastava. 2019. Genattack: Practical black-box attacks with gradient-free optimization. In *Proceedings of the genetic and evolutionary computation conference*. 1111–1119.

[9] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *The Journal of Machine Learning Research* 11 (2010), 1803–1831.

[10] Kshitiz Bansal, Keshav Rungta, and Dinesh Bharadia. 2022. Radsegnet: A reliable approach to radar camera fusion. *arXiv preprint arXiv:2208.03849* (2022).

[11] Stefania Bartoletti, Henk Wymeersch, Tomasz Mach, Oliver Brunnegård, Domenico Giustiniano, Peter Hammarberg, Musa Furkan Keskin, Jesus O Lacruz, Sara Modarres Razavi, Joakim Rönnblom, et al. 2021. Positioning and sensing for vehicular safety applications in 5G and beyond. *IEEE Communications Magazine* 59, 11 (2021), 15–21.

[12] Yulong Cao, S Hrushikesh Bhupathiraju, Pirouz Naghavi, Takeshi Sugawara, Z Morley Mao, and Sara Rampazzi. 2022. You can't see me: physical removal attacks on LiDAR-based autonomous vehicles driving frameworks. In *Proceedings of 31th {USENIX} Security Symposium ({USENIX} Security 22)*.

[13] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. 2021. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 176–194.

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

[15] Ruchir Chauhan. 2014. *A platform for false data injection in frequency modulated continuous wave radar*. Utah State University.

[16] Dongyao Chen, Mert D Pesé, and Kang G Shin. [n. d.]. Guess Which Car Type I Am Driving: Information Leak via Driving Apps. In *Network and Distributed Systems Security (NDSS) Symposium*.

[17] Xingyu Chen, Zhengxiong Li, Baicheng Chen, Yi Zhu, Chris Xiaoxuan Lu, Zhenyu Peng, Feng Lin, Wenyao Xu, Kui Ren, and Chunming Qiao. 2023. MetaWave: Attacking mmWave Sensing with Meta-material-enhanced Tags. In *Network and Distributed Systems Security (NDSS) Symposium*.

[18] Zhiyuan Cheng, Hongjun Choi, James Liang, Shiwei Feng, Guanhong Tao, Dongfang Liu, Michael Zuzak, and Xiangyu Zhang. 2023. Fusion is Not Enough: Single-Modal Attacks to Compromise Fusion Models in Autonomous Driving. *arXiv preprint arXiv:2304.14614* (2023).

[19] Mallesham Dasari, Ramanujan K Sheshadri, Karthikeyan Sundaresan, and Samir R Das. 2023. RoVaR: Robust Multi-agent Tracking through Dual-layer Diversity in Visual and RF Sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–25.

[20] Shaohua Ding, Yulong Tian, Fengyuan Xu, Qun Li, and Sheng Zhong. 2019. Trojan attack on deep generative models in autonomous driving. In *Security and Privacy in Communication Networks: 15th EAI International Conference, SecureComm 2019, Orlando, FL, USA, October 23-25, 2019, Proceedings, Part I 15*. Springer, 299–318.

[21] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.

[22] Tiantian Feng, Digbalay Bose, Tuo Zhang, Rajat Hebbar, Anil Ramakrishna, Rahul Gupta, Mi Zhang, Salman Avestimehr, and Shrikanth Narayanan. 2023. FedMultimodal: A Benchmark For Multimodal Federated Learning. *arXiv preprint arXiv:2306.09486* (2023).

[23] Nakul Garg and Nirupam Roy. 2023. Sirius: A self-localization system for resource-constrained iot sensors. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*. 289–302.

[24] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3354–3361.

[25] Mahanth Gowda, Justin Manweiler, Ashutosh Dhekne, Romit Roy Choudhury, and Justin D Weisz. 2016. Tracking drone orientation with multiple GPS receivers. In *Proceedings of the 22nd annual international conference on mobile computing and networking*. 280–293.

[26] Junfeng Guan, Sohrab Madani, Waleed Ahmed, Samah Hussein, Saurabh Gupta, and Haitham Hassanieh. 2023. Exploiting Virtual Array Diversity for Accurate Radar Detection. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[27] Junfeng Guan, Sohrab Madani, Suraj Jog, Saurabh Gupta, and Haitham Hassanieh. 2020. Through fog high-resolution imaging using millimeter wave radar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11464–11473.

[28] R Spencer Hallyburton, Yupei Liu, Yulong Cao, Z Morley Mao, and Miroslav Pajic. 2022. Security Analysis of {Camera-LiDAR} Fusion Against {Black-Box} Attacks on Autonomous Vehicles. In *31st USENIX Security Symposium (USENIX Security 22)*. 1903–1920.

[29] Zhongyuan Hau, Kenneth T Co, Soteris Demetriou, and Emil C Lupu. 2021. Object removal attacks on lidar-based 3d object detectors. *arXiv preprint arXiv:2102.03722* (2021).

[30] Jack D Jernigan, Meltem F Kodaman, et al. 2001. *An investigation of the utility and accuracy of the table of speed and stopping distances specified in the Code of Virginia.* Technical Report. Virginia Transportation Research Council.

[31] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7122–7131.

[32] Gorkem Kar, Hossen Mustafa, Yan Wang, Yingying Chen, Wenyuan Xu, Marco Gruteser, and Tam Vu. 2014. Detection of on-road vehicles emanating GPS interference. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 621–632.

[33] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3907–3916.

[34] Jinhyeong Kim, Youngseok Kim, and Dongsuk Kum. 2020. Low-level sensor fusion network for 3d vehicle detection using radar range-azimuth heatmap and monocular image. In *Proceedings of the Asian*

Conference on Computer Vision.

[35] Rony Komissarov and Avishai Wool. 2021. Spoofing attacks against vehicular FMCW radar. In *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security*. 91–97.

[36] Belal Korany, Chitra R Karanam, Hong Cai, and Yasamin Mostofi. 2019. XModal-ID: Using WiFi for through-wall person identification from candidate video footage. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–15.

[37] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12697–12705.

[38] Youngki Lee, Chulhong Min, Chanyou Hwang, Jaeung Lee, Inseok Hwang, Younghyun Ju, Chungkuk Yoo, Miri Moon, Uichin Lee, and Junehwa Song. 2013. Sociophone: Everyday face-to-face interaction monitoring platform using multi-phone sensor fusion. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. 375–388.

[39] Patrick Leu, Mridula Singh, Marc Roeschlin, Kenneth G Paterson, and Srdjan Čapkun. 2020. Message time of arrival codes: A fundamental primitive for secure distance measurement. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 500–516.

[40] Tianxing Li, Jin Huang, Erik Risinger, and Deepak Ganesan. 2021. Low-latency speculative inference on distributed multi-modal data streams. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 67–80.

[41] Liangkai Liu, Zheng Dong, Yanzhi Wang, and Weisong Shi. 2022. Prophet: Realizing a predictable real-time perception pipeline for autonomous vehicles. In *2022 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 305–317.

[42] Liangkai Liu, Xingzhou Zhang, Mu Qiao, and Weisong Shi. 2018. SafeShareRide: Edge-based attack detection in ridesharing services. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 17–29.

[43] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. 2023. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2774–2781.

[44] Wenguang Mao, Zaiwei Zhang, Lili Qiu, Jian He, Yuchen Cui, and Sangki Yun. 2017. Indoor follow me drone. In *Proceedings of the 15th annual international conference on mobile systems, applications, and services*. 345–358.

[45] Fraser McLean, Leyang Xue, Chris Xiaoxuan Lu, and Mahesh Marina. 2022. Towards edge-assisted real-time 3D segmentation of large scale LIDAR point clouds. In *Proceedings of the 6th International Workshop on Embedded and Mobile Deep Learning*. 1–6.

[46] Ramin Nabati and Hairong Qi. 2019. Rrpn: Radar region proposal network for object detection in autonomous vehicles. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3093–3097.

[47] Prateek Nallabolu and Changzhi Li. 2021. A Frequency-Domain Spoofing Attack on FMCW Radars and Its Mitigation Technique Based on a Hybrid-Chirp Waveform. *IEEE Transactions on Microwave Theory and Techniques* 69, 11 (2021), 5086–5098.

[48] Phuc Nguyen, Hoang Truong, Mahesh Ravindranathan, Anh Nguyen, Richard Han, and Tam Vu. 2017. Matthan: Drone presence detection by identifying physical signatures in the drone's rf communication. In *Proceedings of the 15th annual international conference on mobile systems, applications, and services*. 211–224.

[49] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz, and Markus Lienkamp. 2019. A deep learning-based radar and camera sensor fusion architecture for object detection. In *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, 1–7.

[50] John Nolan, Kun Qian, and Xinyu Zhang. 2021. RoS: passive smart surface for roadside-to-vehicle communication. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 165–178.

[51] Akarsh Prabhakara, Tao Jin, Arnav Das, Gantavya Bhatt, Lilly Kumari, Elahe Soltanaghai, Jeff Bilmes, Swarun Kumar, and Anthony Rowe. 2023. High resolution point clouds from mmwave radar. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4135–4142.

[52] Akarsh Prabhakara, Diana Zhang, Chao Li, Sirajum Munir, Aswin C Sankaranarayanan, Anthony Rowe, and Swarun Kumar. 2022. Exploring mmWave Radar and Camera Fusion for High-Resolution and Long-Range Depth Imaging. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3995–4002.

[53] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. 2021. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 444–453.

[54] Hang Qiu, Pohan Huang, Namo Asavisanu, Xiaochen Liu, Konstantinos Psounis, and Ramesh Govindan. 2021. Autocast: Scalable infrastructure-less cooperative perception for distributed collaborative driving. *arXiv preprint arXiv:2112.14947* (2021).

[55] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D Lane, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. 2018. Multimodal deep learning for activity and context recognition. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 1, 4 (2018), 1–27.

[56] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[57] Hem Regmi, Moh Sabbir Saadat, Sanjib Sur, and Srihari Nelakuditi. 2021. Squigglemilli: Approximating sar imaging on mobile millimeter-wave devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–26.

[58] Wenjie Ruan, Quan Z Sheng, Peipei Xu, Lei Yang, Tao Gu, and Longfei Shangguan. 2017. Making sense of doppler effect for multi-modal hand motion detection. *IEEE Transactions on Mobile Computing* 17, 9 (2017), 2087–2100.

[59] Wenjie Ruan, Quan Z Sheng, Lina Yao, Lei Yang, and Tao Gu. 2016. HOI-Loc: Towards unobstructive human localization with probabilistic multi-sensor fusion. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, 1–4.

[60] Louis L Scharf and Cédric Demeure. 1991. Statistical signal processing: detection, estimation, and time series analysis. *(No Title)* (1991).

[61] Xian Shuai, Yulin Shen, Yi Tang, Shuyao Shi, Luping Ji, and Guoliang Xing. 2021. millieye: A lightweight mmwave radar and camera fusion system for robust object detection. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. 145–157.

[62] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. 2021. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing* 107 (2021), 104117.

[63] Elahe Soltanaghai, Akarsh Prabhakara, Artur Balanuta, Matthew Anderson, Jan M Rabaey, Swarun Kumar, and Anthony Rowe. 2021. Millimetro: mmWave retro-reflective tags for accurate, long range localization. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 69–82.

[64] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. 2018. Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*.

[65] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. 2020. Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures.

In *Proceedings of 29th {USENIX} Security Symposium ({USENIX} Security 20)*. 877–894.

[66] Zhi Sun, Sarankumar Balakrishnan, Lu Su, Arupjyoti Bhuyan, Pu Wang, and Chunming Qiao. 2021. Who is in control? Practical physical layer attack and defense for mmWave-based sensing in autonomous vehicles. *IEEE Transactions on Information Forensics and Security* 16 (2021), 3199–3214.

[67] Lei Tian, Rahat Rafiq, Shaosong Li, David Chu, Richard Han, Qin Lv, and Shivakant Mishra. 2014. Multi-modal fusion for flasher detection in a mobile video chat application. In *11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*.

[68] Panrong Tong, Mingqian Li, Mo Li, Jianqiang Huang, and Xiansheng Hua. 2021. Large-scale vehicle trajectory reconstruction with camera sensing network. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 188–200.

[69] Richard Yi-Chia Tsai, Hans Ting-Yuan Ke, Kate Ching-Ju Lin, and Yu-Chee Tseng. 2019. Enabling identity-aware tracking via fusion of visual and inertial features. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2260–2266.

[70] James Tu, Huichen Li, Xinchen Yan, Mengye Ren, Yun Chen, Ming Liang, Eilyan Bitar, Ersin Yumer, and Raquel Urtasun. 2021. Exploring adversarial robustness of multi-sensor perception systems in self driving. *arXiv preprint arXiv:2101.06784* (2021).

[71] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. 2020. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13716–13725.

[72] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. {Decimeter-Level} localization with a single {WiFi} access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. 165–178.

[73] Rohith Reddy Vennam, Ish Kumar Jain, Kshitiz Bansal, Joshua Orozco, Puja Shukla, Aanjhan Ranganathan, and Dinesh Bharadia. 2022. mmSpoof: Resilient Spoofing of Automotive Millimeter-wave Radars using Reflect Array. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 1971–1985.

[74] Leichen Wang, Tianbai Chen, Carsten Anklam, and Bastian Goldluecke. 2020. High dimensional frustum pointnet for 3d object detection from camera, lidar, and radar. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1621–1628.

[75] Weiguo Wang, Luca Mottola, Yuan He, Jinming Li, Yimiao Sun, Shuai Li, Hua Jing, and Yulei Wang. 2022. MicNest: Long-Range Instant Acoustic Localization of Drones in Precise Landing. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 504–517.

[76] Yichao Wang, Yili Ren, Yingying Chen, and Jie Yang. 2022. Wi-Mesh: A WiFi Vision-based Approach for 3D Human Mesh Construction. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 362–376.

[77] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. 2020. Adversarial t-shirt! evading person detectors in a physical world. In *Proceedings of the European Conference on Computer Vision*. Springer, 665–681.

[78] Hongfei Xue, Qiming Cao, Chenglin Miao, Yan Ju, Haochen Hu, Aidong Zhang, and Lu Su. 2023. Towards Generalized mmWave-based Human Pose Estimation through Signal Augmentation. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[79] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. 2020. Radarnet: Exploiting radar for robust perception of dynamic objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 496–512.

[80] Mingyu Yang, Roger Hsiao, Gordy Carichner, Katherine Ernst, Jaechan Lim, Delbert A Green, Inhee Lee, David Blaauw, and Hun-Seok Kim. 2021. Migrating monarch butterfly localization using multi-modal sensor fusion neural networks. In *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 1792–1796.

[81] Kexiong Curtis Zeng, Shinan Liu, Yuanchao Shu, Dong Wang, Haoyu Li, Yanzhi Dou, Gang Wang, and Yaling Yang. 2018. All your {GPS} are belong to us: Towards stealthy manipulation of road navigation systems. In *27th USENIX security symposium (USENIX security 18)*. 1527–1544.

[82] Yan Zhang, Yi Zhu, Zihao Liu, Chenglin Miao, Foad Hajiaghajani, Lu Su, and Chunming Qiao. 2022. Towards Backdoor Attacks against LiDAR Object Detection in Autonomous Driving. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 533–547.

[83] Mingmin Zhao, Yingcheng Liu, Aniruddh Raghu, Tianhong Li, Hang Zhao, Antonio Torralba, and Dina Katabi. 2019. Through-wall human mesh recovery using radio signals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10113–10122.

[84] Longyu Zhou, Supeng Leng, Qing Wang, and Qiang Liu. 2022. Integrated sensing and communication in UAV swarms for cooperative multiple targets tracking. *IEEE Transactions on Mobile Computing* (2022).

[85] Yi Zhu, Chenglin Miao, Foad Hajiaghajani, Mengdi Huai, Lu Su, and Chunming Qiao. 2021. Adversarial attacks against lidar semantic segmentation in autonomous driving. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 329–342.

[86] Yi Zhu, Chenglin Miao, Hongfei Xue, Zhengxiong Li, Yunnan Yu, Wenyao Xu, Lu Su, and Chunming Qiao. 2023. TileMask: A Passive-Reflection-based Attack against mmWave Radar Object Detection in Autonomous Driving. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 1317–1331.

[87] Yi Zhu, Chenglin Miao, Tianhang Zheng, Foad Hajiaghajani, Lu Su, and Chunming Qiao. 2021. Can we use arbitrary objects to attack lidar perception in autonomous driving?. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 1945–1960.