Physical Attack on Monocular Depth Estimation with Optimal Adversarial Patches

Zhiyuan Cheng[1], James Liang[2], Hongjun Choi[1], Guanhong Tao[1], Zhiwen Cao[1], Dongfang Liu[2,*], and Xiangyu Zhang[1,*]

[1] Purdue University
[2] Rochester Institute of Technology
{cheng443, choi293, taog, cao270, xyzhang}@cs.purdue.edu
{jcl3689, dongfang.liu}@rit.edu

Abstract. Deep learning has substantially boosted the performance of Monocular Depth Estimation (MDE), a critical component in fully vision-based autonomous driving (AD) systems (*e.g.*, Tesla and Toyota). In this work, we develop an attack against learning-based MDE. In particular, we use an optimization-based method to systematically generate stealthy physical-object-oriented adversarial patches to attack depth estimation. We balance the stealth and effectiveness of our attack with object-oriented adversarial design, sensitive region localization, and natural style camouflage. Using real-world driving scenarios, we evaluate our attack on concurrent MDE models and a representative downstream task for AD (*i.e.*, 3D object detection). Experimental results show that our method can generate stealthy, effective, and robust adversarial patches for different target objects and models and achieves more than 6 meters mean depth estimation error and 93% attack success rate (ASR) in object detection with a patch of 1/9 of the vehicle's rear area. Field tests on three different driving routes with a real vehicle indicate that we cause over 6 meters mean depth estimation error and reduce the object detection rate from 90.70% to 5.16% in continuous video frames.

Keywords: Physical Adversarial Attack, Monocular Depth Estimation, Autonomous Driving.

1 Introduction

Monocular Depth Estimation (MDE) is a technique for estimating the distance between an object and the camera from RGB image inputs. It is a critical vision task for autonomous driving (AD) because it bridges the gap between Lidar sensors and RGB cameras [52] and its measurement has an effect on a variety of downstream perception tasks (*e.g.*, object detection [26,12], visual SLAM [55], and

* Corresponding authors

visual relocalization [27]). For its importance, Tesla has integrated MDE into its production-grade Autopilot system [2,3], and other AD companies such as Toyota [20] and Huawei [5] are also actively investigating this technique. With the increasing popularity of MDE, ensuring its security becomes a critical challenge.

Existing adversarial attacks against MDE are implemented in digital- [66,56] or physical-world platforms [63]. Compared to digital-world attacks, attacks in the physical world are more challenging because they require robust perturbations to overcome various photometric and geometric changes [6], reducing their stealth. Prior efforts for physical-world adversarial attacks [63,45,23,7] generally employ an unnatural-looking adversarial patch and sacrifice stealth for attack effectiveness, leaving plenty of room for improvement. Additionally, with MDE's rapid development, many downstream tasks that previously require expensive Lidar sensors or depth cameras can now be performed entirely with MDE's measurement and achieve competitive performance. However, the investigation of the impact of compromised MDE on these downstream tasks remains largely unknown.

To address the aforementioned problems, in this paper, we investigate the *stealth of physical-world attack against MDE* and present a physical-object-oriented adversarial patch optimization framework to generate *stealthy, effective and robust adversarial patches* for target objects (*e.g.*, vehicles and pedestrians), which, to our best knowledge, is the **FIRST** work in the community. In particular, we are able to achieve the followings: ❶ we design a physical-object-oriented adversarial optimization, which binds the patch and the target object together regarding attack effects and physical-world transformations (§3.2); ❷ we optimize the patch region on the target object with a differentiable patch mask representation, which automatically locates the highly effective area for attack on the target object and improves attack performance with a small patch size (§3.3); ❸ we camouflage the adversarial pattern with natural styles (*e.g.*, rusty and dirty) with deep photo style transfer [29], resulting in stealthier patch for the attack (§3.4); ❹ we investigate the impact of compromised MDE on a representative downstream task in AD — 3D object detection (§4.4). Our attack causes over 6 meters of mean depth estimation error for a real vehicle, with a patch only 1/9 of the vehicle's rear area, and achieves more than 90% attack success rate in 3D object detection (Fig. 1).

2 Related Work

AD Systems Security. In AD, sensor security and autonomy software security are the two important challenges. For sensor security, prior works focus on spoofing/jamming on camera [64,32,35], LiDAR [10,44], RADAR [64], ultrasonic [64], GPS [43] and IMU [50,48]. For autonomy software security, some prior works study regression tasks (*e.g.*, depth estimation [63] and optical flow estimation [37]), and others focus on classification tasks (*e.g.*, 2D object detection and classification [45,7], tracking [21], lane detection [40,41], and traffic light detection [46]). This work focuses on autonomy software security, that is, compromising MDE and its related downstream tasks.

Fig. 1: Attack MDE and 3D object detection with a natural adversarial patch. The left is a benign scenario and the right is the corresponding adversarial scenario. 3D object detection takes the pseudo-Lidar (*i.e.*, point cloud projected from 2D depth map) as input and outputs bounding boxes of recognized objects. Observe in the adversarial scenario (b) that our optimized adversarial patch can disturb the depth estimation of the target vehicle significantly and the effect propagates to an area larger than the patch itself. Pseudo-Lidar of the vehicle is thus distorted and it cannot be detected in the downstream task.

Physical-world Adversarial Attacks. Many prior efforts in adversarial attacks have been directed toward generating patches or perturbations in the digital space [34,19,33,57,60,58,36,49,59]. In comparison, we conduct extensive experiments on adversarial attacks in the physical world. Although existing physical-world attacks have addressed tasks such as image classification [45,7], object detection [11,61,47], face recognition [42,23], the domain of depth estimation attack has received scant attention. Moreover, the correlations between stealth and attack effectiveness are largely understudied in the literature. In this paper, we make an attempt to close the aforementioned knowledge gap.
MDE Attacks. Zhang [66] proposes a multi-task attack strategy to improve the performance in the universal attack scenario. Wong [56] proposes a way to generate targeted adversarial perturbation on images and alter the depth map arbitrarily. These two attacks focus on digital-space perturbations thus are not directly applicable in the physical world. Yamanaka [63] proposes a method to generate printable adversarial patch for MDE but it does not consider stealth of the patch. Different from prior efforts, we focus on the stealth and to the best of our knowledge, we are the **FIRST** work to examine the stealth of adversarial patches for physical-world attack against MDE.

3 Method

3.1 Physical-object-oriented MDE Attack

Motivation. Compared with unconstrained adversarial patches (see Fig. 2a) which often look suspicious, stealthy patches may draw less attention and hence can stay on the target vehicle for an extended period of time, posing a greater threat. We divide the challenge of achieving stealth into two sub-problems: patch size minimization and achieving natural appearance. To minimize patch size, we investigate how to maximize the attack effect with smaller patches and propose two approaches: ❶ enlarging the patch's affected area (see comparison in Fig. 2b and c), and ❷ locating the adversarial patch in a more sensitive region of the

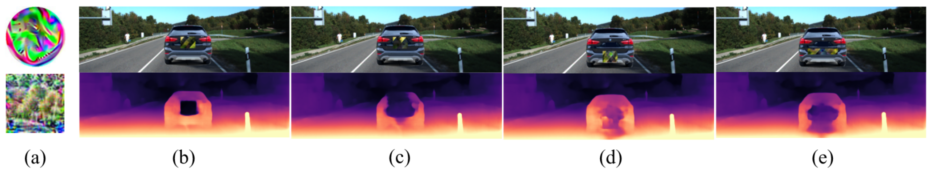


Fig. 2: (a): Unconstrained adversarial patches in [63,30] are easy to be identified; traditional patch-oriented attack in (b) affects smaller area than our object-oriented attack in (c); (c), (d) and (e): different regions on the target object have different sensitivity regarding attack effect even with the same total area.

target vehicle (see Fig. 2c, d and e). In terms of naturalness, as the magnitude of perturbations required to launch attack in the physical world is much more substantial, we cannot simply bound the adversarial noise to a human unnoticeable level via various L_p-norms as in digital-world attacks, which provides little physical-world robustness. Instead, we use style transfer to disguise the adversarial pattern as natural styles (*e.g.*, dirty or rusty).

Attack Pipeline. We use an optimization-based method to generate adversarial patches and there are three main optimization goals: ❶ increasing the estimated distance of target object (§3.2); ❷ minimizing the patch to locate a sensitive (*i.e.*, most effective) region for attack (§3.3), and ❸ camouflaging the adversarial patch with natural styles (§3.4). The optimization is conducted in the digital world. Fig. 3 shows the overview of our attack. From the top left, we start with style transfer on the patch content image. Next, we crop the style-transferred patch with an optimizable patch mask (m_p^{Θ}) and paste it onto a target object (O) (*e.g.*, a vehicle) creating an adversarial one (O'). Then, we synthesize adversarial scenarios (R'_t) by placing the adversarial object into random scenes with physical transformations (t) and estimate scenarios' depth ($\mathcal{D}(R'_t)$). We define an adversarial loss ($\mathcal{L}_a$) to increase depth of the target object. Together with a style transfer loss ($\mathcal{L}_{st}$) maintaining the naturalness and a patch size loss ($\mathcal{L}_m$) minimizing the patch, we perform back propagation and update the patch content and the mask iteratively to address the three optimization goals. The solid lines denote data flow and the dashed lines represent back propagation paths. Each component is explained in details in the following sections.

3.2 Adversarial Perturbation Generation

In preparation, we take a photo of the target object (O) and select a patch content image (x) and a style image. Given the patch mask (m_p), we create an adversarial object (O') by applying the style-transferred patch (x') on the benign object in the following way:

$$O' = O \odot (1 - m_p) + x' \odot m_p, \tag{1}$$

where $\odot$ denotes the element-wise multiplication and O, m_p, x' have the same width and height. We explain the patch mask definition and style transfer later

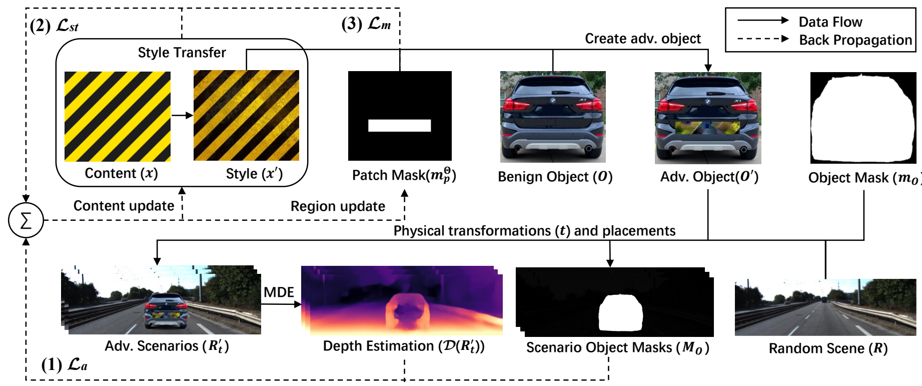


Fig. 3: Overview of the physical-object-oriented framework to generate a stealthy adversarial patch.

in §3.3 and §3.4. We evaluate the depth of the target object inside a scene because the camera on the victim vehicle captures scene frames as input instead of independent objects. Specifically, in each optimization iteration, we randomly sample a scene from the dataset and paste the adversarial object into the scene to create an adversarial scenario. Unlike previous attacks against autonomous driving systems [9,39] that aim at a particular scene or a road section, our attack is universal and scene-independent.

To improve the robustness of our attack in the physical world, we apply Expectation of Transformation (EoT) [6] by randomly transforming the object in size, rotation, brightness, saturation, etc., before pasting. The horizontal position of pasting is random, while the vertical position is calculated according to the size of the object considering physical constraints. Specifically, Fig. 4 shows the perspective model of a vehicle in a side view and we assume the camera is facing straight forward without tilt. H is the height of the target vehicle; h is the height of the camera with respect to the victim vehicle; f is the focus length of the camera and α relates to the camera's view angle. On the image, the vertical position of the vehicle (d) is calculated from the height of the vehicle (s) with Equation 2. Intuitively, objects farther away appear smaller in perspective so a smaller object after transformation is pasted to a higher vertical position on the image, which is closer to *the vanishing point* (of the camera), which denotes the furthest physical point in the camera view, and has further depth estimation.

$$d = -\frac{h}{H}s + \frac{f}{\tan\alpha} \tag{2}$$

Formally, the adversarial scenario R_t' is described as $R_t' = \Lambda_t\left(t(O' \odot m_o), R\right)$, where t is the random transformation applied on the target object; m_o is the object mask used to extract the object from the image; R is the randomly sampled scene from database and $\Lambda(\cdot,\cdot)$ is the paste operation to combine an adversarial object and a scene following the physical constraint in Equation 2. Since our goal is to make the target object further away, we want to maximize the

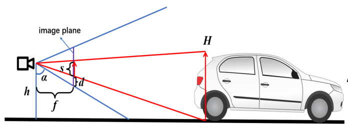


Fig. 4: Perspective projection of a vehicle (side view).

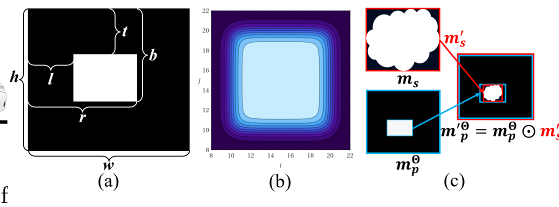


Fig. 5: Patch region definitions.

object's depth estimation (*i.e.*, minimize the reciprocal). Hence, we define the adversarial loss in Equation 3, where T is a set of transformations; D_R is a set of scenes; $MSE(\cdot,\cdot)$ is the mean square error between two variables; $\mathcal{D}$ is the depth estimation model and M_o is the object mask in the scenario.

$$\mathcal{L}_a = \mathbf{E}_{t\sim T, R\sim D_R}\left[MSE\left(\mathcal{D}\left(R'_t\right)^{-1} \odot M_o, 0\right)\right] \tag{3}$$

3.3 Sensitive Region Localization

As described in §3.2, we apply the style transferred patch x' onto the target object by a patch mask m_p which defines the patch region on the target object. Prior works [51,28,25] optimizing masks treat each pixel of the mask as a parameter and the generated mask suffers from low deployability due to sparse and scattered mask regions (See Fig. 11b). Instead we design a novel rectangular patch region optimization method (we call it *regional optimization*) to locate a sensitive region automatically. Although we define the patch region as rectangular, the final patch is not necessarily rectangular but have an arbitrary predefined shape. Details are explained later.

A typical rectangular patch mask has ones within the rectangular borders and zeros otherwise. However, this mask is not differentiable regarding the border parameters because the mask values are not continuous across the borders and border information is not encoded into each mask values, which means that the region cannot be optimized via gradient descent and back propagation. To solve this problem, we design a differentiable soft version of the rectangular mask making it optimizable with respect to four border parameters. Specifically, we define border parameters $\Theta = [l, r, t, b]$ as shown in Fig. 5a. l and r are the left and right borders' column indices and t and b are the top and bottom borders' row indices. Let w and h be the width and height of the mask respectively and we have $0 \leq l \leq r \leq w$ and $0 \leq t \leq b \leq h$.

$$\begin{aligned} m_p^\Theta &= \{m_p^\Theta[i,j] \mid i \in 1...w, j \in 1...h\} \\ m_p^\Theta[i,j] &= \frac{1}{4}(-sign(i-t)\cdot sign(i-b)+1) \\ &\quad \cdot(-sign(j-l)\cdot sign(j-r)+1), \end{aligned} \tag{4}$$

Typically, a mask is defined by Equation 4 with Θ as parameters, where $m_p^\Theta \in \{0,1\}^{w\times h}$ is the patch mask and $[i,j]$ is index of the pixel at i-th row and j-th column; $sign(x)$ outputs 1 when $x \geq 0$ and -1 when $x < 0$; and $m_p^\Theta[i,j]$

evaluates to one if and only if the pixel is within the four borders defined by Θ and zero otherwise. To make each mask value differentiable regarding border parameters and maintain the property of original definition, we approximate $sign(\cdot)$ by $\tanh(\cdot)$ and define the patch mask with Equation 5.

$$\begin{aligned} m_p^{\Theta}[i,j] = &\frac{1}{4}(-\tanh(i-t)\cdot\tanh(i-b)+1) \\ &\cdot(-\tanh(j-l)\cdot\tanh(j-r)+1) \end{aligned} \tag{5}$$

Fig. 5b is an example of the mask defined by us. In this example, w and h are 30, l and t are 10, and r and b are 20. Observe that the borders of the rectangular region change gradually. Each pixel value is encoded with border parameters Θ.

In the beginning, the patch mask is initialized to cover the whole image, (*i.e.*, $l=t=0$, $b=h$ and $r=w$). One of our optimization goal is to minimize the mask area, thus we define a mask loss term (Equation 6) to penalize the area of mask.

$$\mathcal{L}_m = \frac{r-l+b-t}{w+h} \tag{6}$$

We use a linear combination of the width and height of the rectangular region to avoid bias in the update of edges. Otherwise, if we use the ratio of area (*i.e.*, $(r-l)\times(b-t)/(w\times h)$) as the mask loss, parameters of the longer edge (*e.g.*, b and t when $(b-t)<(r-l)$) would have larger gradients and tend to change faster than the shorter edges, which leads to a bias towards updating the longer-edge parameters. Using a linear combination avoids this problem and each mask parameter has the same weight.

Although we define a rectangular patch region, the final patch mask can be an arbitrary shape within the region. As shown in Fig. 5c, given a predefined patch shape mask m_s ($m_s[i,j]\in\{0,1\}$), the final patch mask $m_p'^{\Theta}$ is calculated by element-wise multiplying the scaled shape mask m_s' with the region mask m_p^{Θ} inside the rectangular region. Specifically, in each iteration, given border parameters Θ, we can scale and fit the predefined shape mask m_s into the center of the rectangular region getting mask m_s' , which is denoted by the red color in Fig. 5c. The final patch mask is calculated with Equation 7 by multiplying the region mask and the shape mask within the rectangular region. Without loss of generality, we focus on rectangular shapes (*i.e.*, $m_s \equiv 1$) in our evaluation.

$$m_p'^{\Theta}[i,j] = \begin{cases} m_p^{\Theta}[i,j] * m_s'[i,j] & i \in l...r, j \in t...b \\ m_p^{\Theta}[i,j] & others \end{cases} \tag{7}$$

In addition, our mask definition also supports optimizing with multiple patches. The key point is to take the union of several regions and optimize them together.

3.4 Attack Camouflage

Patches generated in existing adversarial attacks against depth estimation models have obvious perturbations as shown in Fig. 2a. Unlike them, we use style transfer to camouflage the attack with natural styles. There have been works using style transfer [14] in attacking classification models but we are the first to combine style transfer with the more challenging depth estimation attack. We use deep photo style transfer [29] as our style transfer method. This method is a kind of neural style transfer which has demonstrated remarkable results for image stylization [16]. It uses a convolutional neural network (CNN) to extract the deep features of an image and separate the content and style information in the deep feature representations. The source image will be updated iteratively to approach the style information extracted from the style image and keep the content information of the source image. Specifically, as defined in deep photo style transfer [29], there are four terms regarding the style transfer components in the loss function. They are style loss ($\mathcal{L}_s$), content loss ($\mathcal{L}_c$), smoothness loss ($\mathcal{L}_t$) and photorealism regularization loss ($\mathcal{L}_r$). We refer the readers to [29] for more detailed explanation on each term. The style transfer loss therefore is:

$$\mathcal{L}_{st} = \mathcal{L}_s + \mathcal{L}_c + \mathcal{L}_t + \mathcal{L}_r \tag{8}$$

In summary, our adversarial patch generation process can be formulated by the following optimization problem:

$$\begin{aligned} \min_{x', \Theta} \quad & \mathcal{L}_a + \mathcal{L}_m + \lambda \mathcal{L}_{st} \\ s.t. \quad & x' \in [0, 255]^{3\times w\times h}, \Theta = \{l, r, t, b\} \\ & 0 \leq l \leq r \leq w, \ 0 \leq t \leq b \leq h, \end{aligned} \tag{9}$$

where λ is an adjustable weight parameter to balance the style transfer naturalness and attack performance. The weights of other terms are fixed in our experiments. In each iteration, we calculate gradients of x' and Θ with back propagation and, same as in deep photo style transfer [29], we use LBFGS [8] to update the patch x'. We update border parameters Θ with Adam [22] and we only update the edge with the maximum absolute gradient instead of four, which avoids the constraint of compressing the region from all directions in each iteration and provides more flexibility. We set a target ratio of the patch region in advance (*i.e.*, the area of the patch region relative to the object) as the stopping criteria of mask optimization. In other words, the mask will stop updating when it is smaller than the predefined target ratio.

4 Experiments

4.1 Experimental Setup

MDE Model Selection. In our evaluation, we use three widely known, representative monocular depth estimation models: Monodepth2 [18], Depthhints [54], and Manydepth [53].

Target Object Selection. Our attack is generic so it can be applied to any class of objects on public roads. This paper focuses on three representative types

Code can be found at https://github.com/Bob-cheng/MDE_Attack

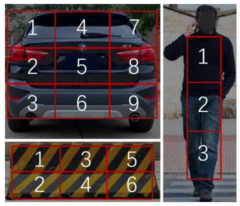


Fig. 6: Target objects and fixed regions.

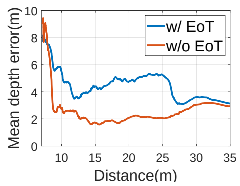


Fig. 7: $\mathcal{E}_d$ of the target vehicle at **different distance**.

Table 1: Mean depth estimation error ($\mathcal{E}_d$) in attacking **fixed regions and optimized regions**.

	Mono			DH			Many		
	V	TB	P	V	TB	P	V	TB	P
Ours	**16.84**	**8.26**	**14.06**	**15.23**	**4.54**	**13.17**	**6.31**	**3.57**	**10.15**
LO	13.90	5.21	11.53	2.51	1.63	10.79	3.03	2.94	8.93
R1	3.70	2.35	10.20	2.25	1.50	11.78	1.12	2.77	9.21
R2	7.41	2.67	11.28	4.66	1.40	10.52	4.23	1.40	8.66
R3	5.20	4.96	5.05	3.92	1.45	4.08	1.33	3.05	5.06
R4	7.31	1.59	-	5.58	1.59	-	4.89	1.59	-
R5	14.95	2.39	-	7.70	0.90	-	5.66	2.43	-
R6	9.69	2.59	-	2.37	0.49	-	1.36	1.15	-
R7	3.23	-	-	2.62	-	-	1.67	-	-
R8	7.74	-	-	4.44	-	-	4.91	-	-
R9	5.36	-	-	1.38	-	-	1.32	-	-

Mono: Monodepth2, DH: DepthHints, Many: Manydepth
V: Vehicle, TB: Traffic Barrier, P: Pedestrian
LO: Location Optimize in [38], R: Region

of objects to attack: vehicles, traffic barriers, and pedestrians as shown in Fig. 6. We choose them because they are most common on public roads in regular driving scenarios, and a failure in detecting them could lead to life-threatening consequences. Vehicles are the most attractive objects for attackers since they are the main targets of perception systems on autonomous driving cars. We mainly focus on vehicles in our experiments.

Evaluation Scene Selection. We select 100 real-world driving scenes from KITTI dataset [17] to evaluate the attack performance of the generated patch on each object in the digital-world. These scenes cover a wide range of roads (*e.g.*, high-way, local, and rural roads) and background objects (*e.g.*, trucks, traffic lights, and cars). Physical-world experiments use three driving routes with various lighting conditions.

Evaluation Metrics. We use mean depth estimation error ($\mathcal{E}_d$) of the target object and ratio of affected region ($\mathcal{R}_a$) as our evaluation metrics. We use depth estimation of the original object as the ground truth and compare with depth estimation of the adversarial object. The mean depth estimation error denotes the attack effectiveness of our adversarial patch. The larger it is, the better the performance. Equation 10 is the formal definition. Meanings of the symbols are the same as those in §3.

$$\mathcal{E}_d = \frac{\mathrm{sum}\left(\left|\mathcal{D}\left(\Lambda(O,R)\right) - \mathcal{D}\left(\Lambda(O',R)\right)\right| \odot M_o\right)}{\mathrm{sum}(M_o)} \tag{10}$$

The ratio of affected region $\mathcal{R}_a$ is defined as:

$$\mathcal{R}_a = \frac{\mathrm{sum}\left(\mathbf{I}\left(\left|\mathcal{D}\left(\Lambda(O,R)\right) - \mathcal{D}\left(\Lambda(O',R)\right)\right| \odot M_o \geq 10\right)\right)}{\mathrm{sum}(M_o)}, \tag{11}$$

where $\mathbf{I}(x)$ is the indicator function that evaluates to 1 only when x is true. We define $\geq$ 10 meters error of depth estimation for a pixel as a valid attack and

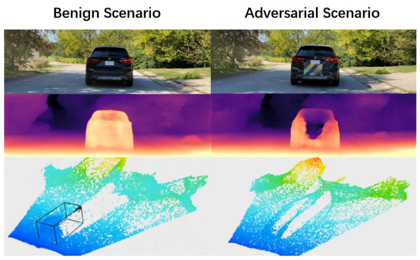


Fig. 8: **Physical world attack** example.

Table 2: **Physical world attack** result.

	Time (s)	Frames	$\mathcal{E}_d$	Detected	Detection Rate
Route 1 Benign	95	477	0.52	469	98.32%
Route 2 Benign	82	412	0.77	354	85.92%
Route 3 Benign	80	402	0.62	348	86.57%
Total Benign	**257**	**1291**	**0.64**	**1171**	**90.70%**
Route 1 Adv.	94	468	6.73	45	9.62%
Route 2 Adv.	82	408	8.92	11	2.70%
Route 3 Adv.	80	402	7.68	10	2.49%
Total Adv.	**256**	**1278**	**7.77**	**66**	**5.16%**

this pixel will be included in the affected region. $\mathcal{R}_a$ is the ratio between the number of affected pixels and all pixels of the object.

4.2 Main results

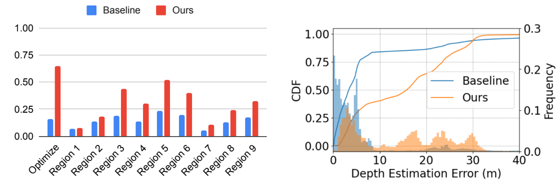
We present our main results regarding effectiveness, robustness and stealth.

Attack Effectiveness. We run our attack with the three MDE models and we target the three types of objects for each model. For each object, we split it into several regions with equal size as shown in Fig. 6 and attack these fixed regions respectively (*i.e.*, optimize the patch on each region.), then we compare with two patch region optimization techniques: our sensitive region localization (§3.3) and the location-optimized patch [38]. In [38], the authors update the location of a fixed-size patch after each optimization iteration. They tentatively move the patch towards four directions with a predefined stride and select the direction with the least adversarial loss as the next patch location. For a fair comparison, we set the target ratio of patch region the same as that of those fixed regions (*e.g.*, 1/9 of the vehicle's read area). Our regional optimization stops when the patch ratio is smaller than the target ratio. In each test, we evaluate the mean depth estimation error ($\mathcal{E}_d$) of the target object in 100 scenes and take the average of them as the result. In each scene, the object is placed at 7 m away from the victim's camera. We choose 7 m since it is the breaking distance [1] while driving at a speed of 25 mph, which is almost the lowest in normal driving. In other words, it is the smallest distance at which the object has to be detected by the victim to avoid a crash in normal driving scenarios [9].

Table 1 reports the effectiveness evaluation result. As shown, our attack is generic and effective on different depth estimation models and objects. With our sensitive region localization, an adversarial patch with 1/9 of the vehicle's rear area causes at least 6 m $\mathcal{E}_d$ across different depth estimation models. Observe that attack performance differs with patch regions. Our sensitive region localization can locate an optimal place that outperforms all those fixed regions and the location optimized regions in [38]. For the physical world experiments, Fig. 8 presents an example. As shown, the adversarial patch on the vehicle fools the vehicle's depth estimation, and the effect is not limited to the patch area but propagates to a broader area. After being projected to 3D space, it is more

(a) Ours (b) Baseline

Fig. 9: **Naturalness** comparison.



(a) $\mathcal{R}_a$ (b) CDF

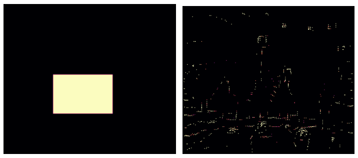
Fig. 10: Comparing **patch-oriented attack** (baseline) with our **object-oriented attack**.

obvious that the point cloud of the adversarial vehicle is distorted comparing with the benign one. Table 2 reports the physical world attack performance. The first column in the table denotes different drives. The second column shows the time of each drive in seconds. The third column shows the total frames evaluated from the video, and we evaluate frames at a frequency of 5 Hz. The fourth column reports the mean depth estimation error ($\mathcal{E}_d$) of the vehicle. As shown, in benign scenarios, the error is under 1 m while the error in adversarial scenarios is over 7 m, which justifies our attack in the physical world.

Attack Robustness. Relative to the victim vehicle, we place the adversarial object at places with longitudinal distances (*i.e.*, forward and back) ranging from 7 m to 35 m and lateral distances (*i.e.*, left and right) ranging from -1 m to 1 m. The 7 m to 35 m longitudinal distance corresponds to the brake distance for driving speed from about 25 to 55 mph [4]. We consider the victim vehicle at the center of the lane, and -1 m to 1 m of lateral deviation from the lane center covers most driving scenarios of the vehicle ahead [13]. We use a vehicle as the target object and Monodepth2 as the depth estimation network. We use the regional optimization and set the target patch size to 1/9 of the vehicle's rear area. We test our attack with and without EoT [6] (see §3.2) during optimization.

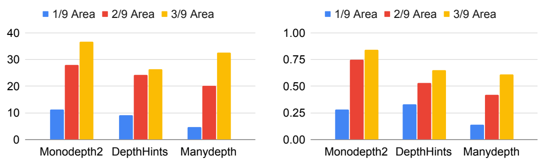
Fig. 7 shows the result of the robustness evaluation. We report the mean depth estimation error of the target object under different longitude distances with the victim vehicle. Observe that our attack is robust and causes more than 3 m of mean depth estimation error in different victim approaching positions. EoT increases the attack performance by 40.63% and makes our attack more robust in different distances. As shown, the closer the target object, the larger the error in depth estimation, which makes the victim vehicle harder to detect the object from the distorted pseudo-Lidar and continue approaching it until collision. In the physical world experiments, our attack is conducted with real driving scenarios. Compared to evaluating with a single image from a specific position in prior work, continuous and dynamic movement is more challenging and practical. Our attack is shown to be robust under different lighting conditions (*e.g.*, shadows and different light directions), driving operations (*e.g.*, moving straight and turning) and background scenes. The dynamic moving video of our physical world attack is at `https://youtu.be/L-SyoAsAM0Y`.

Stealth As we discussed in our motivation, we consider the stealth in two directions: the naturalness of appearance and the patch size. In terms of naturalness,



(a) Regional (b) Per-pixel

Fig. 11: Different **mask optimization methods**.



(a) $\mathcal{E}_d$ (b) $\mathcal{R}_a$

Fig. 12: Attack performance of regional optimization with **different target sizes**.

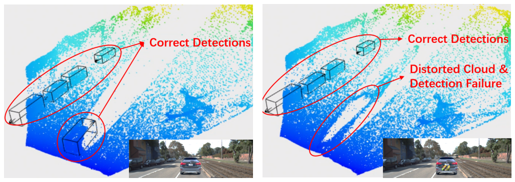
we compare the adversarial patch generated by our method with the baseline method proposed by Yamanaka et al. [63]. As shown in Fig. 9, our method with style transfer-based camouflage generates more natural patches and is less likely to be identified as adversarial but just a normal sticker. Human studies conducted in [29,14] also justify the naturalness of style-transfer-based image processing. As for the patch size, a smaller size suggests more stealth and less effectiveness. We hence investigate maximizing the attack effect with small patches. We compare the $\mathcal{R}_a$ caused by our object-oriented attack and the patch-oriented attack in [63] which only attacks the patch area in their adversarial loss design instead of considering the whole object. For a fair comparison, we use style-transfer-based camouflage in both methods and we test with fixed regions and the regional optimization. This experiment is conducted on Monodepth2 [18] targeting the vehicle and other settings are the same as the previous setup.

As shown in Fig. 10a, our method (object-oriented) has over 2.5 times higher $\mathcal{R}_a$ on the vehicle than the baseline (patch-oriented) in all cases, and our method in the regional optimization case outperforms all other fixed-region cases. Hence, with the same total patch area, our object-oriented attack with regional optimization affects a broader area than the baseline. In other words, *to achieve similar attack effect, using our method requires a smaller patch and is thus stealthier.* Fig. 10b additionally shows the CDF and histogram of depth estimation error in the case with our regional optimization. As shown, more than 80% errors caused by the baseline method are below 10 m, which corresponds to our observation in Fig. 2c that the patch-oriented attack mainly affects the limited patch area and the effect of our method propagates to a broader area causing larger errors.

4.3 Ablation Study

We investigate our method through a set of ablation studies.

Combinations. As detailed in §3, we use the object-oriented adversarial loss design and the regional optimization of the patch mask to maximize the attack effect with a small patch. We conduct ablations on these techniques to see how each component contributes. Table 3 shows the result. We attack Monodepth2 and use the vehicle as the target object and report $\mathcal{E}_d$ and $\mathcal{R}_a$. For those tests without regional optimization, we use #5 fixed region because its attack performance is the best among all the fixed regions in previous evaluations. As shown, the object-oriented adversarial loss itself can improve the attack performance



(a) Benign (b) Adversarial

Fig. 13: Attack against **3D object detection**.

Table 4: **Attack success rate** of different adversarial patches.

	Monodepth2	DepthHints	Manydepth
1/9 Area	95%	93%	98%
2/9 Area	98%	97%	100%
1/3 Area	100%	100%	100%

while the regional optimization cannot. The regional optimization is useful only when object-oriented adversarial loss is applied together. The regional optimization has to consider the whole object to find an optimal place regarding the target object. Since the patch-oriented design does not encode the global information, our regional optimization cannot converge to the most effective region.

Mask Optimization Methods. We compare our regional optimization with another commonly used mask optimization technique which treats pixels of the patch mask m_p as optimizable parameters instead of the four borders. This method has been used in many backdoor scanning works such as Neural Cleanse [51] and ABS [28] to find a trigger that modifies a limited portion of image and causes misclassification (see Fig. 11). Note that the patch mask generated by the baseline method is more sparse and scattered. The patch unit is tiny. Compared with our method, it is not suitable as a physical world attack vector because it is hard to print and deploy these scattered tiny patches.

Table 3: **Ablation study** .

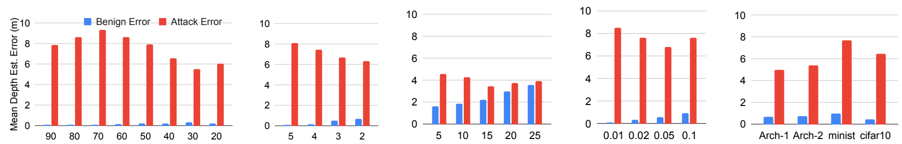
OA	RO	$\mathcal{E}_d$	$\mathcal{R}_a$
		8.47	0.23
	✓	6.38	0.16
✓		14.95	0.52
✓	✓	**16.84**	**0.65**

OA: Object-oriented Adv. Loss
RO: Regional Optimization

Patch Sizes. Larger patches have more effect on depth estimation but are less stealthy. We evaluate our attack on a vehicle object with three different target patch sizes and use three depth estimation models (see Fig. 12). Note that the mean depth estimation error $\mathcal{E}_d$ and the ratio of affected region $\mathcal{R}_a$ increase with the size of patch for all three target networks.

4.4 Downstream Task Impact

We evaluate the impact of our attack on a point cloud based 3D object detection model – PointPillars [24] and use attack success rate (ASR) as the metric to evaluate our method on 3D object detection. We consider the attack is successful when the benign vehicle can be detected by PointPillar while the adversarial object cannot. Fig. 13 gives an example of a successful attack. Fig. 13a presents a benign scenario where the benign vehicle can be correctly detected with a 3D bounding box. Fig. 13b shows the corresponding adversarial scenario where the pseudo-Lidar point cloud of the adversarial vehicle is severely distorted by the patch, and thus the vehicle is not detected. The PointPillar network can correctly detect the benign vehicle in all the 100 scenes and the attack success rate (ASR) of different adversarial patches are reported in Table 4. The first column denotes different patch sizes and columns 2-4 refer to the three different target networks.



(a) JPEG Compression (b) Bit-Depth (c) Median Blur (d) Noise (e) Autoencoder

Fig. 14: Five directly-applicable defence methods. *Benign Error*: Error caused by the defence in benign cases. *Attack Error*: Error caused by our attack.

As shown, the ASR is over 90% with all the patch sizes and target networks. Even when the patch size is just 1/9 of the vehicle's rear area, it can still achieve at least 93% ASR, which shows that our attack is an effective method in fooling the 3D object detection model. In the physical world experiments, the fifth column of Table 2 denotes the number of frames in which the vehicle is detected from the pseudo-Lidar point cloud, and the sixth column reports the object detection rate. For benign cases, the rate of successful object detection is 90.70% in 1291 data frames. The rate drops to 5.16% in adversarial cases with 1278 data frames.

4.5 Defence Discussion

Although many defense techniques against adversarial examples have been proposed, none of them focuses on MDE to the best of our knowledge. As a best effort to understand the performance of our attack under different defences, we apply five popular defence techniques which perform input transformations without retraining the victim network. They are JPEG compression [15], bit-depth reduction [62], median blurring [62], adding Gaussian noise[65] and autoencoder reformation [31]. Fig. 14 presents our results. We report the $\mathcal{E}_d$ of the benign vehicle and the adversarial vehicle under different input transformations. An ideal defence should minimize both errors. As shown, our attack can still cause over 5 meters $\mathcal{E}_d$ in all methods except median blur. In median blur, the attack is mitigated but the benign performance also drops a lot. This shows that these techniques cannot effectively defend our attack. We argue that these defenses are mainly for attacks in digital space [39] instead of physical world settings.

5 Conclusion

In this paper, we investigate stealthy physical-world adversarial patch attack against MDE in the AD scenario. We design a novel physical-object-oriented optimization framework to generate stealthy and effective adversarial patches for attack via an object-oriented adversarial loss design. Experimental results show that our attack is effective, stealthy and robust against different target objects, state-of-the-art models and a representative downstream task (*i.e.*, 3D object detection) in AD.

Acknowledgments. This research was supported, in part by IARPA TrojAI W911NF-19-S-0012, NSF 1901242 and 1910300, ONR N000141712045, N00014-1410468 and N000141712947.

References

1. Break Distance, http://www.csgnetwork.com/stopdistcalc.html
2. Tesla AI day, https://youtu.be/j0z4FweCy4M?t=5295
3. Tesla use per-pixel depth estimation with self-supervised learning, https://youtu.be/hx7BXih7zx8?t=1334
4. Vehicle Stopping Distance, http://www.csgnetwork.com/stopdistcalc.html
5. Aich, S., Vianney, J.M.U., Islam, M.A., Kaur, M., Liu, B.: Bidirectional attention network for monocular depth estimation. arXiv preprint arXiv:2009.00743 (2020)
6. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. In: International conference on machine learning. pp. 284–293. PMLR (2018)
7. Brown, T.B., Mané, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial patch. arXiv preprint arXiv:1712.09665 (2017)
8. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM Journal on scientific computing **16**(5), 1190–1208 (1995)
9. Cao, Y., Wang, N., Xiao, C., Yang, D., Fang, J., Yang, R., Chen, Q.A., Liu, M., Li, B.: 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). pp. 176–194. IEEE (2021)
10. Cao, Y., Xiao, C., Cyr, B., Zhou, Y., Park, W., Rampazzi, S., Chen, Q.A., Fu, K., Mao, Z.M.: Adversarial sensor attack on lidar-based perception in autonomous driving. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. pp. 2267–2281 (2019)
11. Chen, S.T., Cornelius, C., Martin, J., Chau, D.H.P.: Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 52–68. Springer (2018)
12. Cui, Y., Yan, L., Cao, Z., Liu, D.: Tf-blender: Temporal feature blender for video object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8138–8147 (2021)
13. Dominguez, S., Ali, A., Garcia, G., Martinet, P.: Comparison of lateral controllers for autonomous vehicle: Experimental results. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). pp. 1418–1423. IEEE (2016)
14. Duan, R., Ma, X., Wang, Y., Bailey, J., Qin, A.K., Yang, Y.: Adversarial camouflage: Hiding physical-world attacks with natural styles. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1000–1008 (2020)
15. Dziugaite, G.K., Ghahramani, Z., Roy, D.M.: A study of the effect of jpg compression on adversarial images. arXiv preprint arXiv:1608.00853 (2016)
16. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
17. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
18. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth prediction (October 2019)

19. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
20. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
21. Jia, Y., Lu, Y., Shen, J., Chen, Q.A., Zhong, Z., Wei, T.: Fooling detection alone is not enough: First adversarial attack against multiple object tracking. In: International Conference on Learning Representations (ICLR) (2020)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Komkov, S., Petiushko, A.: Advhat: Real-world adversarial attack on arcface face id system. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 819–826. IEEE (2021)
24. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
25. Lee, J., Yi, J., Shin, C., Yoon, S.: Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2643–2652 (2021)
26. Liu, D., Cui, Y., Tan, W., Chen, Y.: Sg-net: Spatial granularity network for one-stage video instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9816–9825 (2021)
27. Liu, D., Cui, Y., Yan, L., Mousas, C., Yang, B., Chen, Y.: Densernet: Weakly supervised visual localization using multi-scale feature aggregation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 6101–6109 (2021)
28. Liu, Y., Lee, W.C., Tao, G., Ma, S., Aafer, Y., Zhang, X.: Abs: Scanning neural networks for back-doors by artificial brain stimulation. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 1265–1282 (2019)
29. Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4990–4998 (2017)
30. Mathew, A., Patra, A.P., Mathew, J.: Monocular depth estimators: Vulnerabilities and attacks. arXiv preprint arXiv:2005.14302 (2020)
31. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. pp. 135–147 (2017)
32. Nassi, B., Nassi, D., Ben-Netanel, R., Mirsky, Y., Drokin, O., Elovici, Y.: Phantom of the adas: Phantom attacks on driver-assistance systems. IACR Cryptol. ePrint Arch. **2020**, 85 (2020)
33. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European symposium on security and privacy (EuroS&P). pp. 372–387. IEEE (2016)
34. Pei, K., Cao, Y., Yang, J., Jana, S.: Deepxplore: Automated whitebox testing of deep learning systems. In: proceedings of the 26th Symposium on Operating Systems Principles. pp. 1–18 (2017)
35. Petit, J., Stottelaar, B., Feiri, M., Kargl, F.: Remote attacks on automated vehicles sensors: Experiments on camera and lidar. Black Hat Europe **11**(2015), 995 (2015)
36. Qiu, H., Xiao, C., Yang, L., Yan, X., Lee, H., Li, B.: Semanticadv: Generating adversarial examples via attribute-conditioned image editing. In: European Conference on Computer Vision. pp. 19–37. Springer (2020)

37. Ranjan, A., Janai, J., Geiger, A., Black, M.J.: Attacking optical flow. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2404–2413 (2019)
38. Rao, S., Stutz, D., Schiele, B.: Adversarial training against location-optimized adversarial patches. In: European Conference on Computer Vision. pp. 429–448. Springer (2020)
39. Sato, T., Shen, J., Wang, N., Jia, Y., Lin, X., Chen, Q.A.: Dirty road can attack: Security of deep learning based automated lane centering under physical-world attack. In: 30th {USENIX} Security Symposium ({USENIX} Security 21). pp. 3309–3326 (2021)
40. Sato, T., Shen, J., Wang, N., Jia, Y.J., Lin, X., Chen, Q.A.: Hold tight and never let go: Security of deep learning based automated lane centering under physical-world attack. arXiv preprint arXiv:2009.06701 (2020)
41. Sato, T., Shen, J., Wang, N., Jia, Y.J., Lin, X., Chen, Q.A.: Wip: Deployability improvement, stealthiness user study, and safety impact assessment on real vehicle for dirty road patch attack. In: Workshop on Automotive and Autonomous Vehicle Security (AutoSec). vol. 2021, p. 25 (2021)
42. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 acm sigsac conference on computer and communications security. pp. 1528–1540 (2016)
43. Shen, J., Won, J.Y., Chen, Z., Chen, Q.A.: Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under {GPS} spoofing. In: 29th {USENIX} Security Symposium ({USENIX} Security 20). pp. 931–948 (2020)
44. Shin, H., Kim, D., Kwon, Y., Kim, Y.: Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In: International Conference on Cryptographic Hardware and Embedded Systems. pp. 445–467. Springer (2017)
45. Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramer, F., Prakash, A., Kohno, T.: Physical adversarial examples for object detectors. In: 12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18) (2018)
46. Tang, K., Shen, J.S., Chen, Q.A.: Fooling perception via location: A case of region-of-interest attacks on traffic light detection in autonomous driving. In: NDSS Workshop on Automotive and Autonomous Vehicle Security (AutoSec) (2021)
47. Thys, S., Van Ranst, W., Goedemé, T.: Fooling automated surveillance cameras: adversarial patches to attack person detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
48. Trippel, T., Weisse, O., Xu, W., Honeyman, P., Fu, K.: Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks. In: 2017 IEEE European symposium on security and privacy (EuroS&P). pp. 3–18. IEEE (2017)
49. Tsai, T., Yang, K., Ho, T.Y., Jin, Y.: Robust adversarial objects against deep learning models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 954–962 (2020)
50. Tu, Y., Lin, Z., Lee, I., Hei, X.: Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors. In: 27th {USENIX} Security Symposium ({USENIX} Security 18). pp. 1545–1562 (2018)

51. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 707–723. IEEE (2019)
52. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: CVPR (2019)
53. Watson, J., Aodha, O.M., Prisacariu, V., Brostow, G., Firman, M.: The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In: Computer Vision and Pattern Recognition (CVPR) (2021)
54. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints. In: The International Conference on Computer Vision (ICCV) (October 2019)
55. Wimbauer, F., Yang, N., von Stumberg, L., Zeller, N., Cremers, D.: Monorec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
56. Wong, A., Cicek, S., Soatto, S.: Targeted adversarial perturbations for monocular depth prediction. arXiv preprint arXiv:2006.08602 (2020)
57. Xiao, C., Li, B., Zhu, J.Y., He, W., Liu, M., Song, D.: Generating adversarial examples with adversarial networks. arXiv preprint arXiv:1801.02610 (2018)
58. Xiao, C., Pan, X., He, W., Peng, J., Sun, M., Yi, J., Liu, M., Li, B., Song, D.: Characterizing attacks on deep reinforcement learning. arXiv preprint arXiv:1907.09470 (2019)
59. Xiao, C., Yang, D., Li, B., Deng, J., Liu, M.: Meshadv: Adversarial meshes for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6898–6907 (2019)
60. Xiao, C., Zhu, J.Y., Li, B., He, W., Liu, M., Song, D.: Spatially transformed adversarial examples. arXiv preprint arXiv:1801.02612 (2018)
61. Xu, K., Zhang, G., Liu, S., Fan, Q., Sun, M., Chen, H., Chen, P.Y., Wang, Y., Lin, X.: Adversarial t-shirt! evading person detectors in a physical world. In: European Conference on Computer Vision. pp. 665–681. Springer (2020)
62. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)
63. Yamanaka, K., Matsumoto, R., Takahashi, K., Fujii, T.: Adversarial patch attacks on monocular depth estimation networks. IEEE Access **8**, 179094–179104 (2020)
64. Yan, C., Xu, W., Liu, J.: Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. Def Con **24**(8), 109 (2016)
65. Zhang, Y., Liang, P.: Defending against whitebox adversarial attacks via randomized discretization. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 684–693. PMLR (2019)
66. Zhang, Z., Zhu, X., Li, Y., Chen, X., Guo, Y.: Adversarial attacks on monocular depth estimation. arXiv preprint arXiv:2003.10315 (2020)