# Physical 3D Adversarial Attacks against Monocular Depth Estimation in Autonomous Driving

Junhao Zheng, Chenhao Lin*, Jiahao Sun, Zhengyu Zhao, Qian Li, Chao Shen*
Xi'an Jiaotong University, Xi'an, 710049, China
{2193412684@stu.,linchenhao@,sunjiahao@stu.,zhengyu.zhao@,qianlix@,chaoshen@mail.}xjtu.edu.cn

## Abstract

*Deep learning-based monocular depth estimation (MDE), extensively applied in autonomous driving, is known to be vulnerable to adversarial attacks. Previous physical attacks against MDE models rely on 2D adversarial patches, so they only affect a small, localized region in the MDE map but fail under various viewpoints. To address these limitations, we propose 3D Depth Fool (3D$^2$Fool), the first 3D texture-based adversarial attack against MDE models. 3D$^2$Fool is specifically optimized to generate 3D adversarial textures agnostic to model types of vehicles and to have improved robustness in bad weather conditions, such as rain and fog. Experimental results validate the superior performance of our 3D$^2$Fool across various scenarios, including vehicles, MDE models, weather conditions, and viewpoints. Real-world experiments with printed 3D textures on physical vehicle models further demonstrate that our 3D$^2$Fool can cause an MDE error of over 10 meters. The code is available at https://github.com/Gandolfczjh/3D2Fool.*

## 1. Introduction

Monocular depth estimation (MDE), i.e., predicting the distance from the camera to each pixel in an image, is a key task in computer vision. This technology finds extensive use in real-world scenarios, such as robot navigation [9, 10, 31] and autonomous driving [25]. The development of deep neural networks (DNNs) has significantly enhanced MDE performance, making it an effective alternative to traditional RGB-D camera-based and Lidar-based depth estimation methods [12, 14, 20, 21, 28, 29, 38, 39]. Leading players in the self-driving vehicle industry, such as Tesla, have been exploring the integration of MDE into their production-grade autopilot systems [1, 2], which leverage low-cost cameras and advanced autonomous driving.

Despite the effectiveness of DNNs, recent studies have demonstrated their vulnerability to adversarial attacks [6, 15], which also poses a practical threat to DNN-based MDE [7, 8, 16, 17, 40, 42]. There are two main types of adversarial attacks: digital [6, 15, 43, 45, 46] and physical [5, 32–34, 36, 37, 41, 44] attacks. Digital attacks involve adding small perturbations to image pixels, and their success is hard to directly translate into the physical world due to their sensitivity to physical transformations, such as printing, weather conditions, and viewpoint changes [32]. Physical attacks address these limitations by optimizing the perturbations under various physical constraints, and they have shown success in misleading real-world autonomous driving systems [32, 33, 36, 37, 44].

In physical-world attacks, the attacker designs a 2D adversarial patch [5, 13, 34] or 3D camouflage texture [32, 33, 36, 37, 44] and pastes it to the target vehicle, which will be captured by cameras and then fed to the victim model. A 2D adversarial patch is pasted on only a small local planar part of the object's surface, failing to achieve adversarial effects at different viewing angles and distances. In contrast,

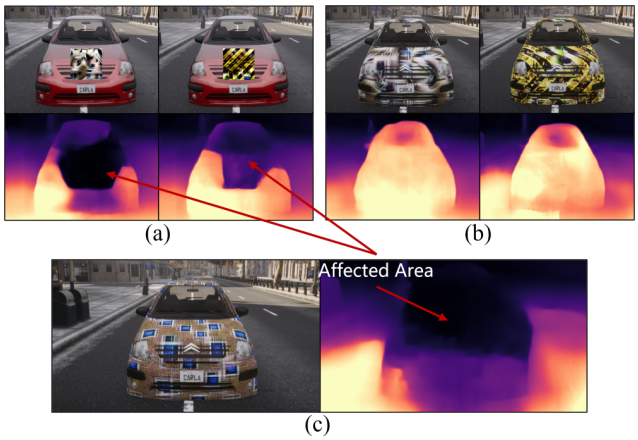


Figure 1. (a) Existing 2D adversarial patch-based attacks [8, 16] and (b) their modified versions with 3D adversarial textures fail to completely remove the vehicle from the MDE map, while (c) our 3D$^2$Fool with robust 3D adversarial textures makes the car vanish.

*Corresponding authors

a 3D camouflage texture is crafted to cover the entire surface of the vehicle, leading to a better attack performance regardless of the viewpoint.

However, existing physical-world attacks in autonomous driving have been mainly focused on object detection [32, 33] with only a few on MDE. Moreover, all existing attacks against MDE are based on 2D adversarial patches [8, 16, 17, 42], which are inevitably limited in challenging conditions with various angles and distances. In this work, we propose 3D Depth Fool (3D$^2$Fool), the first 3D adversarial camouflage attack against MDE models. 3D$^2$Fool generates robust camouflage texture applicable to a wide range of target vehicles regardless of the viewpoint changes. Moreover, beyond those similar studies on object detection, we further simulate weather conditions during attack optimization to achieve improved attack performance in bad weather.

The optimization of 3D$^2$Fool consists of two main modules: texture conversion (TC) and physical augmentation (PA). First, TC converts the 2D adversarial texture seed into the 3D camouflage texture pasted onto the full surface of the vehicle. In particular, TC is independent of the object-specific UV map, so it can be directly applied to various types of target objects, such as cars, buses, and even pedestrians. Second, PA places the rendered 3D vehicle (with textures) into different scenes to obtain photo-realistic images. In particular, we add noise and perturb local image regions to simulate various weather conditions, such as extreme brightness and fog. This improves the robustness of our 3D$^2$Fool in bad weather. Figure 1 shows that our 3D texture-based 3D$^2$Fool makes the vehicle under attack vanish from the MDE map completely, while other 2D patch-based methods only affect a small region of the vehicle.

Our main contributions can be summarized as follows:

- We propose 3D Depth Fool (3D$^2$Fool), the first 3D adversarial camouflage attack against MDE models. 3D$^2$Fool can be applied to a wide range of target vehicles (and even pedestrians) under various physical constraints, such as viewpoint changes.
- We design a new module called texture conversion in 3D$^2$Fool, to generate object-agnostic 3D camouflage textures, by optimizing the 2D adversarial texture seed independent of the vehicle-specific UV map.
- We design a new module called physical augmentation in 3D$^2$Fool, to improve the robustness of 3D$^2$Fool under various weather conditions, by integrating weather-related data augmentation into the attack optimization.

## 2. Related Work

### 2.1. Monocular Depth Estimation

Monocular depth estimation (MDE) plays an important role in perceiving environmental information from 2D images. Eigen *et al.* [12] first utilized deep neural networks to predict depth estimation. Monodepth2 [14] greatly improved the performance through self-supervised learning and multi-scale loss function. RobustDepth [29] improved the robustness and performance of MDE by employing data augmentation to simulate different weather conditions. In this work, we propose 3D$^2$Fool, a new attack that is shown to be effective against various widely-used MDE methods.

### 2.2. Physical Adversarial Attacks

Physical adversarial attacks have been extensively studied. Conventional works [5, 13, 23, 24, 34] rely on adversarial patches with only digital-space constraints, making it hard to achieve effective attacks in the complex, physical world. Later studies [32, 33, 36, 37, 44] propose 3D texture-based attacks to improve the robustness by painting the texture onto the nonplanar surface of vehicles. Specifically, Dual Attention Suppression (DAS) [37] suppresses both model and human attention based on differentiable rendering [18]. Differentiable Transformation Attack (DTA) [32] designs a novel differentiable transformation network to reflect various real-world characteristics and complex scenes.

**Physical adversarial attacks against MDE.** Early studies on attacking MDE [7, 40] rely on conventional and image-level perturbations, known to be ineffective in the physical world. Recent works [8, 16] improve them by instead relying on printable adversarial patches. Specifically, Stealthy and Physical-Object-Oriented (SPOO) [8] restricts the patch to be small and leverages style transfer [22] to further improve the stealthiness. Adaptive Adversarial (APARATE) selectively fools MDE by corrupting the estimated distance and manifesting an object into disappearing. Different from 2D patch-based attacks, our attack, 3D$^2$Fool, is the first 3D texture-based attack against MDE, leading to state-of-the-art attack performance under various viewpoint changes. Moreover, 3D$^2$Fool is designed to be robust to weather changes and applicable to multiple target vehicles.

## 3. Methodology

### 3.1. Problem definition

The problem we need to solve is to cover the entire surface of a vehicle with adversarial texture to attack MDE in the physical world regardless of viewpoints, under various weather, such as rain, fog, etc.

To realize an adversarial camouflage attack, we repeat the 2D adversarial texture seed $t_{\mathrm{s}}$ as a whole camouflage texture to paint on the vehicle's surface. To make the seed suitable for different kinds of objects, we use the texture conversion TC (introduced in Section 3.2) to eliminate the influence of object-specific UV maps. We can calculate the final 2D adversarial texture $t_{\mathrm{adv}}$ by the following:

$$t_{\mathrm{adv}} = \mathrm{TC}(t_{\mathrm{s}}) \tag{1}$$

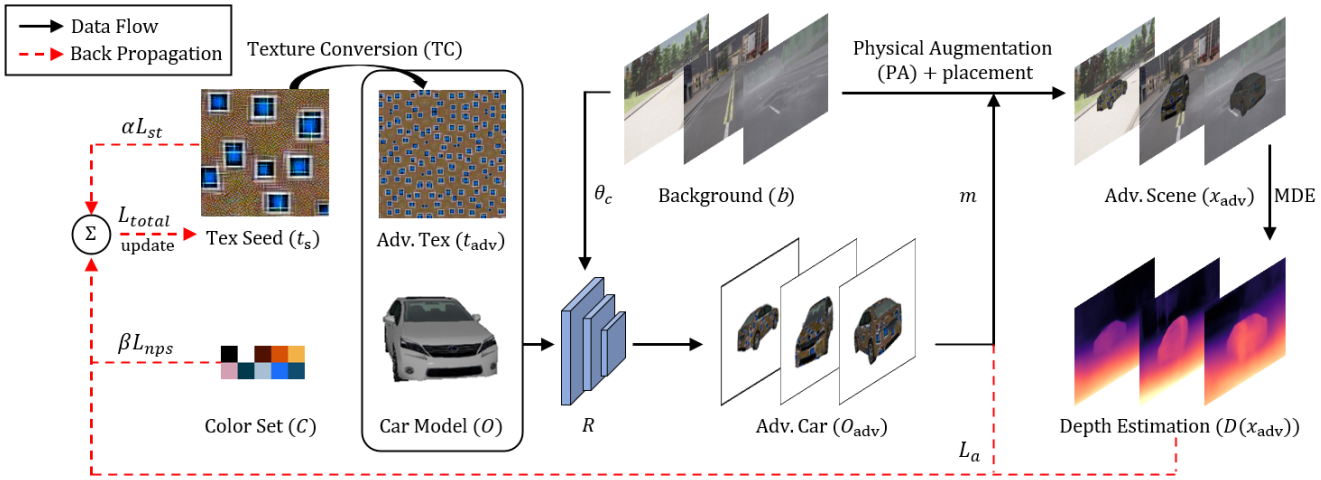


Figure 2. Overview of our 3D$^2$Fool attack against MDE models. 3D$^2$Fool optimizes the adversarial texture seed $t_s$ via backpropagation using $L_{\text{total}}$ through our new texture conversion (TC) and physical augmentation (PA) modules.

We apply the adversarial texture on the surface of a 3D object $O$ and project it to the camouflaged 2D vehicle image with a differentiable renderer [18]. Then we can obtain the adversarial image $x_{\text{adv}}$ through the physical augmentation PA (introduced in Section 3.2), mimicking the natural lightness and weather conditions. We define $R$ as the renderer and the adversarial image is expressed as follows:

$$x_{\text{adv}} = \text{PA}(R(t_{\text{adv}}, O; \theta_{\text{c}}), b, m) \tag{2}$$

where $\theta_{\text{c}}$ is the camera parameters (i.e., transformation and location) required for rendering, $b$ is the road background image and $m$ is the object mask in the scenario.

$D(\cdot)$ is a hypothesis function for the monocular depth estimation model, and $x$ denotes a 2D image input where the vehicle may be with a benign or adversarial texture. We can obtain the prediction $d = D(x)$, where $d$ as the output denotes the depth estimation map. The goal of our proposed method is to make MDE mispredict the depth of the target vehicle, visually making the vehicle vanish, by modifying the surface texture of the target vehicle, which satisfies $d_{\text{t}} = D(x_{\text{adv}})$. The notion $d_{\text{t}}$ represents the target depth we expect MDE to predict, and $x_{\text{adv}}$ denotes the 2D image where the target vehicle is covered with an adversarial texture. Suppose $L(D(x), d)$ is the loss function applied to $D(\cdot)$ that makes the depth estimation of input image $x$ close to $d$. So we can ultimately obtain the adversarial texture seed by solving Equation (3):

$$t_{\text{s}} = \arg\min L(D(x_{\text{adv}}), d_{\text{t}}) \tag{3}$$

### 3.2. Generating Adversarial Texture

To generate robust and effective adversarial texture, we propose the 3D adversarial camouflage attack framework, illustrated in Figure 2. Our training set $(B, M, \Theta_{\text{c}})$ is sampled under different camera parameters and environment settings from Carla [11], a photo-realistic simulator. 3D$^2$Fool first converts the adversarial texture seed $t_{\text{s}}$ to the adversarial texture $t_{\text{adv}}$ through our new texture conversion (TC) module. Then, it renders the camouflage texture onto the vehicle's surface with the same camera parameters to obtain the camouflaged 2D vehicle $O_{\text{adv}}$. Next, it transfers the camouflaged vehicle into different physical scenarios through our new physical augmentation (PA) module. The adversarial texture seed is optimized via backpropagation with the total loss function $L_{\text{total}}$ (introduced below).

**Texture Conversion.** Since the texture UV map is specialized for different vehicles, it is difficult to directly apply camouflage attack to other vehicles. Inspired by [32], we propose the Texture Conversion (TC) to convert the adversarial texture seed to the vehicle texture in a repetitive manner, which is beneficial to generating object-agnostic adversarial textures.

Different from the repeated texture projection in [32], it uses a 3D rotation operation to convert the repeated pattern to the 3D view, based on the same camera pose as the projection of the target vehicle, which would lose the original details of the car surface. So the Differential Transformation Network (DTN) [32] is employed to simulate the surface details. In contrast, we can adjust the region where the adversarial texture can be pasted and only need to employ a differentiable renderer [18] to obtain the accurately rendered 2D vehicle texture without distortion.

The conversion process is shown in Figure 3. We define an $n \times n$ adversarial texture seed by $t_{\text{s}}$. First, we add a variety of random transformations to improve the adversarial texture robustness, such as rotation, flipping, etc. Then we augment the transformed texture seed to the size of $\tau n \times \tau n$,

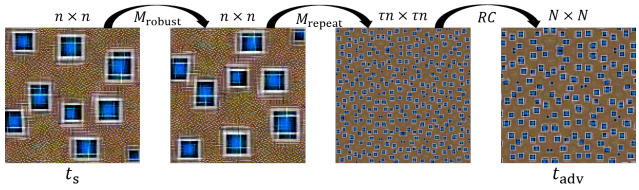


Figure 3. The initial texture seed is transferred into a specified-size texture through the texture conversion module.

where $\tau$ denotes the magnification, and concatenate multiple texture seeds at the edges. To resist the impact of object-specific texture UV map, we use random clipping to obtain the final texture, which can directly replace the original texture to cover the target vehicle through a mask. The above transformations can be expressed as Equation (4):

$$t_{\text{adv}} = \text{RC}(M_{\text{repeat}} \cdot M_{\text{robust}} \cdot t_{\text{s}}) \tag{4}$$

where $M_{\text{robust}}$ is a matrix representing the composition of the random transformations, $M_{\text{repeat}}$ is the repeat operation, and $\text{RC}(\cdot)$ is the random clipping function to crop the texture to the specified size of $N \times N$.

**Physical Augmentation.** The Expectation over Transformation (EoT) [4] is not enough to resist the impact of bad weather conditions. Inspired by the weather rendering transformation such as [27, 35], we propose the Physical Augmentation (PA) to bridge the gap between simulation and physical environment, improving the attack robustness under various weathers. Specifically, we add data augmentation perturbations to the 2D rendered vehicle image $O_{\text{adv}}$, such as exposure, shadow, rain noise, and fog noise. Refer to Automold [3], we employ a polynomial model for generating an exposure or shadow mask and utilize a Gaussian blur to smooth the border for naturalness. Let $\theta_{\text{b}}$ denote the pixel area where the exposure or shadow is placed, by modifying each-pixel intensity to simulate the effect of natural lighting. Similar to physics-based renderer (PBR) [35], we utilize prior information from depth maps to create realistic rain and fog augmentations. Let $\theta_{\text{r}}$ and $\theta_{\text{f}}$ denote the pixel-wise rain and fog noise matrices, respectively. Finally, we apply EoT by randomly transforming the rendered texture in saturation, etc., expressed as $\text{EoT}(\cdot)$. Thus, PA can be summarized as Equation (5):

$$O_{\text{adv\_p}} = \text{EoT}(\theta_{\text{b}} \odot O_{\text{adv}} + \theta_{\text{r}} + \theta_{\text{f}}) \tag{5}$$

where $O_{\text{adv\_p}}$ is the adversarial 2D texture after PA, which can be directly placed in natural scenarios in training set through a mask $m$. We can get the final adversarial images by Equation (6):

$$x_{\text{adv}} = b \odot (1 - m) + O_{\text{adv\_p}} \odot m \tag{6}$$

**Vehicle Vanishing Loss.** Our goal is to make the vehicle with the adversarial texture disappear from the perspective of the MDE, that is, the depth estimation map is wrong in the area of the target vehicle. In previous works attacking MDE with adversarial patches, a mask was usually used to cover the object area and compute the mean squared error between prediction depth and target depth as a loss function. We observe that the patches obtained by previous methods can only successfully affect part of the target vehicle, which is not enough to make the whole object vanish. To overcome the limitation, we cover the adversarial texture over the full surface of the target vehicle. Our loss function is:

$$L_{\text{a}} = \text{MSE}(D(x_{\text{adv}})^{-1} \odot m, 0) \tag{7}$$

where $\text{MSE}(\cdot, \cdot)$ is the mean square error between two variables, and the mask $m$ covers the whole area of the target vehicle. Our goal is to minimize $L_{\text{a}}$ so that we optimize the adversarial texture seed.

**Smooth Loss.** To ensure the naturalness of the generated adversarial texture seed, we utilize a smooth loss (i.e., Total Variation loss [26]) to reduce the inconsistency among adjacent pixels. For the adversarial texture seed $t_{\text{s}}$, the smooth loss can be calculated as:

$$L_{\text{st}} = \sum_{i,j} \sqrt{(t_{\text{s}}^{i,j} - t_{\text{s}}^{i+1,j})^2 + (t_{\text{s}}^{i,j} - t_{\text{s}}^{i,j+1})^2} \tag{8}$$

where $t_{\text{s}}^{i,j}$ is the pixel value of $t_{\text{s}}$ at coordinate $(i, j)$.

**NPS Loss.** To attack MDE in the physical world, the printability of the adversarial texture seed by the printer is necessary. We utilize Non-Printability Score (NPS) [30] loss to regulate the object texture color set. Meanwhile, considering the stealthiness, we only randomly select 10 colors. For the adversarial texture seed $t_{\text{s}}$, the NPS loss can be calculated as:

$$L_{\text{nps}} = \frac{1}{n \times n} \sum_{i,j} \min_{c \in C} |c - t_{\text{s}}^{i,j}| \tag{9}$$

where $n$ is the side length of the texture seed as a scale factor and $C$ is the object texture color set. Finally, our total loss, $L_{\text{total}}$, is constructed as Equation (10):

$$L_{\text{total}} = L_{\text{a}} + \alpha L_{\text{st}} + \beta L_{\text{nps}} \tag{10}$$

where $\alpha$ and $\beta$ are the weights to control the contribution of each loss function. Algorithm 1 summarizes our 3D$^2$Fool against MDE models.

## 4. Experiments

In this section, we first describe the experimental settings. Then we conduct comprehensive experiments to investigate the performance of our proposed method in multiple aspects and compare it with previous works.

**Algorithm 1** 3D$^2$Fool against MDE

**Input:** Car model $O$, Texture Conversion module TC, Physical Augmentation module PA, Color set $C$, Neural renderer $R$, Training set $(B, M, \Theta_c)$, MDE model $D(\cdot)$, number of training iterations K

**Output:** Adversarial Texture Seed $t_s$

```
Initial t_s with random noise
for k ← 1 to K do
    Sample minibatch b ∈ B, m ∈ M, θ_c ∈ Θ_c
    t_adv = TC(t_s)
    O_adv = R(t_adv, O; θ_c)
    x_adv = PA(O_adv, b, m)
    Calculate L_total by Equation (10)
    Update t_s based on gradients of L_total
end for
return t_s
```

### 4.1. Implementation Details

**MDE Model Selection.** In our experiments, we use four MDE models: the Monodepth2 [14], Depthhints [38], Manydepth [39], and Robustdepth [29]. The first three models are chosen regarding [8], while Robustdepth is chosen because of its robustness to severe weather.

**Datasets.** For the adversarial texture training, we randomly select 210 spawn locations and capture the background pictures from Carla [11], including urban roads, highways, country roads, etc., in different weather environments. For the locations used to place the vehicle, we take 8400 images with the RGB camera sensor in Carla at a random angle of 0-360° and within a distance range of 3-15m. For attack evaluation, we overlay the generated adversarial texture on four kinds of vehicles and other objects common in autonomous driving scenarios such as buses and pedestrians, by world-aligned texture function in Unreal Engine, ignoring the specific texture UV map of each vehicle, and collect 6124 images in total. The camera positions are chosen in the same way as for the training set. In addition, to evaluate the attack performance under severe weather, we choose four kinds of weather: foggy, rainy, sunny, and cloudy. For the experiment in the physical world, we use the Tesla car model instead of the real vehicle.

**Evaluation Metrics.** To evaluate the performance of our proposed attack, we use the mean depth estimation error $E_d$ of the target object and the ratio of the affected region $R_a$ [8]. $E_d$ represents the average of the differences between the depth prediction of the adversarial vehicles and the benign vehicles. The larger it is, the better the performance, the same for the $R_a$ metric. When the depth estimation error of a pixel location exceeds a certain threshold, the attack is considered to be successful. Therefore, $R_a$ represents the proportion of pixels where the attack is successful over the whole target vehicle area. Given the difference $\triangle d = |D(x_{adv}) - D(x_{benign})|$, $E_d$ can be expressed as:

$$E_d = \frac{sum(\triangle d \odot m)}{sum(m)} \tag{11}$$

The ratio of the affected region $R_a$ can be represented by:

$$R_a = \frac{sum(I(\triangle d \odot m \geq V_{thre}))}{sum(m)} \tag{12}$$

where $I(x)$ is the indicator function that evaluates to 1 only when $x$ is true. We choose the threshold of 10, i.e., when the depth estimation error exceeds 10, the attack is considered successful and the affected region is counted.

**Compared Methods.** We compare our adversarial camouflage attack with previous works [8, 16, 17, 42] in the physical domain against MDE models. SPOO [8] and APARATE [16] are object-oriented methods for attacking the target objects with patches on them, while Adversarial Patches Attack(APA) [42] and Stealthy Adversarial Attack (SAAM) [17] are patch-oriented methods that affect the local patch area independent of objects. To conduct a fair comparison, we retrain the above methods on our training set and then overlay their generated patches and a random color texture on the target vehicles in the same way.

**Attack Parameters.** Our adversarial texture is optimized using Adam [19] with 10 epochs. For EoT, we use 0.2 random brightness, [0.9, 1.1] random contrast, etc. For texture conversion, we use a random vertical or horizontal flip and a rotation of $\pm 90°$ with 0.5 probability. We set the initial size of the adversarial patch $n = 128$, repetition parameter $\tau = 6$, and the size after random clipping $N = 512$. For loss hyperparameters, we use $\alpha = 0.1, \beta = 5$ as default.

### 4.2. Main results

**Attack Effectiveness.** We run our adversarial camouflage attack and previous attack methods on the four MDE models, and we target the four types of vehicles for each model. For each type of vehicle, instead of a specific texture UV map, we apply a world-aligned texture [33] to the surface. Then we use the above four MDE models to predict the depth of the target vehicles with the adversarial texture to evaluate the performance. As shown in Table 1, our method consistently has the best attack performance on all models. In the test of full texture coverage, the patch-oriented and object-oriented methods achieve similar effectiveness. Among them, SPOO suffers the degradation in performance for naturalness. Compared with the patch attack methods shown in the original papers, camouflage attacks by pasting the generated patches on the vehicle's surface get poorer results. Beyond insufficient adaptability to complicated weather conditions in prior approaches, significant degradation arises due to the warping and deformation

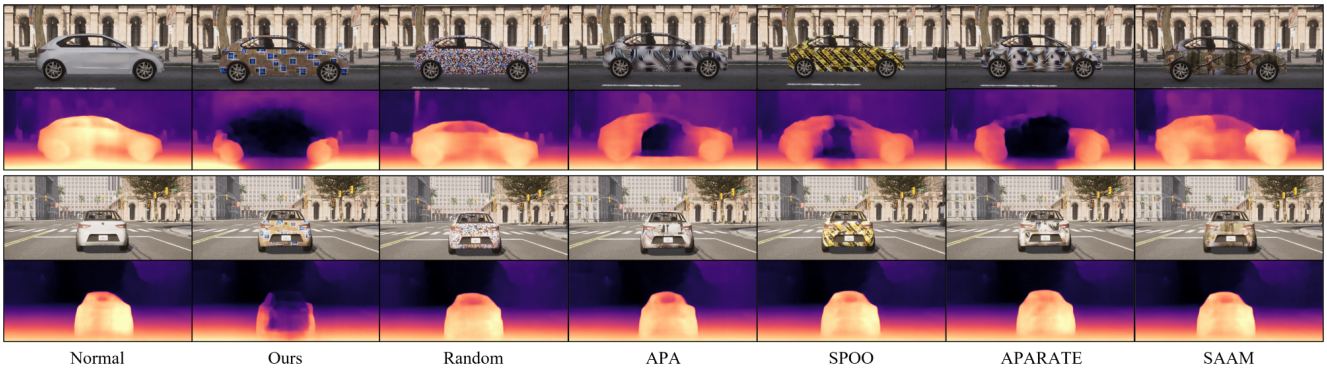


Figure 4. Comparison of our attack and other attacks in Carla simulation. The first column shows the normal vehicle and the rest columns show the vehicles covered with adversarial textures achieved by different attacks. The depth estimation map is generated by Monodepth2.

Table 1. Comparison of our attack and other attacks in Carla simulation regarding the mean depth estimation error $E_{\mathrm{d}}$ and the ratio of the affected region $R_{\mathrm{a}}$.

| Methods | Monodepth2 | | Depthhints | | Manydepth | | Robustdepth | |
|---|---|---|---|---|---|---|---|---|
| | $E_{\mathrm{d}}$ | $R_{\mathrm{a}}$ | $E_{\mathrm{d}}$ | $R_{\mathrm{a}}$ | $E_{\mathrm{d}}$ | $R_{\mathrm{a}}$ | $E_{\mathrm{d}}$ | $R_{\mathrm{a}}$ |
| Normal | 1.25 | 0.019 | 1.12 | 0.016 | 0.82 | 0.006 | 0.13 | 0.000 |
| Random | 3.07 | 0.076 | 2.83 | 0.041 | 1.09 | 0.011 | 0.24 | 0.000 |
| APA [42] | 6.14 | 0.223 | 5.79 | 0.199 | 2.62 | 0.031 | 0.83 | 0.005 |
| SPOO [8] | 5.62 | 0.194 | 4.78 | 0.144 | 2.20 | 0.023 | 0.51 | 0.002 |
| APARATE [16] | 6.88 | 0.265 | 6.05 | 0.213 | 3.13 | 0.097 | 0.95 | 0.007 |
| SAAM [17] | 2.82 | 0.034 | 2.21 | 0.025 | 1.01 | 0.015 | 0.31 | 0.000 |
| Ours | **12.75** | **0.496** | **10.31** | **0.413** | **6.78** | **0.25** | **2.24** | **0.032** |

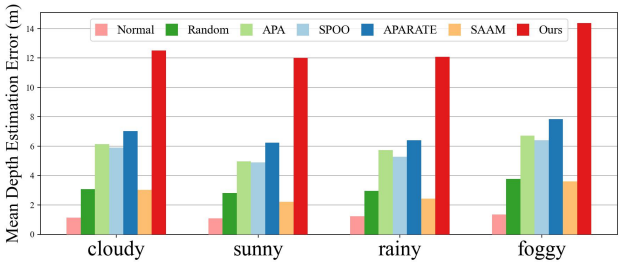


Figure 5. Attack comparison under different weather conditions.

of their patches. In contrast, our method maintains attack performance from the texture seed to camouflage texture, thanks to our proposed texture conversion. Illustrated in Figure 4, our adversarial attack affects the depth estimation of almost the entire target vehicle, effectively causing it to vanish from the perspective of MDE models.

For the attack performance on different models, it is noteworthy that both Monodepth2 and Depthhints exhibit substantial vulnerability to adversarial attacks, while Robustdepth emerges as the most robust among the four MDE models. This is understandable because Robustdepth applies data augmentation to simulate adverse weather conditions during the training phase.

**Attack Robustness.** To evaluate the resistance to severe weather conditions, we conducted the experiments under multiple weather in Carla. In the context of autonomous driving scenarios, we have chosen four prevalent weather conditions, namely cloudy, sunny, rainy, and foggy. Cloudy conditions represent normal weather conditions with sufficient and suitable lighting. Figure 5 shows the impact of different weather conditions on the attacks. Our proposed method demonstrates superior performance compared to previous works under both benign and adverse weather conditions. Prior attacks exhibit performance degradation under sunny and rainy weather conditions compared to the baseline cloudy conditions. For instance, APA experiences a decline of 19.1% and 6.7% under sunny and rainy weather, respectively. In contrast, our proposed attack method demonstrates comparatively marginal decreases, registering reductions of only 3.9% and 3.4%.

Figure 6 shows the attack effect of our camouflage texture on three kinds of vehicles under four weather conditions. At different viewing angles and distances, our method maintains a good attack performance, which can effectively fool the depth estimation of the target vehicle by MDE, almost not affected by the changing weather conditions. In addition to the negative impact of extreme brightness and rain on attacks, an intriguing observation emerges in foggy weather, wherein attacks manifest an unexpectedly positive impact. This phenomenon is attributed to the inherent susceptibility of Monodepth2. Even in the absence of adversarial attacks, model performance is affected in foggy weather.

To evaluate the attack robustness, we randomly set the camera position at different viewpoints of the target vehicles. Figure 7 shows the attack comparison with the mean depth estimation error of Monodepth2 at different viewpoints, including viewing distance and angles. Our attack has a better attack performance under any observation angle

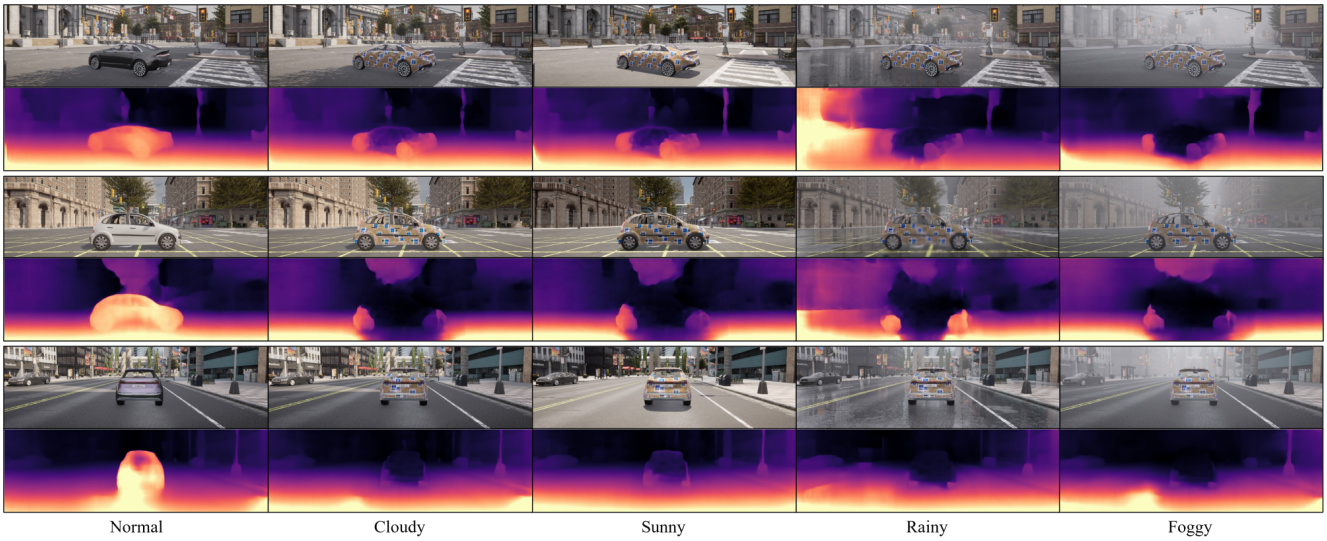


Figure 6. Evaluation with various target vehicles under different weather conditions. The first column shows the normal vehicle and the rest columns show the vehicles covered with our adversarial textures. The depth estimation map is generated by Monodepth2.

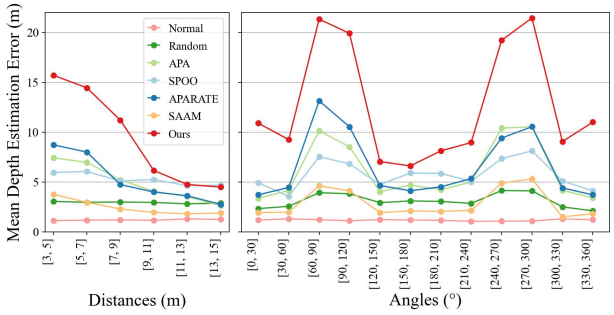


Figure 7. Attack comparison at different viewpoints, including viewing distance and angles. Values are the mean depth estimation error $E_{\mathrm{d}}$ of Monodepth2.

and distance than previous methods. In the range of 60-120° and 230-300°, the mean depth estimation error our attack achieved is more than 20m. At certain viewing angles, the attack effect is affected because of the small coverage area of the adversarial texture and the excessively sloping surface of the target vehicle. Most of the methods have good attack effects within 9m, and with the increase of distance, the attack performance gradually decreases. In particular, the performance of SPOO is relatively stable under different distances. We believe that it is related to the style transfer it adopts, which represents the perturbation of adversarial attacks by the overall style feature.

To evaluate the textures when applied to different objects common in autonomous driving scenarios, we handle trucks, buses, and pedestrians with the same parameters and texture seed, against Monodepth2. As shown in Table 2, our attack surpasses others across diverse objects, signifying its object-agnostic property. As demonstrated in Figure 8, the adversarial textures pasted on the buses and pedestrians severely affect the depth prediction of Monodepth2.

Table 2. Attack comparison on diverse objects. Values are the mean depth estimation error $E_{\mathrm{d}}$ of Monodepth2.

| Methods | truck | | bus | | pedestrian | |
|---|---|---|---|---|---|---|
| | $E_{\mathrm{d}}$ | $R_{\mathrm{a}}$ | $E_{\mathrm{d}}$ | $R_{\mathrm{a}}$ | $E_{\mathrm{d}}$ | $R_{\mathrm{a}}$ |
| Normal | 1.11 | 0.015 | 1.06 | 0.013 | 1.29 | 0.022 |
| Random | 3.12 | 0.079 | 3.09 | 0.076 | 2.59 | 0.057 |
| APA [42] | 6.93 | 0.231 | 7.05 | 0.242 | 3.76 | 0.137 |
| SPOO [8] | 6.21 | 0.214 | 6.77 | 0.226 | 3.22 | 0.126 |
| APARATE [16] | 7.04 | 0.282 | 7.29 | 0.304 | 3.71 | 0.144 |
| SAAM [17] | 2.90 | 0.067 | 3.13 | 0.103 | 1.91 | 0.038 |
| Ours | **14.11** | **0.576** | **15.07** | **0.614** | **8.37** | **0.353** |

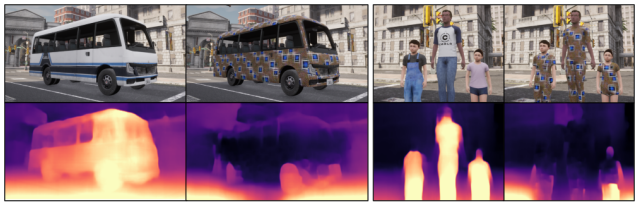


Figure 8. Our attack on different objects in Carla simulation.

**Attack in the Real World.** As for the real-world attack, we conduct several experiments to validate the practical effectiveness of our generated adversarial texture. Specifically, we print and paste our adversarial texture on a 1:24 scaled Tesla Model Y car, and place it under different back-

Table 3. Physical-world evaluation using two scaled car models in different scenes. Values are $E_{\mathrm{d}}$ and $R_{\mathrm{a}}$ of Monodepth2.

| Methods | outdoor | | indoor | |
|---|---|---|---|---|
| | $E_{\mathrm{d}}$ | $R_{\mathrm{a}}$ | $E_{\mathrm{d}}$ | $R_{\mathrm{a}}$ |
| Normal | 1.05 | 0.014 | 1.22 | 0.023 |
| Ours | **10.21** | **0.403** | **10.67** | **0.424** |

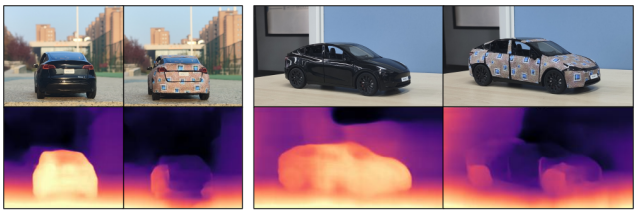


Figure 9. Physical-world evaluation using two scaled car models in different scenes.

Table 4. Ablation study for each proposed module and loss. Values are $E_{\mathrm{d}}$ of Monodepth2.

| Proposed losses | | | Proposed modules | | | |
|---|---|---|---|---|---|---|
| $L_{\mathrm{a}}$ | $L_{\mathrm{st}}$ | $L_{\mathrm{nps}}$ | None | TC | PA | Full |
| ✓ | | | 7.36 | 10.93 | 8.08 | 12.54 |
| ✓ | | ✓ | 7.24 | 10.1 | 7.62 | 11.36 |
| ✓ | ✓ | | 8.06 | 11.42 | 9.31 | **13.04** |
| ✓ | ✓ | ✓ | 7.67 | 11.06 | 8.15 | 12.75 |


Figure 10. Attack comparison with different modules.

ground and lighting conditions. We collect a total of 300 images on various environmental conditions (i.e., directions, angles, distances, and surroundings) using a Redmi K60 phone. Figure 9 shows that the depth of the normal appearance of the car can be accurately estimated by the MDE model. In contrast, the car with adversarial texture successfully deceived the MDE model, extending to regions devoid of texture coverage. On evaluation, the real-world attack results on Monodepth2 are presented in Table 3, which shows that our adversarial texture can successfully perform an adversarial attack even in the real world.

### 4.3. Ablation Study

To evaluate how each component contributes, we investigate our proposed modules and the loss function items using ablation studies with default parameters. We attack Monodepth2 and use the vehicle as the target object to report $E_{\mathrm{d}}$ and $R_{\mathrm{a}}$. The results in Table 4 verify that our proposed two transformation modules play a key role in enhancing the attack performance. When both modules are used, the attack performance is improved from 7.67m to 12.75m. Figure 10 presents the texture seeds and their corresponding attack effects under each module combination. Notably, the texture seed after applying the texture conversion reveals a substantial shift from irregular patterns to more structured and visually natural configurations.

Table 4 also illustrates the results of different combinations of loss functions. The best performance is achieved using the combination of adversarial loss and smoothness loss. However, as expected, adding the NPS loss slightly decreases the attack performance since it additionally constrains the texture to be more natural.

## 5. Conclusion

In this paper, we propose 3D Depth Fool (3D$^2$Fool) and validate its superior performance over the current state-of-the-art attacks across various scenarios, including vehicles, MDE models, weather conditions, and viewpoints. In particular, we validate the effectiveness of 3D$^2$Fool in the physical world under different backdoor and indoor backgrounds and lighting conditions by printing and pasting the 3D adversarial texture on a scaled car model.

For future work, we would further improve 3D$^2$Fool in relatively challenging settings, e.g., for certain angles where the texture coverage area is limited, and for car models with complex shapes. In addition, we would explore the transferability of 3D$^2$Fool in practical, black-box attack scenarios.

## 6. Acknowledgments

We would like to thank Zijun Chen, Ziyi Jia, Chen Ma, and the anonymous reviewers for their valuable feedback. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB3100700; the National Natural Science Foundation of China under Grant 62376210, 62161160337, 62132011, U21B2018, U20A20177, U20B2049, 62206217; the Shaanxi Province Key Industry Innovation Program under Grant 2023-ZDLGY-38 & 2021ZDLGY01-02; the China Postdoctoral Science Foundation under Grant 2022M722530 & 2023T160512; and the Fundamental Research Funds for the Central Universities under Grant xzy012022082, & xtr052023004. Chenhao Lin and Chao Shen are the corresponding authors.

## References


[1] Tesla ai day. https://youtu.be/j0z4FweCy4M?t=5295, . 1

[2] Tesla use per-pixel depth estimation with self-supervised learning. https://youtu.be/hx7BXih7zx8?t=1334, . 1

[3] Automold. https://github.com/UjjwalSaxena/Automold--Road-Augmentation-Library, 2022-12-20. 4

[4] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, (ICML)*, pages 284–293, 2018. 4

[5] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 1, 2

[6] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, Los Alamitos, CA, USA, 2017. IEEE Computer Society. 1

[7] Hemang Chawla, Arnav Varma, Elahe Arani, and Bahram Zonooz. Adversarial attacks on monocular pose estimation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12500–12505, 2022. 1, 2

[8] Zhiyuan Cheng, James Liang, Hongjun Choi, Guanhong Tao, Zhiwen Cao, Dongfang Liu, and Xiangyu Zhang. Physical attack on monocular depth estimation with optimal adversarial patches. In *Computer Vision – ECCV 2022*, pages 514–532, Cham, 2022. Springer Nature Switzerland. 1, 2, 5, 6, 7

[9] Eva Coupeté, Fabien Moutarde, and Sotiris Manitsaris. Gesture recognition using a depth camera for human robot collaboration on assembly line. *Procedia Manufacturing*, 3: 518–525, 2015. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015. 1

[10] G.N. Desouza and A.C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002. 1

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 3, 5

[12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Neural Information Processing Systems*, 2014. 1, 2

[13] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 1, 2

[14] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. 2019. 1, 2, 5

[15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. 1

[16] Amira Guesmi, Muhammad Abdullah Hanif, Ihsen Alouani, and Muhammad Shafique. Aparate: Adaptive adversarial patch for cnn-based monocular depth estimation for autonomous navigation, 2023. 1, 2, 5, 6, 7

[17] Amira Guesmi, Muhammad Abdullah Hanif, Bassem Ouni, and Muhammad Shafique. Saam: Stealthy adversarial attack on monoculor depth estimation. *ArXiv*, abs/2308.03108, 2023. 1, 2, 5, 6, 7

[18] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 5

[20] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016. 1

[21] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE T. Pattern Analysis and Machine Intelligence*, 2015. 1

[22] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[23] Chen Ma, Ningfei Wang, Qi Alfred Chen, and Chao Shen. Slowtrack: Increasing the latency of camera-based perception in autonomous driving using adversarial examples. *arXiv preprint arXiv:2312.09520*, 2023. 2

[24] Chen Ma, Ningfei Wang, Qi Alfred Chen, and Chao Shen. Wip: Towards the practicality of the adversarial attack on object tracking in autonomous driving. In *ISOC Symposium on Vehicle Security and Privacy (VehicleSec)*, 2023. 2

[25] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:6120–6127, 2019. 1

[26] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196, Los Alamitos, CA, USA, 2015. IEEE Computer Society. 4

[27] Fabio Pizzati, Pietro Cerri, and Raoul de Charette. CoMoGAN: continuous model-guided image-to-image translation. In *CVPR*, 2021. 4

[28] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5506–5514, 2016. 1

[29] Kieran Saunders, George Vogiatzis, and Luis J. Manso. Self-supervised Monocular Depth Estimation: Let's Talk About

The Weather. In *The International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 5

[30] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 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*, page 1528–1540, New York, NY, USA, 2016. Association for Computing Machinery. 4

[31] Danail Stoyanov, Marco Visentini Scarzanella, Philip Pratt, and Guang-Zhong Yang. Real-time stereo reconstruction in robotically assisted minimally invasive surgery. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, pages 275–282, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 1

[32] Naufal Suryanto, Yongsu Kim, Hyoeun Kang, Harashta Tatimma Larasati, Youngyeo Yun, Thi-Thu-Huong Le, Hunmin Yang, Se-Yoon Oh, and Howon Kim. Dta: Physical camouflage attacks using differentiable transformation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15305–15314, 2022. 1, 2, 3

[33] Naufal Suryanto, Yongsu Kim, Harashta Tatimma Larasati, Hyoeun Kang, Thi-Thu-Huong Le, Yoonyoung Hong, Hunmin Yang, Se-Yoon Oh, and Howon Kim. Active: Towards highly transferable 3d physical camouflage for universal and robust vehicle evasion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4305–4314, 2023. 1, 2, 5

[34] S. Thys, W. Ranst, and T. Goedeme. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 49–55, Los Alamitos, CA, USA, 2019. IEEE Computer Society. 1, 2

[35] Maxime Tremblay, Shirsendu S. Halder, Raoul de Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision*, 2020. 4

[36] Donghua Wang, Tingsong Jiang, Jialiang Sun, Weien Zhou, Zhiqiang Gong, Xiaoya Zhang, Wen Yao, and Xiaoqian Chen. Fca: Learning a 3d full-coverage vehicle camouflage for multi-view physical adversarial attack. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2414–2422, 2022. 1, 2

[37] J. Wang, A. Liu, Z. Yin, S. Liu, S. Tang, and X. Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8561–8570, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 1, 2

[38] Jamie Watson, Michael Firman, Gabriel J. Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *The International Conference on Computer Vision (ICCV)*, 2019. 1, 5

[39] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 5

[40] Alex Wong, Safa Cicek, and Stefano Soatto. Targeted adversarial perturbations for monocular depth prediction. In *Advances in neural information processing systems*, 2020. 1, 2

[41] Tong Wu, Xuefei Ning, Wenshuo Li, Ranran Huang, Huazhong Yang, and Yu Wang. Physical adversarial attack on vehicle detector in the carla simulator. *ArXiv*, abs/2007.16118, 2020. 1

[42] Koichiro Yamanaka, Ryutaroh Matsumoto, Keita Takahashi, and Toshiaki Fujii. Adversarial patch attacks on monocular depth estimation networks. *IEEE Access*, 8:179094–179104, 2020. 1, 2, 5, 6, 7

[43] Yulong Yang, Chenhao Lin, Qian Li, Zhengyu Zhao, Haoran Fan, Dawei Zhou, Nannan Wang, Tongliang Liu, and Chao Shen. Quantization aware attack: Enhancing transferable adversarial attacks by model quantization. *IEEE Transactions on Information Forensics and Security*, 19:3265–3278, 2024. 1

[44] Yang Zhang, Hassan Foroosh, Phiip David, and Boqing Gong. Camou: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In *International Conference on Learning Representations*, 2018. 1, 2

[45] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *CVPR*, 2020. 1

[46] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. On success and simplicity: A second look at transferable targeted attacks. In *NeurIPS*, 2021. 1