

# Transient Adversarial 3D Projection Attacks on Object Detection in Autonomous Driving

Ce Zhou<sup>1</sup>, Qiben Yan<sup>1\*</sup>, and Sijia Liu<sup>1</sup>

Michigan State University, East Lansing MI 48823, USA  
 {zhouce, qyan, liusiji5}@msu.edu

**Abstract.** Object detection is a crucial task in autonomous driving. While existing research has proposed various attacks on object detection, such as those using adversarial patches or stickers, the exploration of projection attacks on 3D surfaces remains largely unexplored. Compared to adversarial patches or stickers, which have fixed adversarial patterns, projection attacks allow for transient modifications to these patterns, enabling a more flexible attack. In this paper, we introduce an adversarial 3D projection attack specifically targeting object detection in autonomous driving scenarios. We frame the attack formulation as an optimization problem, utilizing a combination of color mapping and geometric transformation models. Our results demonstrate the effectiveness of the proposed attack in deceiving YOLOv3 and Mask R-CNN in physical settings. Evaluations conducted in an indoor environment show an attack success rate of up to 100% under low ambient light conditions, highlighting the potential damage of our attack in real-world driving scenarios.

**Keywords:** Autonomous driving · 3D projection attack · Object detection · Adversarial patch.

## 1 Introduction

Object detection is a crucial task of autonomous driving systems, playing a pivotal role in ensuring the safety of human life. Accurate and reliable object detection enables autonomous vehicles (AVs) to perceive and respond to their environment, and to recognize and track objects such as pedestrians, vehicles, and obstacles. This capability is essential for making informed decisions and taking appropriate actions to navigate through complex and dynamic traffic scenarios.

However, it is a well-recognized challenge that Deep Neural Networks (DNNs) based object detectors are vulnerable to adversarial perturbations. Previous research has designed various adversarial examples (AEs) within the digital domain to attack object detection models. To ensure physical realizability, existing studies create adversarial patches or stickers [6,23,22] as AEs that take real-world

---

\* Corresponding author.

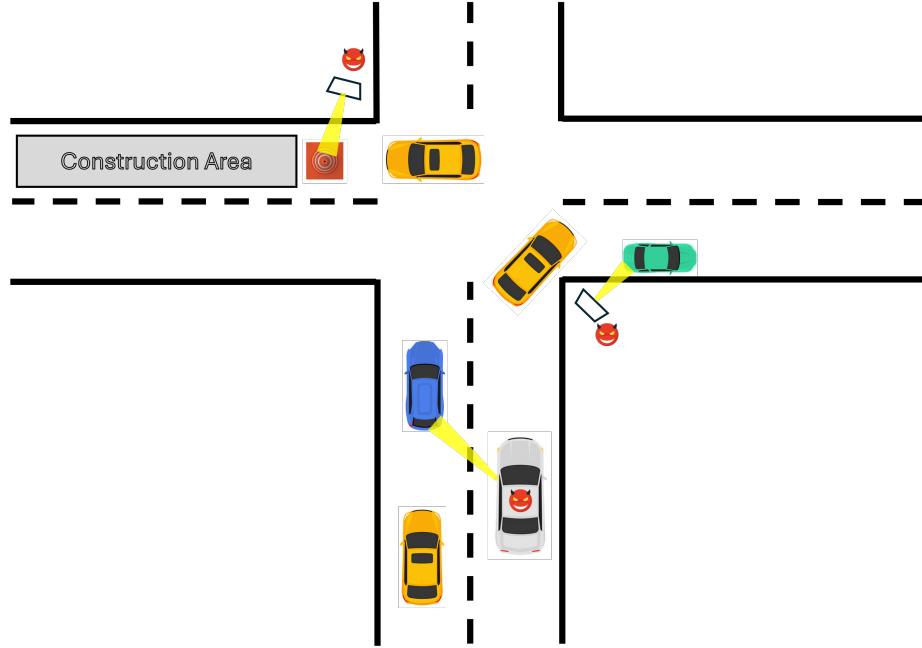


Fig. 1: The attack scenarios of 3D projection attacks: ① The blue car and the white car are driving relatively static. ② The green car parks at the roadside. ③ The traffic cone is placed at the entrance of the road construction area. The attackers position themselves either on the roadside or in another car. From these vantage points, the attacker can project an adversarial patch (projection light beams in yellow) onto the target vehicles or traffic cones, to render it undetectable by the victim AV (vehicles in yellow). This strategy could potentially lead to a collision between the victim AV and the target vehicles, or lead the victim vehicle into a dangerous road construction area.

environmental conditions into account, including diverse viewing distances or angles. However, patch-based attacks are indiscriminate toward all AVs, lacking flexibility in the physical world. This type of attack usually cannot bypass the defense mechanisms [4], as these adversarial attacks would largely perturb the model input. To overcome these limitations, the researchers proposed projection attacks to project the short-lived perturbations onto the target object to render it undetectable by the object detector [16,9]. These adversarial perturbations are crafted for use on 2D surfaces or relatively flat small 3D surfaces. However, most objects have curved or uneven large 3D surfaces, making the existing perturbations ineffective as their shapes can become distorted when projected onto a 3D surface.

In this paper, we propose an adversarial 3D projection attack against object detection in autonomous driving scenarios. The proposed attack can work on a

curved or uneven 3D surface with varying viewing angles. To the best of our knowledge, this is the first physically-realizable 3D projection attack towards AVs. We illustrate three attack scenarios in Fig. 1. The attack goal is to make the target vehicle undetectable by the victim AV, which could potentially lead to a collision between the victim AV and the target objects. In the first attack scenario, the attacker is driving the white car at the same driving speed as the blue car. The attacker projects the adversarial patch onto the surface of the blue car, causing it to disappear from the victim AV’s object detector. In the second attack scenario, the attacker projects an adversarial patch onto a green car parked on the roadside. The victim AV fails to detect the green car, potentially leading to a crash. The third attack scenario is to project an adversarial patch onto the traffic cone to render it undetected by the victim AV’s object detector. This could result in the victim AV driving into a dangerous road construction area.

Several research challenges arise in implementing the proposed attack. When projecting the image onto the object’s surface, the resulting images are influenced by various factors, including surface color, material composition, ambient light, projection distortion, viewing angle, etc. First, we need to solve the research question: “*How to map the projected image to the resulting image on the object surface to preserve the intended color schemes?*”. Second, the 3D surface is always curved or uneven, which leads to the distortion of the projected adversarial patch. Thus, the research question is: “*How to transform the projected patch onto the 3D surface to ensure its effectiveness?*”. Third, when projecting the attack pattern onto the target vehicle, the attacker and the target vehicle may both move. So, another research question is: “*How to ensure the attack’s robustness in a dynamic environment?*”.

To address these challenges, we employ a color mapping model to map the projection color onto the 3D attack surface. Additionally, we introduce a geometric transformation model that employs the Thin Plate Spline (TPS) algorithm to perform the geometric transformation from the projected patch to the 3D attack surface. To enhance attack robustness, we enhance the training dataset by applying different transformations, i.e., Expectation Over Transformation (EOT), to input images captured with various environmental factors, such as view angles, distances, etc.

We conduct experimental evaluations in an indoor environment with a model vehicle and present a video demo<sup>1</sup> to demonstrate the 3D projection attack. We list our contributions as follows:

- We are the first to propose a transient and physically-realizable projection attack by projecting an adversarial patch on a 3D object surface in autonomous driving scenarios.
- We formulate the adversarial patch generation as an optimization problem by combining a color projection model and a geometric transformation model with the target object detection network. We enhance the training data by

---

<sup>1</sup> <https://youtu.be/8RbDpAAmsjs>

considering various environmental conditions and different viewing angles of the adversarial patch, which ensures the effectiveness and robustness of the attack.

- We evaluate our attack on two object detection models using a 1/10 scale remote control (RC) car in an indoor environment. We run the experiments under varying attack distances, angles, and ambient light conditions. Evaluations show an attack success rate of up to 100% under low ambient light conditions, demonstrating the potentially damaging consequences of the proposed attack in real-world driving scenarios.

## 2 Background

We first introduce the background knowledge of projector technology and object detection. Then, we present existing literature related to physically-realizable AEs.

### 2.1 LCD Projector Technology

Liquid Crystal Display (LCD) projectors, widely employed in business presentations, classrooms, and home entertainment, utilize liquid crystal technology to project images. The fundamental concept involves splitting white light emitted by a lamp into its red, green, and blue components through dichroic mirrors. These mirrors reflect specific wavelengths while allowing others to pass through. Consequently, the white light transforms into the primary colors of red, green, and blue, forming the basis for deriving all other colors [28].

Lumens of the projector measure the brightness of a projector, indicating the total amount of light it emits. A higher lumen rating in LCD projectors enhances image visibility, especially in well-lit surroundings. Indoor projectors typically fall within the 2,000 to 3,000 lumens range for emitted light, whereas some outdoor projectors can achieve super-high lumens.

The throwing ratio of an LCD projector is defined as the distance between the projector and the screen relative to the width of the projected image. Lux is the unit for one lumen per square meter. Therefore, for a determined projecting distance, the throwing ratio influences the size and brightness of the projected image.

When we project a 2D image onto a 3D surface using a projector, the appearance of the image can be affected by the shape and texture of the surface. On smooth surfaces, the image is likely to be more uniform. However, any irregularities in the surface might cause slight distortions or uneven brightness. Textured surfaces can cause the projected image to look uneven, as the texture may interfere with the clarity of the image. If the surface is flat, the projected image will appear as it does on a flat screen, whereas, on curved 3D surfaces, the image may appear distorted, especially towards the edges. The amount of distortion depends on the degree and nature of the curvature. In a real-world attack scenario, both vehicles and road objects, such as traffic cones, have 3D

curved surfaces, making existing adversarial attacks ineffective. This paper aims to develop a new projection attack that ensures attack effectiveness on these 3D surfaces.

## 2.2 Object Detection

Object detection, a computer technology within the domain of computer vision and image processing, is centered on identifying instances of semantic objects belonging to specific classes, such as pedestrians and cars, in images and videos [27]. In this paper, we focus on Mask-RCNN [21] and Yolov3 [20], as they are representative object detectors widely used in computer vision research.

Faster-RCNN is an evolution of the initial R-CNN object detector network [8]. Utilizing a two-stage detection method, the first stage is to generate region proposals, and the second stage is to predict labels for the proposals. Additionally, Mask-RCNN [11] expands Faster-RCNN to incorporate instance segmentation, providing precise pixel-level masks alongside accurate object detection, making it ideal for applications requiring detailed object boundaries. Its flexible architecture and strong performance on benchmark datasets make it a versatile choice.

Yolo object detectors are designed to detect objects in real-time, making them highly suitable for various applications [13]. Yolov3 is a one-stage detector utilizing a single convolutional neural network (CNN), which incorporates a backbone network to compute feature maps for each square-grid cell in the input image. It employs three different grid sizes to enhance the accuracy of detecting smaller objects. Its unified architecture simplifies implementation, and its ability to perform multi-scale detection ensures robustness across various object sizes. Both Yolov3 and Mask-RCNN employ non-maximum suppression in post-processing to eliminate highly overlapped redundant boxes.

## 2.3 Physical Adversarial Examples

Athalye et al. [2] introduce the EOT framework, focusing on crafting adversarial perturbations resilient to random linear transformations. Although this approach uses a 3D printer to print out the 3D object with the AE, which can successfully deceive the image classifiers, it is not flexible in the context of dynamic driving scenarios.

Other studies [6,23,30] design robust physical perturbations, like stickers or patches, for traffic signs. These perturbations are proven resilient to changes when reproduced in the physical world, such as alterations in distance and viewing angle. Sharif et al. [22] showcase the feasibility of physical AEs for face recognition using colored eye-glass frames and enhance the perturbation realizability in the presence of input noise. Xu et al. [29,7] design a patch on a non-rigid object (i.e., T-shirt) to make the person undetected in the object detector. However, such patch-based attacks also lack the flexibility in a complex autonomous driving scenario.

Patches have drawbacks that can be mitigated by the projectors, particularly due to the short-lived and dynamic nature of projections. This allows attackers

to control and adjust the projection easily, targeting specific vehicles without leaving traces of the malicious attack. Nassi et al. [19] project 2D images directly into the real world to deceive object detectors. Man et al. [18] project optimized adversarial perturbations directly to the camera using the flare lens effect to deceive image classification. Similarly, Zhou et al. [31] exploit vulnerabilities in stereo depth estimation algorithms and use two projectors to deceive depth estimation results for AVs. In [16,9], the researchers project the perturbation on the traffic sign to make it undetectable by the object detector. In these cases, adversarial perturbations are crafted for use on 2D surfaces, despite most objects having 3D surfaces, rendering flat patches ineffective on curved 3D object surfaces.

### 3 Threat Model

Our proposed attack targets an autonomous driving scenario, in which the decision-making relies exclusively on camera sensors for the AV, such as Tesla vehicles with Full Self-Driving features and Active Safety Features [26]. The AV utilizes these cameras to identify and monitor 3D objects within the driving scene, including other moving vehicles.

**Attack Goal.** The adversaries aim to cause a traffic accident by projecting a transient adversarial patch onto a 3D object, making it undetectable by a DNN-based object detector model that processes the camera feeds within the AV. For instance, the lack of detection may cause the surrounding vehicles to collide with the victim AV without engaging the brakes.

**Attacker’s Capabilities.** Our attack is a *white-box, hiding attack*. We consider a scenario where an attacker seeks to promptly generate and project an attack patch onto an existing real-world 3D object to *hide* it from an object detection system employing a DNN model. During this timeframe, every captured frame of the scene is anticipated to yield a deceptive object detection outcome. We assume the attacker possesses comprehensive knowledge of the target DNN, including the model parameters and architecture, while remaining unaware of the technical specifications of the cameras. Additionally, the attacker is assumed to have physical proximity to the target 3D objects, such as other vehicles. This proximity enables the attacker to capture images of the target 3D objects for adversarial patch training and facilitates the projection of the attack patch into the real world. We also assume that the attacker has the capability to purchase or locate an identical object as a target for the projection attack, thereby gathering all the required data for the attack preparation (i.e., adversarial patch generation).

### 4 Attack Design

We first present an overview of the attack pipeline, and then we illustrate the design details regarding the color mapping model, geometric transformation model, adversarial patch generation, and training data enhancement.

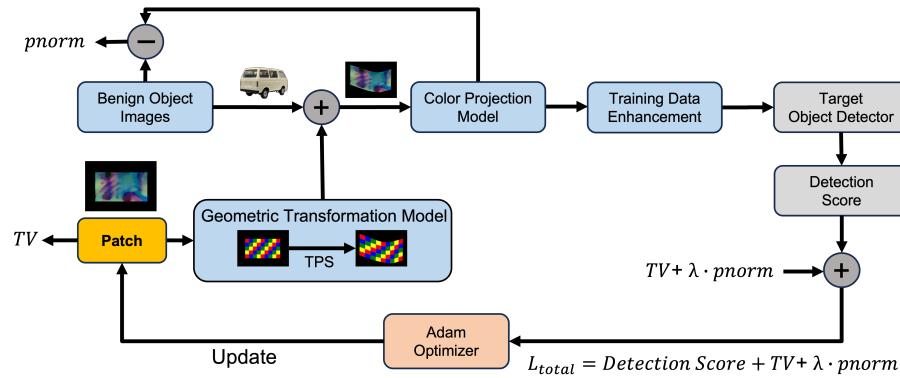


Fig. 2: Overview of the adversarial patch generation pipeline.

#### 4.1 Attack Overview

Fig. 2 shows the adversarial patch generation pipeline. The goal is to optimize the adversarial patch using a geometric transformation model and a color mapping model to minimize the target object detection score on a specific DNN-based object detector. The adversarial patch undergoes geometric transformation, and then the color mapping model is applied to simulate the patch's appearance on the target object. The geometric transformation model simulates the distortion on the 3D object surface, applying the transformation on the patch based on the 3D structure of the target object. The color mapping model simulates the projected color on the target surface. The transformed patch and the benign target object are combined through the color projection model to generate the patched final object. We then perform training data enhancement to place the patched target object in different backgrounds to enhance robustness in real-world driving scenarios. Finally, the optimization process is conducted iteratively across the enhanced training dataset to generate the adversarial patch (for projection) by minimizing the loss function.

#### 4.2 Color Mapping Model

To create physically-realizable perturbations, prior studies [22,6,23] typically leverage the non-printability score (NPS) to characterize printable colors by printers. In our attack scenarios, projecting different colors onto the target 3D object surface yields results influenced by multiple factors, such as projection strength, ambient light, projection surface, etc. Some previous investigations on the projection-based physical attacks [16,9] propose color mapping models to simulate the color overlays. The attainable range of colors is significantly reduced in comparison to the spectrum accessible to printed patches, primarily because of the color and materials of the target surface.

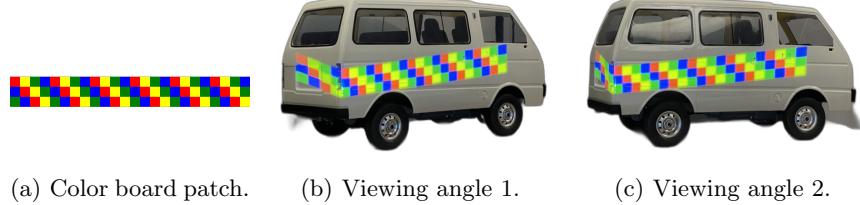


Fig. 3: Examples of two viewing angles (b)(c) of the same (a) color board patch projection on the same vehicle.

Our objective in developing the color mapping model is to establish a mapping that associates colors from the projected image onto the target 3D surface. Therefore, we formulate our color mapping model  $\mathcal{P}_c$  as follows:

$$\mathcal{P}_c(\theta_p, S, P) = O, \quad (1)$$

where  $\theta_p$  is the model parameters,  $S$  is the 3D target surface,  $P$  is the projected image, and  $O$  is the color overlays captured by the camera on the target 3D surface.

Subsequently, we collect projected data to train the color mapping model, with the goal of approximating the resulting output color for specific projected images and target 3D surfaces. The fundamental approach involves projecting various colors with RGB values ranging from 0 to 255 onto the target object surface, capturing images each time to document the outcomes. The distinct color mappings are denoted as  $(S, P_i) \Rightarrow O_i$ , where  $i$  represents each color index.

To train the color mapping model, we utilize triples  $(S, P_i, O_i)$  to fit a neural network consisting of two hidden layers with ReLU activation. Then, Eq. (1) is transformed into an optimization problem with the following loss function:

$$Loss_{\mathcal{P}_c} = \operatorname{argmin}_{\theta_p} \sum_{\forall i} \|\mathcal{P}_c(S, P_i) - O_i\|_1, \quad (2)$$

where  $\|\cdot\|_1$  denotes the L1 norm. We optimize the network parameters  $\theta_p$  using gradient descent and the Adam optimizer to derive the color mapping model.

### 4.3 Geometric Transformation Model

The EOT process [2] involves using images with various transformations, encompassing scaling, translation, rotation, brightness, additive Gaussian noise, etc. This collection aids in creating a single perturbation effective in deceiving the target neural network across diverse views. However, this approach may not adequately capture the projected image warping on a 3D object surface, particularly when viewed from different angles. For instance, in Fig. 3, the projected patch, i.e., the color board, on the 3D surface is unevenly distorted on the car's

surface. Thus, it is imperative to consider this viewing angle aspect when modeling the transformation.

Different viewing angles or relative movements between the projected 3D surface and the projector can result in varying distortions in each video frame. To address this, we incorporate TPS [3] to model the projection distortion on the 3D surface. TPS has been widely employed as a non-rigid transformation model in prior studies [12,29,7]. TPS transformation is a mathematical model used for non-linear spatial transformations, often applied in image warping and deformation. It is particularly useful in morphing one image into another while preserving local structures. TPS transformation is defined as a combination of affine, rigid, and non-rigid components. Our exploration will reveal that the non-rigid warping aspect of TPS proves to be an effective means of modeling projection distortion on 3D surfaces in learning adversarial patterns.

TPS involves learning a parametric deformation mapping that describes the displacement of each pixel from an original image  $x$  to a target image  $z$  using a set of control points with predefined positions. Suppose that the control points in the source image are  $(x_i, y_i)$  and their corresponding positions in the target image are  $(z_i, w_i)$ . The TPS transformation  $\mathcal{P}_g$  is defined as:

$$\mathcal{P}_g(x, y) = f(x, y) + \sum_{i=1}^N w_i \cdot \phi(r_i), \quad (3)$$

where  $f(x, y)$  is an affine function, and  $\phi(r_i)$  is a radial basis function defined as  $\phi(r) = r^2 \cdot \log(r)$ . The Euclidean distance  $r_i$  between the point  $(x, y)$  and the control point  $(x_i, y_i)$  is given by:

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}.$$

With the determined transformation parameters, it also allows the reverse transformation from the target image  $z$  back to the original image  $x$ . The reverse transformation is given by:

$$\mathcal{P}_g^{-1}(z) = \sum_{i=1}^N w_i \cdot \phi(r_i).$$

Please note that Eq. (3) does not explicitly define  $z_i$ . Instead, it uses the notation  $w_i \cdot \phi(r_i)$  to represent the non-rigid component of the transformation.  $z_i$  is implicitly part of the control points  $(z_i, w_i)$  in the target image but does not appear directly in the equation. In other words,  $\sum_{i=1}^N w_i \cdot \phi(r_i)$  already incorporates the influence of these control points in the transformation.

Implementing TPS to design a 3D adversarial projection patch is challenging due to the difficulty in determining the control points on both the projected and captured target surface images. To tackle this challenge, we project a checkerboard onto the target object and identify the intersection points between adjacent grid regions as the control points by manually marking them. We then scale the projected image and mark the control points using the same procedure.

#### 4.4 Adversarial Patch Generation

Our approach for generating the adversarial projection pattern involves combining the color projection model and the geometric transformation model with the target network. We employ gradient descent along both dimensions to optimize the projected image. Our objective function is defined as:

$$\underset{\delta}{\operatorname{argmin}} J(f_{\theta}(b + \mathcal{P}(x, \delta))), \quad \text{s.t. } 0 \leq \delta \leq 1, \quad (4)$$

where  $\delta$  is the projected image,  $x$  is the target object image,  $J$  is the detection loss,  $f_{\theta}$  is the target object detector, and  $b$  is the input image background.  $\mathcal{P}$  is the projection model which is defined as:

$$\mathcal{P}(x, \delta) = (1 - M) \cdot x + \mathcal{P}_c(M \cdot x, \delta_g), \quad (5)$$

where  $M$  is the mask of the patch on the target object image, and  $\delta_g$  is the patch on the target object image. Here,  $M = \mathcal{P}_g(S_{\delta})$ , and  $\delta_g = \mathcal{P}_g(\delta)$ , respectively.  $S_{\delta}$  denotes the shape of the projected image. We then rewrite Eq. (5) as:

$$\mathcal{P}(x, \delta) = (1 - \mathcal{P}_g(S_{\delta})) \cdot x + \mathcal{P}_c(\mathcal{P}_g(S_{\delta}) \cdot x, \mathcal{P}_g(S_{\delta})). \quad (6)$$

We predefine the projected patch shape. The projected patch goes through the geometric transformation model and color mapping model before it is applied to the input image. We mask the attack area on the object to ensure the perturbations only apply to the region of interest (ROI) on the 3D object surface.

To improve the practical feasibility of the projection, we impose a constant grid granularity of  $n \times n$  cells on the projection, similar to the procedure in [16]. This guarantees that each cell consists of pixels with identical colors, promoting uniform projections across various viewing distances of the target object. Additionally, we integrate the total variation in the loss function. This inclusion is intended to alleviate the effects of camera smoothing and/or blurring on the overall projection [17].

To simplify the flowing of gradients when backpropagating, we substitute  $\delta$  with a new variable  $w$  such that:

$$u = \frac{\tanh \delta}{2} + 0.5. \quad (7)$$

Since  $\delta$  is bounded in  $[0,1]$ , therefore,  $u$  is bounded in  $[-1,1]$ , which leads to faster convergence in the optimization [16].

Now, we can update our loss function Eq. (4) to constrain the amount of perturbations as:

$$\underset{\delta}{\operatorname{argmin}} J(f_{\theta}(b + \mathcal{P}(x, u))) + \lambda \|\mathcal{P}(x, u) - x\|_p + TV(u) \\ \text{s.t. } -1 \leq u \leq 1, \quad (8)$$

where  $\lambda$  is a parameter utilized to regulate the significance of the p-norm  $\|\cdot\|_p$  and  $TV$  represents the total variation. The loss  $J$  is based on bounding boxes

$b \in B$  outputted from the target object detection network. Each bounding box has a confidence score indicating the probability of containing an object of class  $j$ , denoted as  $p_j^{(b)}$ . The loss  $J$  is then defined as the sum of the confidence scores for the target object, expressed as  $\sum_{b \in B} p_j^{(b)}$ .

#### 4.5 Training Data Enhancement

Generating AEs that effectively function in autonomous driving scenarios necessitates the consideration of various environmental conditions, such as different viewing angles, distances, rotation, and brightness. To craft a robust adversarial patch, the optimization process needs to incorporate different input transformations. We adopt EOT to generate a set of training images synthetically. Our final loss is formulated as follows:

$$\begin{aligned} \operatorname{argmin}_{\delta} & \mathbb{E}_{x_q \sim X} \mathbb{E}_{b_i \sim B, m_j \sim M} J(f_\theta(b + m_j \cdot \mathcal{P}(x_q, u))) + \lambda \|\mathcal{P}(x_q, u) - x_q\|_p + TV(u), \\ & \text{s.t. } -1 \leq u \leq 1. \end{aligned} \quad (9)$$

where  $\mathcal{P}(x_q, \delta) = (1 - \mathcal{P}_g^q(S_\delta)) \cdot x_q + \mathcal{P}_c(\mathcal{P}_g^q(S_\delta) \cdot x_q, \mathcal{P}_g^q(S_\delta))$ .  $X$  represents a collection of input images with various viewing angles,  $B$  is a set of background images, and  $M$  denotes a number of linear transformations to the target object with the patch.  $\mathcal{P}_g^q$  denotes the geometric transformation model corresponding to the input object image  $x_q$ .

#### 4.6 Overall Attack Process

We summarize the detailed attack process of our attack step by step as follows:

- Step 1: The attackers determine the patch shape and obtain the parameters for the geometric transformation model. First, the attackers design a patch shape, such as a rectangle, to be used as input to the projector. They use a color board with the same shape as the patch to project it onto the surface of the target 3D object. They collect images of the object both with and without the projection. Finally, they select the control points for the TPS and derive the parameters for the geometric transformation model based on these control points.
- Step 2: The attackers collect data for the color projection model and train the model. They project different colors onto the target object and capture images of the object with these projections. This data is then fed into the color projection model for training.
- Step 3: The attackers generate the patch using the enhanced training dataset. The transformed adversarial patch and collected benign images are processed through the color mapping model and applied to the target objects. They then enhance the training dataset by incorporating EOT, placing the target objects with the adversarial patches in different backgrounds.

- Step 4: The attackers run the optimization process and update the patch. They conduct the optimization process across the enhanced training dataset using the Adam optimizer. The goal is to minimize the loss function, which includes the summation of the detection score, total variation, and p-norm of the patch. Finally, they update the patch accordingly based on the optimization results.

## 5 Evaluation

To evaluate the proposed attack, we conduct experiments on 3D projection attacks in the physical world. We investigate our attack under different environment settings to verify its feasibility and effectiveness.

### 5.1 Experimental Setup

**Attack Devices.** Fig. 4 illustrates the experimental setup. The target object is a 1/10 scale RC car [1]. For the projection, we use a PowerLite 1771W WXGA 3LCD Projector [5], an indoor projector priced around \$740, which offers a maximum brightness of 3,000 lumens and a maximum resolution of 1,280×800 pixels. The projector has a throw ratio range of 1.04–1.26. The initial experiments were conducted in a lecture room, with the projector positioned 1.5 to 2.5 meters away from the target object. Given the 1/10 scale of the RC car, this setup simulates an attack distance of 15 to 25 meters in a real-world driving scenario.

To capture images and videos of the target object, we use the default camera on an iPhone 12 Pro Max. The iPhone is mounted on a phone slider [24] to capture videos from viewing angles ranging from  $-20^\circ$  to  $20^\circ$ . These videos are recorded at various distances and angles while the projection is active, under ambient light conditions of 100 lux, 200 lux, and 500 lux.

Additionally, the data for the color mapping model are collected using the built-in webcam on an Alienware m15 R3 laptop [25]. The data for the geometric transformation model is collected by iPhone. The evaluation of the attack is conducted on the images and videos captured with the iPhone.

**Target Attack Models.** In our experiments, we evaluate our attack on YOLOv3 and Mask R-CNN object detectors. For YOLOv3, we utilize the Darknet-53 backbone [20]. For Mask R-CNN, we use ResNet-101 as a backbone along with a feature pyramid network for the region proposals [14]. Both object detectors produce a set of bounding boxes with associated confidence scores for each output class. We establish the detection threshold at 0.6, meaning that an object is detected as a “car” if the confidence score is higher than 0.6.

**Evaluation Metrics.** We feed each frame from the videos into the target attack models and count the instances where a “car” is detected at the output.

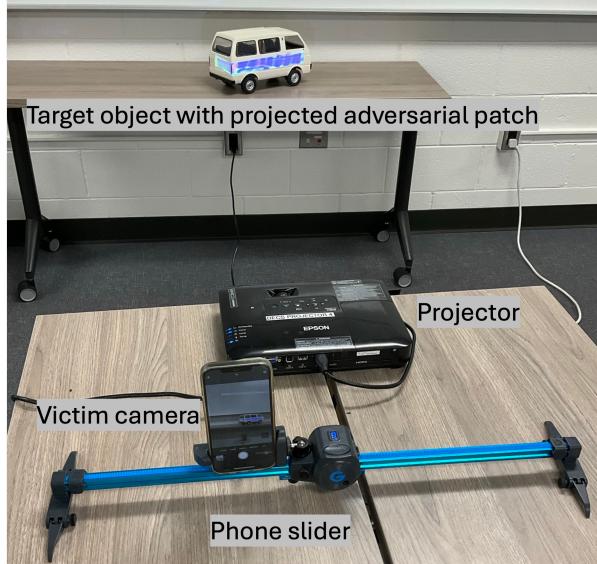


Fig. 4: Attack setup. We use the projector to project the simulated adversarial patch on the target vehicle. The phone slider is used to collect videos when the victim’s camera is moving.

We use *object misdetection rate (OMDR)* as the main evaluation metric in the experiments. It is expressed as:

$$\text{OMDR} = \frac{\text{The number of frames that do not output a class of car}}{\text{Total number of frames}}$$

**Experimental Procedure.** We follow these essential steps to conduct our experiments:

- Designing the Attack Pattern: we determine an attack pattern shape (a rectangular shape) as the input to the projector. We design a color board based on this pattern shape and project it onto the target object. Using the iPhone, we capture images of the target object both with and without the projection to obtain the benign object images and the images needed to identify the target control points, respectively. With these collected data, we run the geometric transformation model to obtain the model parameters.
- Running the Color Mapping Procedure: we run the color mapping procedure to collect different color projection images using the webcam. The collected data is then used to train and construct a color mapping model.
- Generating the Adversarial Pattern: we use the color mapping model and the geometric transformation model to generate the adversarial pattern and obtain the optimized attack pattern to input into the projector.

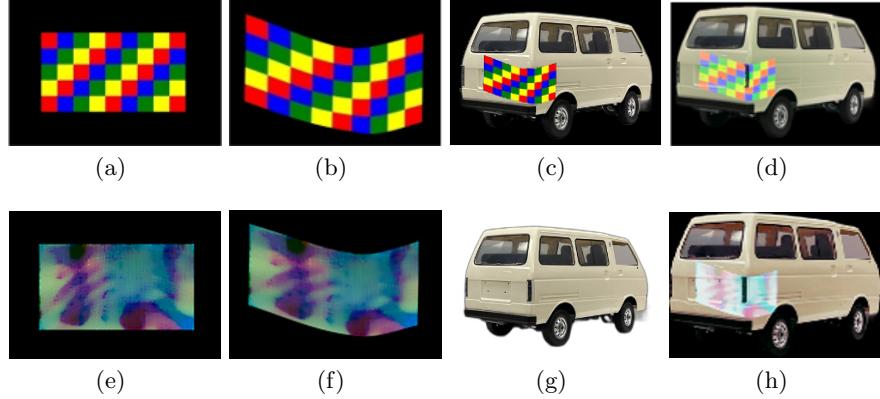


Fig. 5: A visualization example of geometric transformation using TPS. (a) The color board used to collect TPS source control points; (b) The simulated color board after TPS transformation; (c) The simulated color board projected on the vehicle; (d) The projection of the color board on the real vehicle to collect TPS target control points; (e) The trained adversarial patch; (f) The simulated adversarial patch after geometric transformation; (g) The benign vehicle image; (h) The simulated patch projected on the vehicle.

- Launching Attacks and Recording: we project the trained patch onto the target vehicle and record a set of videos at different distances and angles under various ambient light conditions.

## 5.2 Visualization of the Geometric Transformation

Fig. 5 illustrates an example of geometric transformation using TPS. Figs. 5(a)(d) are used to get source and target control points for TPS, respectively. We can then build up the geometric transformation model and obtain its corresponding parameters. Figs. 5(b)(c) are the simulated projected patch after geometric transformation, and the simulated projected patch on the vehicle. It can be seen that the simulated projected patch on the vehicle (Fig. 5(c)) and the data collected in real life (Fig. 5(d)) match very well. We utilize the parameters in the geometric transformation model and the color mapping model to convert the adversarial patch in the projector (Fig. 5(e)) to the adversarial patch on the vehicle (Fig. 5(f)). We add the transformed adversarial patch (Fig. 5(f)) onto the benign object image (Fig. 5(g)) to generate the final projected patch on the vehicle, which becomes the input to the object detector (Fig. 5(h)).

We also show visualization examples of attacks on Yolov3 and Mask R-CNN in Fig. 6. The left column on both Fig. 6(a) and Fig. 6(b) is the well-trained projected attack patch, the simulated patch after geometric transformation, and the simulated patch on the vehicle from top to bottom. The right columns are

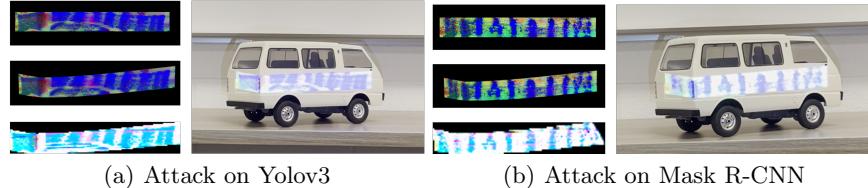


Fig. 6: Visualization examples of attacks on Yolov3 and Mask R-CNN. The left column on both (a) and (b) is the well-trained projected attack patch, the simulated patch after geometric transformation, and the simulated patch on the vehicle from top to bottom. On the right side of each figure is the captured vehicle with the projected patch.

the captured vehicle with the projected patch in the physical world. It can be seen that the simulated patch on the vehicle and the physical world match well regarding the the patch shape and color.

### 5.3 Attack Performance under Different Settings

We evaluate the 3D projection attack under three ambient light conditions with varying distances and angles for Yolov3 as shown in Fig. 7. The first row shows the baseline results of Yolov3. When there is no attack present, OMDR is usually very close to 0. Only when the ambient light is 500 lux and the viewing angle is between  $-20^\circ$  to  $7.5^\circ$  at the attack distance of 2 meters, OMDR is relatively higher. The results might be attributed to the fact that the back view of the vehicle becomes harder to detect in Yolov3 under strong ambient light.

For the scenarios with the attacks (Figs. 7(d)(e)(f)), it shows that OMDR significantly increases. With lower ambient light, the attack is usually more successful. This is because, as ambient light increases, the range of achievable colors diminishes due to the reduced impact of the projector-emitted light on the vehicle's appearance. Besides, when the attack distance is between 1.5 to 2 meters and the attack angle is between  $-7.5^\circ$  to  $7.5^\circ$ , the average OMDR is around 96%. Notably, when the ambient light is 100lux, we can achieve 100% OMDR in this attack setting. A video demo of this attack can be viewed through the link: <https://youtu.be/8RbDpAAmsjs>.

Moreover, with the attack angle between  $-20^\circ$  to  $7.5^\circ$ , it is usually easy to make the vehicle disappear in the detector. It is because that when the attack angle is between  $7.5^\circ$  to  $20^\circ$ , part of the attack patch on the backside of the vehicle has less portion of the view as shown in Fig. 3(c). On the other hand, the attack patch has a better view in the rest of the viewing angles in our attack setting (Fig. 3(b)).

We also examine the performance of our attack on different object detectors under varying ambient light conditions at an attack distance of 1.5 meters. The average OMDR results for YOLOv3 and Mask R-CNN are summarized in

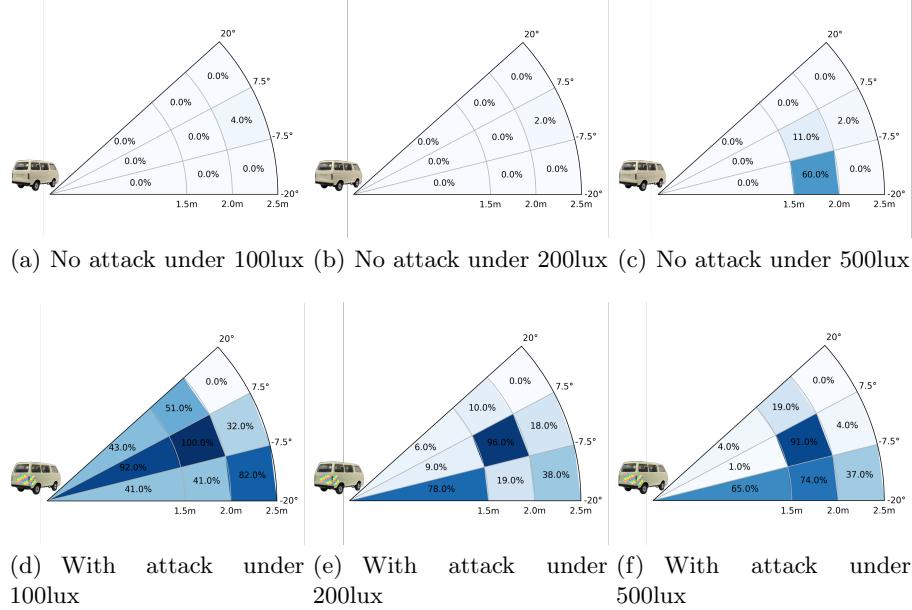


Fig. 7: OMDR for Yolov3 with varying attack distance and attack angles under ambient light 100lux, 200lux and 500lux respectively. (a)(b)(c) show the OMDR without attack. (d)(e)(f) show the OMDR of 3D projection attack.

Table 1. Our findings indicate that as ambient light increases, the success rate of the attack generally decreases for both models. Furthermore, Mask R-CNN consistently demonstrates greater resilience compared to YOLOv3, likely due to its ability to learn more robust features and its utilization of a region proposal network for detection [11]. However, this extra resilience comes at the expense of slower execution. For example, Mask R-CNN can take up to 14 times longer to perform than YOLOv3 [16].

## 6 Discussion

We mainly discuss the feasibility and practicality of the 3D projection attack.

**Attack Feasibility and Practicality.** Unlike patch attacks, which involve placing physical stickers or patches on objects, projection attacks can be temporary. The projections can be turned on and off quickly, making them harder to detect and track over time. Once the projection is turned off, there are no physical traces left behind, unlike patches or stickers that can be discovered upon inspection.

Projection attacks can change patterns dynamically, adapting to different conditions and object surfaces. This flexibility makes it more challenging for

Ambient Light (lux)	Yolov3		Mask R-CNN	
	With Attack W/O Attack			
100	59%	0%	46%	26%
200	31%	0%	36%	30%
500	23%	0%	18%	25%

Table 1: Average misdetection rate under different ambient light conditions

detection systems to recognize and filter out adversarial patterns. Projected adversarial patches can also be modified in real time based on the feedback from the detection system, which allows for continuous optimization of the attack and maintains its effectiveness.

**Ambient Light.** Our experiments show that the increasing ambient light could rapidly degrade the effectiveness of the attack in bright conditions. In a real-world physical attack scenario, the attacker could launch the attack during the daytime on overcast days or near sunset or sunrise when the ambient light level is below 400 lux.

**Moving Target Object.** Our attack is effective only on static objects or those moving at low speeds relative to the attacking device. Otherwise, targeting the vehicle becomes problematic. Therefore, our attack is more suitable for vehicles parked on the roadside or traffic cones on the road. Though the attack may only last for several seconds due to movement issues, it is a split-second attack. This means that even if it functions for only a few seconds, it can potentially lead to severe consequences.

**Data Collection.** The geometric transformation model and the color mapping model need to be well-trained in advance. Consequently, essential data collection is required. The attacker may purchase or locate an identical object to the target and gather all necessary data to develop these models beforehand to launch a more accurate attack.

**Potential Defense.** One potential defense method is adversarial training [15,10], which incorporates AEs, including those generated by projection attacks, into the training process. This helps the model learn to recognize and ignore projected adversarial patterns. Previous research [15] proposed a method to locate and segment the adversarial patch, incorporate adversarial training, and then completely remove the patch. However, in the case of our projection attack, if the projected patch size on the target object is enlarged or if the patch fully covers the object, removing the patch entirely might still result in the benign object being mis-detected due to the large area of the object being obscured.

Another possible defense is to verify the consistency of detections over time. Since our projection attack is short-lived, it may cause sudden changes in detected objects over time. Checking for temporal consistency can help identify and discard adversarial detections and alert the human driver.

## 7 Conclusion

We propose a transient adversarial 3D projection attack targeting object detection in autonomous driving scenarios, which projects a well-crafted adversarial patch onto a 3D surface. The 3D projection attack is formulated as an optimization problem, combining a color mapping model and a geometric transformation model. We enhance the robustness of our attack by considering various environmental factors. We conduct experiments to evaluate the proposed attack against YOLOv3 and Mask R-CNN object detectors in physical attack scenarios. Our evaluation results show an attack success rate of up to 100% under low ambient light conditions. This research underscores the need for defense strategies to mitigate transient projection attacks on AI-driven autonomous vehicles. Future work should focus on creating adaptive and resilient countermeasures that can detect and neutralize such attacks in real time, thereby safeguarding both passengers and pedestrians in diverse driving conditions.

## Acknowledgements

We would like to extend our appreciation to the anonymous reviewers for their invaluable input on our study. This work was supported in part by the U.S. National Science Foundation grant CNS-2235231.

## References

1. Amazon: Mostop Remote Control Car D42. [https://www.amazon.com/dp/B0BRRJ1GV7?psc=1&ref=ppx\\_yo2ov\\_dt\\_b\\_product\\_details](https://www.amazon.com/dp/B0BRRJ1GV7?psc=1&ref=ppx_yo2ov_dt_b_product_details) (2020), accessed: 2024-07-20
2. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. In: International conference on machine learning. pp. 284–293. PMLR (2018)
3. Bookstein, F.L.: Principal warps: Thin-plate splines and the decomposition of deformations. IEEE Transactions on pattern analysis and machine intelligence **11**(6), 567–585 (1989)
4. Chou, E., Tramer, F., Pellegrino, G.: Sentinel: Detecting localized universal attacks against deep learning systems. In: 2020 IEEE Security and Privacy Workshops (SPW). pp. 48–54. IEEE (2020)
5. Epson: PowerLite 1771W WXGA 3LCD Projector. <https://epson.com/For-Work/Projectors/Portable/PowerLite-1771W-WXGA-3LCD-Projector/p/V11H477020> (2020), accessed: 2024-07-20

6. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust physical-world attacks on deep learning visual classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1625–1634 (2018)
7. Fan, Q., Xu, K., Chen, C.F., Liu, S., Zhang, G., Cox, D.D., Lin, X.: Generating realistic physical adversarial examples by patch transformer network (2021)
8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
9. Gnanasambandam, A., Sherman, A.M., Chan, S.H.: Optical adversarial attack. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 92–101 (2021)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
12. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28** (2015)
13. Kanjee, R.: Is YOLOv6 v3.0 better than YOLOv8? <https://augmentedstartups.medium.com/is-yolov6-v3-0-better-than-yolov8-4bb2a9a18805> (2023), accessed: 2024-07-20
14. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
15. Liu, J., Levine, A., Lau, C.P., Chellappa, R., Feizi, S.: Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14973–14982 (2022)
16. Lovisotto, G., Turner, H., Sluganovic, I., Strohmeier, M., Martinovic, I.: SLAP: Improving physical adversarial examples with short-lived adversarial perturbations. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 1865–1882 (2021)
17. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5188–5196 (2015)
18. Man, Y., Li, M., Gerdes, R.: GhostImage: Remote perception attacks against camera-based image classification systems. In: 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020). pp. 317–332 (2020)
19. Nassi, B., Mirsky, Y., Nassi, D., Ben-Netanel, R., Drokin, O., Elovici, Y.: Phantom of the adas: Securing advanced driver-assistance systems from split-second phantom attacks. In: Proceedings of the 2020 ACM SIGSAC conference on computer and communications security. pp. 293–308 (2020)
20. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
21. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
22. 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)

23. 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)
24. Store, N.: Neewer 39.4"/100cm Motorized Camera Slider, 2.4G Wireless Control Carbon Fiber Dolly Rail Slider, Support Video Mode, Time-Lapse Photography, Horizontal, Tracking and 120° Panoramic Shooting (VS-100WC). [shorturl.at/cfjNO](http://shorturl.at/cfjNO) (2022), accessed: 2024-07-20
25. Technologies, D.: ALIENWARE m15 GAMING LAPTOP. <https://www.dell.com/en-us/shop/dell-laptops/alienware-m15-r3-gaming-laptop/spd/alienware-m15-r3-laptop> (2020), accessed: 2024-07-20
26. Tesla: Autopilot and full self-driving capability. <https://www.tesla.com/support/autopilot> (2022), accessed: 2024-07-20
27. wikipedia: Object detection. [https://en.wikipedia.org/wiki/Object\\_detection#:~:text=Object%20detection%20is%20a%20computer,in%20digital%20images%20and%20videos.](https://en.wikipedia.org/wiki/Object_detection#:~:text=Object%20detection%20is%20a%20computer,in%20digital%20images%20and%20videos.) (2023), accessed: 2024-07-20
28. Wilkinson, S.: LCD, LCoS, or DLP: Choosing a Projector Imaging Technology. <https://www.projectorcentral.com/Digital-Projector-Imaging-Technologies-Explained.htm> (2023), accessed: 2024-07-20
29. 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: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. pp. 665–681. Springer (2020)
30. Zhao, Y., Zhu, H., Liang, R., Shen, Q., Zhang, S., Chen, K.: Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. pp. 1989–2004 (2019)
31. Zhou, C., Yan, Q., Shi, Y., Sun, L.: DoubleStar: Long-range attack towards depth estimation based obstacle avoidance in autonomous systems. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 1885–1902 (2022)