TPatch: A Triggered Physical Adversarial Patch

Wenjun Zhu and Xiaoyu Ji, *USSLAB, Zhejiang University;* Yushi Cheng, *BNRist, Tsinghua University;* Shibo Zhang and Wenyuan Xu, *USSLAB, Zhejiang University*

https://www.usenix.org/conference/usenixsecurity23/presentation/zhu

This paper is included in the Proceedings of the
32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

TPatch: A Triggered Physical Adversarial Patch

Wenjun Zhu
USSLAB, Zhejiang University
zwj_@zju.edu.cn

Xiaoyu Ji
USSLAB, Zhejiang University
xji@zju.edu.cn

Yushi Cheng*
BNRist, Tsinghua University
yushithu@mail.tsinghua.edu.cn

Shibo Zhang
USSLAB, Zhejiang University
zhsb@zju.edu.cn

Wenyuan Xu
USSLAB, Zhejiang University
xuwenyuan@gmail.com

Abstract

Autonomous vehicles increasingly utilize the vision-based perception module to acquire information about driving environments and detect obstacles. Correct detection and classification are important to ensure safe driving decisions. Existing works have demonstrated the feasibility of fooling the perception models such as object detectors and image classifiers with printed adversarial patches. However, most of them are indiscriminately offensive to every passing autonomous vehicle. In this paper, we propose TPatch, a physical adversarial patch triggered by acoustic signals. Unlike other adversarial patches, TPatch remains benign under normal circumstances but can be triggered to launch a hiding, creating or altering attack by a designed distortion introduced by signal injection attacks towards cameras. To avoid the suspicion of human drivers and make the attack practical and robust in the real world, we propose a content-based camouflage method and an attack robustness enhancement method to strengthen it. Evaluations with three object detectors, YOLO V3/V5 and Faster R-CNN, and eight image classifiers demonstrate the effectiveness of TPatch in both the simulation and the real world. We also discuss possible defenses at the sensor, algorithm, and system levels.

1 Introduction

The growth of autonomous vehicles (AVs) spawns the wide deployment of automotive cameras. According to the forecast [26], the automotive camera market is predicted to grow at a CAGR of 12.4% and reach a value of USD 12.2 billion by 2025. By providing precise images of surrounding environments, the automotive cameras and the subsequent perception models such as object detectors and image classifiers constitute the vision-based perception modules, which detect and classify obstacles such as cars and stop signs on the roads to help AVs make safety-critical driving decisions. As a result, correct detection and classification in the midst of a dedicated adversary are important to ensure safe driving.

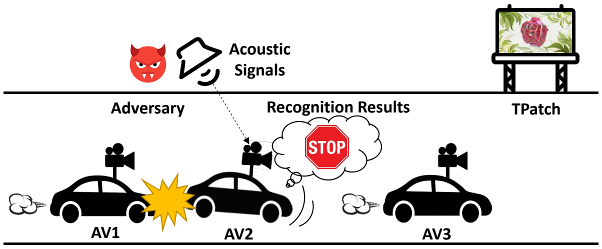


Figure 1: TPatch attacks. The patch is preset at the roadside, which is benign to most passing vehicles such as AV1 and AV3, but can render the targeted vehicle under acoustic injection attacks (AV2) recognize a non-existing stop sign, leading to tragic results.

However, prior work [3, 5, 8, 17, 38, 47] has demonstrated the vulnerabilities of the vision-based perception module to carefully-crafted adversarial examples. Among the existing ones, the adversarial patch that appears in the form of localized perturbations draws much attention in the physical attacks towards AVs since it is physically realizable and robust to slight distortions. Much work [3, 5, 8, 47] has demonstrated the effectiveness of physically-printed adversarial patches against classifiers or detectors. However, existing adversarial patches, either in the digital or physical worlds, are indiscriminately offensive to every passing AV and thus stand a chance of being detected by pilot cars equipped with recent countermeasures [9, 41].

In this paper, we ask *"Can we design a physical adversarial patch that is only activated when triggered but remains benign under normal circumstances?"* Inspired by prior work [20, 31] that demonstrates physical signals in addition to the natural lights may also cause distortions to images, we propose TPatch, a physical adversarial patch triggered by acoustic signals. Unlike common adversarial patches, TPatch remains benign under normal circumstances but can be triggered to launch a hiding, creating or altering attack toward a targeted AV by a designed distortion introduced by signal

[*] Yushi Cheng is the corresponding author.

[†] Source code & demo: https://github.com/USSLab/TPatch

injection attacks towards cameras. Specifically, we consider the following attack scenario: *The adversary puts a TPatch at the roadside, which is benign to most passing vehicles but can render the targeted vehicle under acoustic injection attacks to recognize a non-existing stop sign, resulting in an emergency brake and thus a car collision, as shown in Fig. 1.*

TPatch is promising yet challenging. Several research challenges remain unsolved. First, how to ensure that TPatch is sensitive to the image distortion caused by the selected acoustic signal while dull to other natural or unintentional distortions? Second, how to enhance the visual stealthiness of TPatch and reduce the suspicion of human drivers or passengers when placed on the real road? Third, how to improve the robustness of TPatch to make it more practical in the real world?

To address the aforementioned challenges, we first design a pair of triggers, where each stands for a kind of distortion, consisting of a *positive trigger* that activates TPatch and a *negative trigger* that suppresses TPatch. To find the feasible positive triggers, we estimate the image distortion caused by easy-to-implement physical signals and regularize the estimated distortion with a proposed theoretical model. Based on the selected positive and negative triggers, we design trigger-oriented loss functions for object detectors and image classifiers receptively to generate optimized TPatch. Moreover, to reduce the human suspicion of the generated TPatch, we propose a content-based camouflage to make the patch similar to an image or object. Finally, to improve the robustness of TPatch in real-world attacks, we employ expectation over transformation and triggerable space enlargement to further enhance it.

To validate our attacks, we conduct both stimulation and real-world evaluations with three object detectors YOLO V3/V5 and Faster R-CNN, and eight image classifiers. In summary, our contributions include the points below:

- To the best of our knowledge, this is the first work on the physical adversarial patch that can be triggered by sensor attacks.
- We propose a trigger-oriented optimization method to generate TPatch, a content-based camouflage method to reduce the suspicion of human drivers, and an attack robustness enhancement method to further make the patch practical and robust in the real world.
- We evaluate the effectiveness of TPatch with three object detectors, YOLO V3/V5 and Faster R-CNN, and eight image classifiers in both white-box and black-box settings.
- We realize three kinds of realistic attack implementation in the outdoor real-vehicle driving-based experiments and achieve the max success rate of 100% and 86.4%, w.r.t. hiding attack and creating attack, in 150 consecutive frames.

2 Background

In this section, we first introduce the vision-based perception module used in autonomous driving, then present adversarial patches that can fool machine learning algorithms, and finally elaborate on physical signals that may cause distortions to images.

2.1 Vision-based Perception Module

In autonomous driving, a vision-based perception module is critical for decision-making. The camera first converts the lights reflected from physical objects to electrical signals, processed and digitized to create digital images. Then, the perception models utilize machine learning algorithms to detect or classify objects in the images, where the perception results will be used for decision-making.

In recent years, convolutional neural networks (CNNs) based perception models have achieved high accuracy in many perception tasks. Object detectors and image classifiers are two representative types commonly used in autonomous driving. Object detectors are usually utilized to coordinate the location of critical objects, e.g., pedestrians, vehicles, etc., and classify them in a given group of classes. State-of-the-art object detectors include (1) one-stage ones represented by the YOLO series [28, 40] and (2) two-stage ones represented by Faster R-CNN [29]. By contrast, image classifiers are used for more fine-grained and specialized classification tasks [2], e.g., recognizing the states of the traffic light (red, yellow, or green) or the meanings of the traffic sign (stop, speed-limit, school, etc.) [19]. Representative image classifiers include models of various network architectures such as VGG [35], ResNet [18] and Inception [37]. Both types of models provide important perception results for higher-level functions such as tracking, planning, and decision-making. In this paper, we study the vulnerabilities of both object detectors and image classifiers and their potential threats to autonomous driving.

2.2 Adversarial Patch

Adversarial patch [5] is one type of adversarial example that appears in the form of localized perturbations. It can be a sticker attached to an existing object or a stand-alone image such as a poster. Compared to various pixel-wise perturbations directly adding to images in the digital world [17, 38], the adversarial patch is more likely to be practical in the real world since (1) it is physically realizable, (2) it can be placed anywhere within the field of view to cause the perception model to output a targeted class, and (3) it is image-independent and robust for rotation and scaling. Due to its practicality and robustness, it has drawn much attention recently, and a few prior works [3, 5, 8, 47] have demonstrated the feasibility of physical adversarial patch attacks on either classifiers or detectors. However, existing adversarial patches are indiscriminately offensive to every passing vehicle and thus risk being detected by pilot cars. In this paper, we investigate the possibility of targeted adversarial attacks toward a specific victim vehicle by using a triggered patch.

2.3 Physical Signals Affecting Imaging

In general, the natural lights reflected from physical objects and received by the camera's image sensors are the main physical signals that affect the output images. However, other physical signals may cause distortions to the images during the imaging process, leading to misclassification of the subsequent perception models. For instance, prior work [20] has demonstrated that acoustic signals can cause controllable blur patterns on the output images by affecting the image stabilization system of the camera. Another work [31] has shown that laser diodes can inject colorful strips into the output images by exploiting the rolling shutter of the camera. These works inspire us to explore the feasibility of an adversarial patch that needs a specific image distortion caused by physical signals as a trigger to be effective. Since those physical signals, e.g., acoustic signals, lasers, etc., are commonly observed and not difficult to generate in the real world, they may be candidates for the trigger signals. In this paper, we study the blur effects caused by acoustic signals as the trigger to serve as the first example of `TPatch`.

3 Threat Model

3.1 Attack Goal

In this paper, we consider the following attack goals:

- **Hiding Attacks (HA)** against object detectors that hide the detection of an existing object.
- **Creating Attacks (CA)** against object detectors that induce the detection of a non-existing object.
- **Altering Attacks (AA)** against image classifiers that render an object misclassified to another one.

3.2 Adversary Capability

To achieve the aforementioned attack goals, we assume the adversary has the following capabilities:

Prior Knowledge of Perception Algorithms. We assume the adversary has prior knowledge of the object detection and image classification algorithms similar to those used in the victim AV, including their architecture, parameters, etc. Besides, the adversary can utilize the transferability of adversarial examples to achieve black-box attacks.

Camera and Sensor Awareness. The adversary can acquire a camera of the same model as the one used in the victim AV, from which she can learn the information of the camera and sensors, e.g., the camera exposure time, the physical locations and parameters of the inertial sensors, etc.

Signal Injection Capability. The adversary can launch acoustic injection attacks toward the inertial sensors in the victim AV by the ultrasonic transducers (1) attached to the surface of the victim AV, (2) placed at a following vehicle, or (3) placed at the roadside.

3.3 Design Requirement

To make the attacks practical in the real world, `TPatch` shall achieve the following design requirements:

Conditionally Triggered. `TPatch` is adversarial if and only if it is triggered by a specific physical signal. In other words, `TPatch` shall be benign to object detectors or image classifiers when not triggered, and shall not be triggered by signals other than the designed one.

Scene Independent. In the scenario of open roads, traffic participants are dynamic and hard to predict. As a result, `TPatch` shall be adaptive to visual changes in the targeted scene, i.e., the patch shall be somewhat universal.

Easy Implementation. The physical trigger signal shall be easy to generate and transmit. Thus, the patch optimization shall be trigger-oriented, i.e., we shall design an easy-to-implement signal and consider it during the patch optimization instead of finding an optimal trigger difficult to realize. In addition, `TPatch` shall be robust to slight distortions of the trigger signal.

4 Design

4.1 Overview

In this paper, we propose `TPatch`, a physical adversarial patch triggered by specific physical signals. Different from other adversarial patches, `TPatch` remains benign under normal circumstances but can be triggered to launch a universal attack by a designed distortion introduced by signal injection attacks towards cameras. To craft `TPatch`, it is important to tackle the following challenges:

Challenge 1: How to ensure that `TPatch` is sensitive to the image distortion caused by the selected trigger signal while dull to other natural or unintentional distortions?

Challenge 2: How to enhance the visual stealthiness of `TPatch` while keeping it functionally stealthy?

Challenge 3: How to improve the robustness of `TPatch` to make it more practical in the real world?

To address these challenges, we design `TPatch` attacks with four modules, as shown in Fig. 2. The **Trigger Design** module first defines the positive trigger that activates `TPatch` and the negative trigger that suppresses `TPatch`. Then, to find the feasible positive triggers, the module estimates the image distortion caused by physical signals and regularizes the

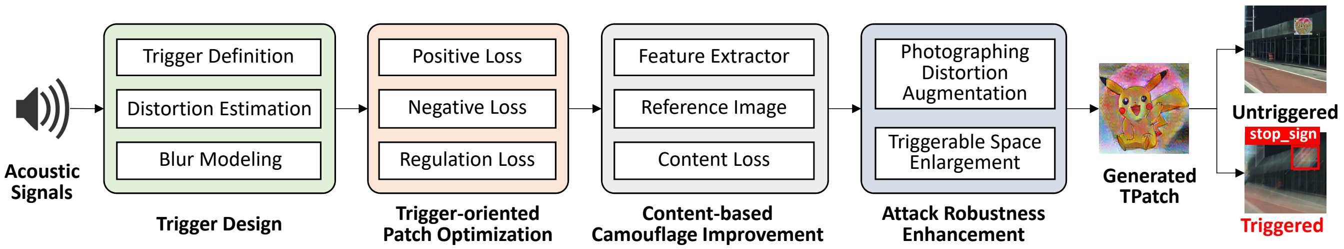


Figure 2: Overview of TPatch generation. Based on the selected physical trigger signal, the adversary first estimates the image distortion caused by it and designs the positive and negative triggers. Then, she trains a TPatch in accord with the designed triggers, and finally improves the visual camouflage and the robustness of the patch to make it more practical in the real world. The generated TPatch can then be attached to any objects to launch hiding, creating or altering attacks.

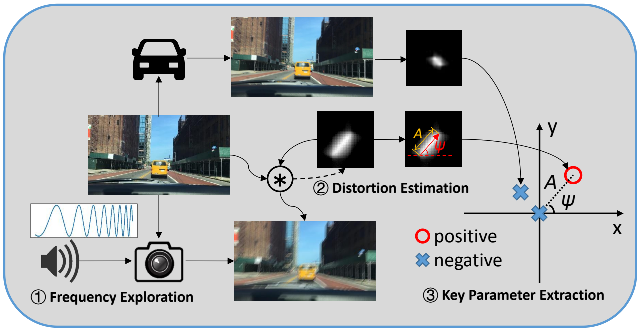


Figure 3: Diagram of the trigger design. ① Explore the resonant frequency by injecting frequency-modulated acoustic signals. ② Estimate the PSF kernel with the clear image and the corresponding blurred image. ③ Extract the strength and orientation of blur.

distortion with a proposed theoretical model. The **Trigger-oriented Patch Optimization** module then designs loss functions based on the selected positive and negative triggers to generate TPatch capable of achieving the corresponding attack goals via gradient optimization. The **Content-based Camouflage** module further improves the visual stealthiness of TPatch by applying meaningful content extracted from a reference image using a pre-trained feature extractor. The **Real-world Robustness Improvement** module improves the robustness of TPatch in the real world by addressing both the deformation of the patch and the errors of the trigger signal. In the following subsections, we present the details of each module.

4.2 Trigger Design

TPatch is designed to launch adversarial attacks when triggered by a specific physical signal while remaining benign when not triggered. To generate TPatch, we first design its trigger.

Trigger Definition. In this paper, we design the trigger of TPatch to be a carefully crafted image blur. Specifically, we consider two opposite types of triggers: (1) positive trigger means that TPatch is activated by acoustic signals and leads to adversarial detection consequences, and (2) negative trigger means that TPatch is not triggered yet other factors cause undesired detection results, e.g., strong vibration caused during normal driving.

Signal Injection. We investigate which physical signal can create the expected image blur. Cameras rely on the outputs of motion sensors for image stabilization, which however can be manipulated by resonant acoustic signals via aliasing effects. As shown in Eq. 1, the aliasing frequency f_a depends on the injected signal frequency f as follows:

$$f_a = |f - nf_s| \leq 0.5 f_s \tag{1}$$

When the injected signal frequency f is an integer multiple n of the sampling frequency f_s, the output of motion sensors, i.e., the aliased signal, is a constant bias. Thus, it can be reshaped arbitrarily through an amplification modulation [39].

However, this method faces two major challenges in practice: (1) The phase or frequency shifting caused by the lack of synchronization may damage the attack capability in consecutive frames, as discussed in Appendix A.1. (2) The designed image blur may be difficult to implement in the real world due to the limited acoustic signal injection capability. For example, the outputs of the x/y-axis gyroscope are coupling because of their identical resonant frequencies.

To solve the first challenge, we propose a simple yet effective signal injection approach. As the resonant frequencies are a range rather than a point [39], we shift the frequency of the injected signal from the integer multiple of the sampling frequency nf_s to the frequency $nf_s \pm f_e$, where f_e represents the exposure frequency. In this way, the generated image blur can be resistant to phase shifting and frequency shifting, as discussed in detail in Appendix A.2.

Blur Modeling. Then, we model the blurred image under signal injection attacks. A blurred image can be viewed as the superposition of a series of translated clean images. To simulate the blur effects, we formulate the relationship between the blurred image and the translation of image sensor, as shown in Eq. 2:

$$
\begin{aligned}
\hat{u}^{\gamma}_{i,j} &= \frac{1}{N}\sum_{t=1}^{N} Sample(i+\Delta i(t), j+\Delta j(t); v^{\gamma}) \\
\Delta i(t) &= A\cos\psi\sin(2\pi\frac{t}{N}+\phi_i) \\
\Delta j(t) &= A\sin\psi\sin(2\pi\frac{t}{N}+\phi_j)
\end{aligned} \tag{2}
$$

where $\hat{u}$ and v denote the blurred and clean image, respectively. i and j represent the geometric position of a pixel. $\Delta i(t)$ and $\Delta j(t)$ represent the translation of the camera, which is in a form of sine wave with the similar phases ϕ_i, ϕ_j but different amplitudes $A\cos\psi, A\sin\psi$. $Sample(\cdot;\cdot)$ denotes the sampling function by interpolation, e.g., bilinear. N represents the fineness of discretion. $\gamma = 2.2$ is the parameter of gamma correction [1], which converts the RGB values into the luminosity that is linear with exposure duration. The strength A and the orientation ψ are the key parameters of blur modeling. The slight phase difference $\Delta\phi = |\phi_i - \phi_j|$ has limited effects on TPatch as shown in Fig. 19 in Appendix F.

Parameter Estimation. In the prior work, the attacker first optimizes the desired blur by simulation and then implements it in the real world. However, not all the optimized blur can be realized due to the coupling of the x/y-axis gyroscope. To ease the burden of the physical attack and make it more practical in the real world, we propose an estimation-based method to obtain the possible image blur set by injecting acoustic signals easy to implement. In general, the image blur caused by motions (introduced by acoustic signals) can be described with a convolutional kernel named Point Spread Function (PSF) [14], as shown in Eq. 3:

$$
\hat{u}^{\gamma}_{i,j} = \sum_{a=-k}^{k}\sum_{b=-k}^{k} v^{\gamma}_{i-a,j-b} h_{a,b} \tag{3}
$$

where h denotes the PSF, whose kernel size is $(2k+1)\times(2k+1)$ and k is an integer. $\hat{u}$ is the convolutional result of the PSF and v is the clean image. To estimate the kernel, we formulate it as an optimization problem, as shown in Eq. 4:

$$
\min_h \|u-\hat{u}\|_2 = \min_h \|u-(v^{\gamma} * h)^{1/\gamma}\|_2 \tag{4}
$$

where u denotes the blurred image captured from a physical signal injection attack (the ground truth), and symbol $*$ denotes the convolutional operator. We use the Mean-Square-Error (MSE) between the convolutional result $\hat{u}$ and the real-captured image u as the loss function and solve the optimization problem via gradient descent. In this way, we obtain the estimated PSF h and calculate its strength A and orientation ψ as the parameters of a possible image blur.

4.3 Trigger-oriented Patch Optimization

With the designed positive and negative triggers, we then optimize TPatch based on them. Note that in contrast to common adversarial patches that have a straightforward goal to fool image classifiers, object detectors, or other victim systems, TPatch is benign until it is triggered. As a result, we consider several losses when design the loss functions: (1) a positive loss L_{pos} that makes the patch achieve the adversarial effects with a positive trigger, (2) a negative loss L_{neg} that makes the patch dull to any negative triggers, (3) a regulation loss L_{TV} that reduces the risk of over-fitting and enhances the capability of migrating the attack from the digital world to the physical world, and (4) a content loss $L_{content}$ that realizes a meaningful camouflage on the generated patch (details can be found in Sec. 4.4). The integral loss function is shown in Eq. 5:

$$
\arg\min_p \mathbb{E}_{x\sim X, l\sim L, t\sim T}\left[L_{pos}+\lambda L_{neg}\right]+\alpha L_{TV}+\beta L_{content} \tag{5}
$$

where p is the TPatch. x, l, t represent the input image, the location where the patch is applied, and the random transformation respectively. Then, X, L, T denote the corresponding distribution. $\mathbb{E}$ denotes the expectation of the loss. Hyper-parameters, e.g., λ, α, β, are used to balance different loss components.

The positive and negative losses, i.e., L_{pos} and L_{neg}, are different across object detectors and image classifiers. The classifier outputs a single vector that represents the probability of each class while the detector predicts the objectiveness scores, the classification scores, and the object's location. For classifiers, the aggregation of features enables the patch to manipulate all the outputs. However, for detectors that have multiple predictive cells to recognize multiple objects, the patch's location mainly determines the affecting area due to the locality of the patch. As a result, we shall align the target object's location with the adversarial patch to achieve an effective attack. What's more, the losses vary as the attack goal changes. Therefore, we discuss the detailed design of the loss function concerning each attack type.

For HA, the adversarial case is to render the target object with the TPatch undetected, while the benign one is to keep the target object correctly recognized. The loss functions of HA are formulated as Eq. 6.

$$
\begin{aligned}
L_{pos} &= -\log(1-\max(p_{obj}\cdot p_t)) \\
L_{neg} &= -\log\left(p_{obj}\cdot p_t\right)+\zeta L_{reg}
\end{aligned} \tag{6}
$$

where p_t and p_{obj} denote the classification scores of target class t and the objectiveness scores respectively. L_{reg} is the regression loss to guide the detected box. ζ is the hyperparameter to balance the recognition loss and regression loss.

For CA, the adversarial case is to render the TPatch recognized as the target object, while the benign one is to keep the TPatch undetected. The loss functions of CA are formulated as Eq. 7.

$$
\begin{aligned}
L_{pos} &= -\log\left(p_{obj}\cdot p_t\right)+\zeta L_{reg} \\
L_{neg} &= -\log(1-\max(p_{obj}\cdot p_t))
\end{aligned} \tag{7}
$$

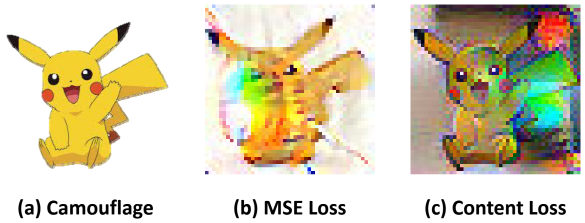


Figure 4: Comparison between MSE loss and content loss. (a) shows the target camouflage of the patch. (b) and (c) show the generated patch by MSE loss and content loss respectively.

For AA, the adversarial case is to render the images with the TPatch recognized as the target class, while the benign one is to keep the original recognition results. The loss functions of AA are formulated as Eq. 8.

$$\begin{aligned} L_{pos} &= -\log(p_t) \\ L_{neg} &= -\log(1-p_t) \end{aligned} \tag{8}$$

where p_t is the probability of target class t predicted by the classifier.

The total variation loss is used for regularizing the TPatch. As shown in Eq. 9, the TV loss aims to reduce the color changes between the adjacent pixels, which eliminates the overfitting of TPatch in the digital simulation as well as improves the image quality.

$$L_{TV} = \sum_{i,j} \sqrt{\left|x_{i+1,j} - x_{i,j}\right|^2 + \left|x_{i,j+1} - x_{i,j}\right|^2} \tag{9}$$

4.4 Content-based Camouflage Improvement

Common adversarial patches are usually generated without constraints on the L_2/L_∞ for the sake of universality, and thus are inevitably perceptible to human eyes. Although prior work has demonstrated the feasibility of changing the shape of the patch arbitrarily, it remains abnormal since the pattern of the patch is meaningless. To address it, we propose the content-based camouflage improvement to make the patch more meaningful, e.g., to be like a cartoon character or an art painting.

A vanilla approach is to add a loss function that constrains the patch with a euclidean distance between the target image and the patch. However, this vanilla approach leads to a tricky problem regarding the choice of the target image, i.e., the punishment of the loss varies dramatically with different target images. Besides, the vanilla loss function is color-based, which means the outline of the camouflage is prone to diminish, rendering the camouflage ambiguous and over-smooth.

To improve the camouflage, we utilize a content loss proposed by style transfer works [15, 21] to regularize the patch based on high-level features extracted by a pre-trained CNN. The content loss can be formulated as Eq. 10. We use the j-th layer ϕ_j of the CNN pre-trained on ImageNet to respectively extract the high-level feature map of target image $\hat{x}$ and patch x with the same shape $C_j \times H_j \times W_j$. The loss is calculated with the euclidean distance in the feature space rather than the pixel space. It encourages the patch to learn the content and spatial structure of the target image rather than the details, e.g., color, texture, and exact shape. A comparison of TPatch with the proposed content loss and the conventional MSE loss is shown in Fig. 4, from which we can see that the content loss helps TPatch to be more natural as a cartoon character.

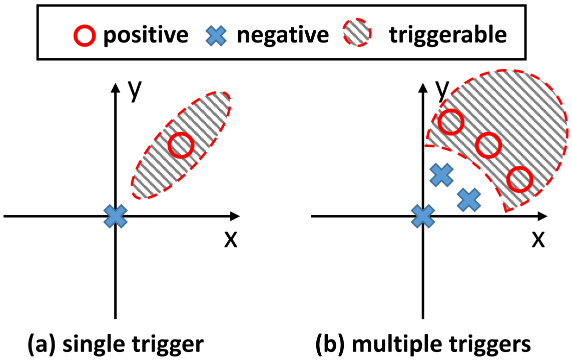
$$L_{content} = \frac{1}{C_j H_j W_j} \left\|\phi_j(\hat{x}) - \phi_j(x)\right\|_2^2 \tag{10}$$

4.5 Attack Robustness Enhancement

The challenges of robust TPatch attacks mainly lie in two aspects: (1) the captured patch can be different from the digital one due to the additional process of printing and filming, and (2) the injected trigger can be coarse and noisy due to the limited physical attack capability and the estimation errors. The former challenge is shared with other adversarial patches. Factors commonly considered affecting the filming include lighting conditions, shooting distances and angles. The latter is unique for TPatch since it requires a certain trigger to be adversarial. Inevitably, trigger errors can be introduced in the pipeline of physical signal injection, trigger estimation, and trigger formulation. The difference between the estimated trigger and the ground truth probably causes the failure of activating the TPatch. To tackle these challenges, we utilize the expectation over transformation for the former and the triggerable region enlargement for the latter.

Expectation over Transformation. To address the patch distortion caused by photographing, we use the Expectation over Transformation (EoT) method [3], which augments the training of TPatch with random transformation to overcome the various situations in the real world. We augment the patch with three dimensions, i.e., resize, rotation, and color shift (including brightness, contrast, saturation, and hue). We perform three different transformations simultaneously with a uniform distribution to randomize the degree of each transformation.

Triggerable Region Enlargement. Since an adversarial subspace usually includes more than one point, triggers close to the designed positive/negative trigger can be positive/negative as well, according to our observation. Inspired by this phenomenon, we propose to enlarge the triggerable region by employing more positive and negative triggers, as illustrated in Fig. 5. Such an operation helps avoid the mild blur existing in normal driving from activating TPatch and improves the robustness of TPatch with more possible directions and strengths of the blur.



(a) single trigger

(b) multiple triggers

Figure 5: Illustration of the relationship between the triggerable region and the designed triggers.

5 Evaluation

In this section, we evaluate `TPatch` against both object detectors and image classifiers. We consider two sets of evaluations in this paper: (1) simulation evaluation, where `TPatch` is attached to digital images directly, and (2) real-world evaluation, where `TPatch` is physically printed to conduct attacks, and the experiments are conducted with the outdoor real-vehicle driving-based setup. We use the attack success rate (ASR) as the metric to evaluate the simulation experiments and use the highest attack success rate in n consecutive frames $f_{succ}^{max(n)}$ [34, 47] to evaluate the real-world experiments.

In summary, we highlight the key results of our attacks as follows:

- In the simulation evaluation, `TPatch` can achieve overall ASRs of 88.4% and 91.9% for HA and CA against three object detectors YOLO V3/V5 and Faster R-CNN, and achieve an overall ASR of 85.7% for AA against eight image classifiers.
- In the real-world evaluation, `TPatch` can achieve overall $f_{succ}^{max(150)}$ of 100.0% and 86.4% for HA and CA against three object detectors YOLO V3/V5 and Faster R-CNN.
- We conduct attacks in three attack scenarios, i.e., in-car, car-to-car, and roadside, with an autonomous driving vehicle on closed roads. The farthest feasible attack range reaches about 6 meters.
- We extensively investigate the transferability and effectiveness of `TPatch` under various driving scenarios in the real world.

5.1 Simulation Evaluation

In the simulation evaluation, we investigate three types of attacks, i.e., hiding attack (HA), creating attack (CA), and altering attack (AA), in the public datasets extensively. We first evaluate multiple classes of interest for each attack under a consistent setting. Then we investigate the impact of critical factors such as patch sizes and trigger parameters on the effectiveness of `TPatch`. Finally, we study the transferability of `TPatch` with two black-box attacks, i.e., single model attack and ensemble model attack.

5.1.1 Experimental Setup

Object Detectors. We evaluate `TPatch` using three popular object detectors, including both one-stage ones YOLO V3/V5 and a two-stage one Faster R-CNN. The backbones of the pre-trained models Faster R-CNN, YOLO V3, and YOLO V5 are ResNet-50, Darknet-53, and CSPDarknet, respectively. All the three models are trained on the MS Common Objects in Context (COCO) dataset [10] for detection.

Image Classifiers. We evaluate `TPatch` on eight widely-used CNNs, i.e., VGG-13/16/19, ResNet-50/101/152, Inception v3, and MobileNet v2, which cover models of different depths and architectures. All of them are trained on the large vision database ImageNet [11] for classification.

Datasets. For object detectors, we utilize two popular autonomous driving datasets KITTI [16] and BDD100K [45] that consist of images captured from real driving scenarios for evaluation. For the image classifier, we utilize the ImageNet validation set for evaluation. We use 2,000 and 10,000 images that are unseen in the training of the `TPatch` for testing detectors and classifiers, respectively.

Classes of Interest. We select 8 classes for object detectors and 20 classes for image classifiers as the classes of interest. The classification or detection results of these classes are security-related in the scenario of autonomous driving.

Metrics. We use the attack success rate (ASR) as the metric in the simulation evaluation, which is the ratio of the number of successful attacks against an object detector or image classifier over the total number of conducted attacks. We define a `TPatch` attack to be successful when it achieves the targeted adversarial effect under triggering signals while it does not modify the detection results without triggering. With this definition, the metric can be formulated as Eq. 11:

$$ASR = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}_{F(x,tr_p)=y_a \& F(x,tr_n)=y_b}(x) \qquad (11)$$

where $\mathbb{I}(x)$ is the indicator function. F means the recognition function of the target model. x means the image embedded with `TPatch`. tr_p and tr_n are the positive trigger and negative trigger respectively. y_a and y_b denote the adversarial and benign recognition results respectively.

More concretely, the predicted class is always the one that has the highest confidence, and the prediction of `TPatch` is decided by the Intersection-over-Union (IoU) between any bounding box with the bounding box of `TPatch`, where we use the threshold 0.5 to generate a binary result. In accord with common sense, we refer to the object attached with the

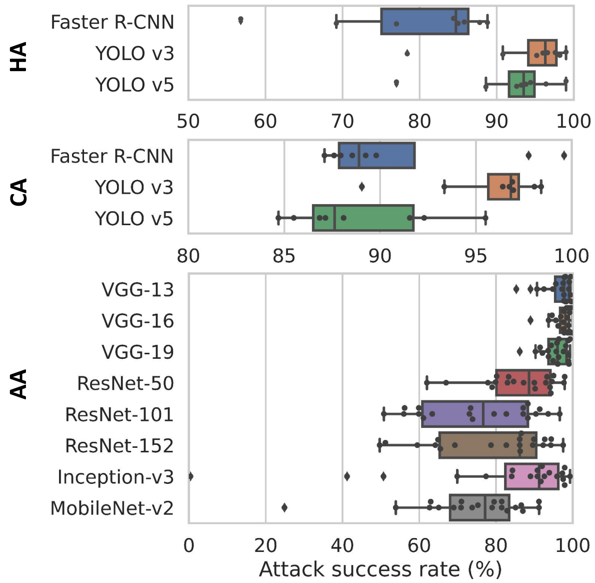


Figure 6: Overall effects of TPatch with regard to HA (top), CA (middle) and AA (bottom). The box plots show the min/max and quarterlies, and the dots represent different classes of interest.

TPatch as the targeted box rather than the TPatch itself when evaluating HA.

5.1.2 Overall Performance

In this subsection, we evaluate the effectiveness of TPatch attacks on different detectors and classifiers, respectively. During the evaluation, we used a pair of positive and negative triggers. The strength of the positive and negative triggers are 8 pixels and 0 pixels, respectively, and the orientation of the positive trigger is 45°. The patch size is set to 48×48 pixels in the overall evaluation.

For every single test of CA and AA, we randomly place the TPatch on a single image and then use a pair of the designed positive and negative triggers to generate two images that stand for the adversarial and benign cases for recognition. According to the metric formulated in 11, the overall success rate is calculated. Unlike the other attacks, the TPatch of HA is attached to the target object rather than standalone. According to its annotations, the target objects are cropped from the MS COCO dataset. The appropriate size bounds filter them to ensure the occupied proportion, round 20% of the object's area, of the patch.

The results of overall performance evaluation are shown in Fig. 6. For HA, the overall success rates are 79.4%, 94.0%, and 91.9%, w.r.t. Faster R-CNN, YOLO V3 and YOLO V5. For CA, the overall success rates are 91.0%, 95.7%, and 89.0%, w.r.t. Faster R-CNN, YOLO V3 and YOLO V5. For AA, the VGG series are the easiest to attack (over 95%), followed by ResNet and Inception (over 75%), and the worst is MobileNet (73.8%). We find that the results have notable differences across classes, shared with different classifiers and detectors. For example, in the experiments of Inception v3, the most vulnerable class is *traffic light*, which exceeds the 99% success rate. In contrast, the least vulnerable class, *trolleybus*, only achieves a 0.04% success rate, which means it hardly has adversarial effects. In addition, it is observed that the worst classes are shared with different models, e.g., *trolleybus*, *pickup*, etc. The same phenomenon is observed in the HA and CA experiments on detectors. For instance, *car* is the easiest to achieve CA, and *person* is the most difficult to achieve HA in all the detectors.

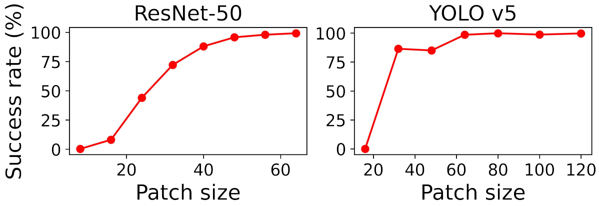


Figure 7: Attack success rates under different patch sizes. The shape of patch is square, and the patch size is denoted by the side length of the square.

5.1.3 Impact of Patch Details

In this subsection, we investigate several possible factors that may influence the adversarial effects of TPatch, including the size of the patch and the strength and orientation of the trigger.

For simplicity, we use several representative models for patch detail analysis. As for the image classifier, we choose the ResNet-50, whose overall performance lies in the medium position among the eight models, and the *traffic light* as the attack target of AA. As for the object detector, we choose the YOLO V5, which is one of the latest models, and the *stop sign* as the attack target of CA, which keeps consistency with the real-world evaluation.

Impact of Patch Size. The patch size represents the capability of an attacker to manipulate the input image. In the autonomous driving scenario, the small patch size indicates the far distance between the camera and the patch, and the large patch size represents that the camera is near the patch. As shown in Fig. 7, too tiny patches hardly achieve the success of the attack. The success rate curve is relatively smooth in the classifier and achieves the success rate of nearly 100% when the patch size is 64×64 pixels, while two leaps of success rate in the detector can be observed at 32×32 pixels and 64×64 pixels. The reason is related to the three scales in YOLO V5. As the patch size arises, it is more likely to be wrongly predicted by the detector.

Impact of Triggers. To investigate whether different image blurs would influence the adversarial effect of TPatch, which is crucial for our trigger-oriented optimization, we set different positive triggers whose strength and orientation are

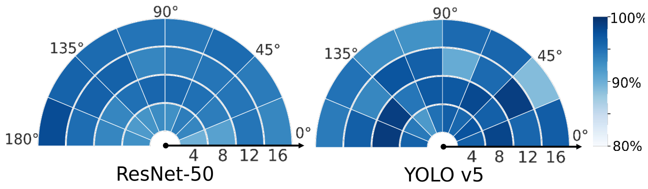


Figure 8: Attack success rates with different triggers. The strength and orientation of blur trigger is illustrated by polar coordinates.

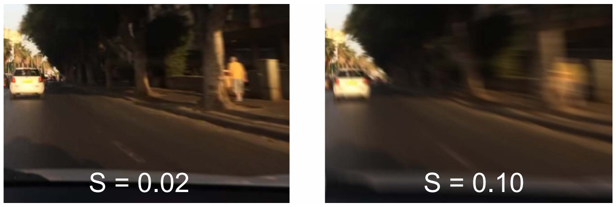


Figure 9: Images with different degrees of radial blur. The degrees of radial blur equal 0.02 (left) and 0.1 (right).

diverse. Due to the symmetrical effects of linear blur, we design the range of orientation from 0 to 180 degrees, and we set the range of strength from 4 to 16 pixels. Meanwhile, we keep other parameters as default in the overall experiment. As shown in Fig. 8, the TPatch achieves more than 88% success rate in every blur pattern with different orientations and strengths. Thus, it is demonstrated that the effectiveness of TPatch does not rely on the specific blur.

5.1.4 Stealthiness Study

The captured images are not always clear because of the car movements and vibrations. To investigate the functional stealthiness of TPatch, we simulate the possible blur effects via two methods, i.e., (1) radial blur for car movements and (2) other triggers for car vibrations.

Car Movements. We use radial blur to simulate the blur effects caused by car movements. To measure the boundary of the radial blur, we drive at a speed of 40-60 km/h in the low light scenario meanwhile sampling real-world blur effects. We compare the captured images with the simulated images to estimate the degree of radial blur $S \approx 0.02$. Then we test the TPatch for CA attacks on 500 images with different degrees of radial blur. The experiment results show that the mis-triggering rate is 0% when $S = 0.02$ and achieves 5.2% when $S = 0.1$ (corresponding to a car speed around 200-300 km/h), as illustrated in Fig. 9. Thus, the radial image blur caused by a moving car at a reasonable speed may not trigger the TPatch.

Car Vibrations. Since the car vibrations are various in the real world, we simulate the effects by investigating the ASRs with triggers of various strengths A and orientations ψ. Tab. 1 outlines the ASRs of TPatch under different triggers. The shadowed cells represent the designed positive triggers, while the rest are negative triggers. We find the following observations. In most cases, TPatch cannot be mis-triggered, and the exceptions are a few cases that have similar strengths A and orientations ψ as the positive ones. Nevertheless, the average mis-triggering rate is only 1%.

Table 1: Attack success rates under different triggers.

ψ A	15°	45°	75°	105°	135°	165°
4.5	0%	0%	0%	0%	0%	0%
9	0%	5%	10%	0%	0%	0%
18	54%	97%	87%	0%	0%	0%

5.1.5 Transferability Study

When the adversary has very limited prior knowledge of the DNN models used in commercial autonomous cars, the gradient-based optimization approach can not be directly applied to those black-box models. However, the attacker is still possible evade the target model via the transferability of adversarial examples among similar DNN models. To evaluate the transferability of TPatch, we conduct a single model attack on detectors and an ensemble model attack on classifiers. During the evaluation, the design details of TPatch, e.g., target class, patch size, trigger setting, etc., is all fixed except for the recognition with different models.

Tab. 2 list the results of transfer attacks on different detectors. Due to the similar model architecture, the success rates of transfer attacks between YOLO V3 and YOLO V5 are relatively high, 86% (average), than the one between the one-stage detectors (YOLO series) and the two-stage detectors (Faster R-CNN), 49% (average), in all the HA and CA experiments. As we use the same metric as formulated in Eq. 11, the suppression effect from negative triggers is confirmed in the black-box models in addition to the adversarial effect from positive triggers.

We investigate the ensemble model attack on different classifiers. Given that the attacker has a series of white-box models with the same architecture while attacking a black-box model, we build two ensemble models, i.e., VGG-ens (VGG13+VGG16+VGG19) and Res-ens (Res50+Res101+Res152), as the white-box model to optimize the TPatch to launch transfer attacks on the other five models. Tab. 3 list the detailed success rates of ensemble transfer attacks. The VGG-ens and Res-ens attack achieve a 38% and 46% average success rate on five black-box models. Since the number of classes in ImageNet, 1000, is much more than the number of classes in MS COCO, 80, the overlap of decision space between two different models is smaller theoretically; thus, the targeted transfer attack would be more challenging.

Table 2: Transferability across different detectors.

Attack Type	Hiding Attack			Creating Attack		
Detector	Faster R-CNN	YOLO V3	YOLO V5	Faster R-CNN	YOLO V3	YOLO V5
Faster R-CNN	85.8%	43.7%	42.8%	97.9%	31.2%	46.3%
YOLO V3	63.8%	98.9%	86.2%	34.9%	99.1%	87.1%
YOLO V5	70.1%	78.6%	93.8%	61.5%	93.8%	99.8%

Table 3: Transferability across different classifiers.

Classifier	VGG13	VGG16	VGG19	Res50	Res101	Res152	Incv3	Mobv2
VGG-ens	94.0%	96.0%	96.2%	42.7%	32.0%	49.4%	34.3%	32.8%
Res-ens	26.1%	51.8%	51.9%	91.8%	90.9%	91.4%	52.4%	45.6%

5.2 Real-world Evaluation

In the real-world evaluation, we focus on these two kinds of attacks, i.e., HA and CA against object detectors under both white-box and black-box settings. We implement attacks with three scenarios, i.e., in-car, car-to-car, and roadside. Furthermore, we extensively investigate the influence of various factors on the attack performance, including attack distances, victim cameras, vehicle speeds, road conditions, and lighting conditions.

5.2.1 Experimental Setup

Real-world Setup. We utilize an advanced Apollo Kit autonomous vehicle as the victim vehicle, a platform to experiment with future self-driving techniques. Its sizes are 2.77 m (length) * 1.01 m (width) * 1.67 m (height). We mount a smartphone, e.g., Samsung S20 or iPhone 7, on the front of the vehicle to mimic its computer vision system. The acoustic signals are generated with an arbitrary waveform generator, an audio amplifier and ultrasonic devices, as shown in Fig. 10. The detailed setup varies with different attack scenarios, i.e., in-car, car-to-car, and roadside scenarios. The required attack power and the scale of ultrasonic devices increase as the attack range rises. We use the same waveform generator and audio amplifier to drive the ultrasonic transducers for convenience. Yet the realistic setup can be on a much smaller scale, especially for the in-car scenario. For HA, the TPatch is printed on a sticker attached to a normal stop sign board. For CA, the TPatch is printed on a board, with a realistic size of $60cm \times 60cm$. All the driving experiments are done on the closed roads in our institute.

Methodology and Metric. In the real-world evaluation, we take the in-car scenario as the default scenario, and the attack performance in different scenarios is investigated standalone in Sec. 5.2.3. The victim vehicle is driving towards the TPatch at a constant speed meanwhile keeping the TPatch within the view of camera. The vehicle starts at a distance of round 15 m to the TPatch and stops at a distance of 1-2 m to the TPatch. The smartphone records the videos of driving experiments, which is recognized by the victim detector on the server. To have a better understanding of the effectiveness of the TPatch, we split the positive trigger and negative trigger evaluation and adopt a new metric, i.e., the best attack success rate in n consecutive frames $f_{succ}^{max(n)}$ [34, 47], to evaluate the recognition of videos. The metric is formulated as Eq. 12:

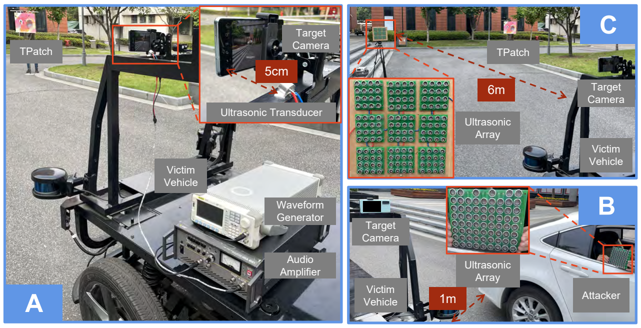


Figure 10: Experimental setups in three attack scenarios. (A) shows the in-car scenario where a single ultrasonic transducer is attached near the camera to inject acoustic signals. (B) shows the car-to-car scenario where the attacker launches an acoustic injection from the following car. (C) shows the roadside scenario where a stationary ultrasonic array is utilized to conduct roadside attacks.

$$f_{succ}^{max(n)} = \max_{j} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{F\left(x_{i+j}\right)=y_a}\left(x_{i+j}\right) \qquad (12)$$

where x_i denotes the i_{th} frame of the captured video x. To quantify our evaluation, we choose three frame lengths n 50, 100, and 150, i.e., $f_{succ}^{max(50)}$, $f_{succ}^{max(100)}$, and $f_{succ}^{max(150)}$.

5.2.2 Overall Performance

To evaluate the overall performance, we craft 6 TPatches (3 for HA and 3 for CA) against three detectors, i.e., Faster R-CNN, YOLO V3, and YOLO V5. The victim vehicle is configured to a slow speed, 5 km/h, and the average duration of captured video clips is around 10 s, and then the number of frames is about 300 at an fps of 30. Since every TPatch is trained against a specific detector by gradient-based optimization, we investigate its transferability via recognizing the same video clips with the other black-box detectors.

Table 4: Overall performance of hiding attacks and creating attacks in real-world setups.

Attack Type		Hiding Attack			Creating Attack		
Source	Target	$f_{succ}^{max(50)}$	$f_{succ}^{max(100)}$	$f_{succ}^{max(150)}$	$f_{succ}^{max(50)}$	$f_{succ}^{max(100)}$	$f_{succ}^{max(150)}$
Faster R-CNN	Faster R-CNN	100.0%	100.0%	100.0%	100.0%	90.0%	64.0%
Faster R-CNN	YOLO V3	100.0%	100.0%	98.0%	96.0%	65.0%	46.7%
Faster R-CNN	YOLO V5	100.0%	98.0%	96.7%	30.0%	27.0%	20.0%
YOLO V3	Faster R-CNN	100.0%	100.0%	100.0%	100.0%	95.0%	92.7%
YOLO V3	YOLO V3	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
YOLO V3	YOLO V5	100.0%	98.0%	96.0%	100.0%	79.0%	77.3%
YOLO V5	Faster R-CNN	100.0%	100.0%	100.0%	96.0%	82.0%	71.3%
YOLO V5	YOLO V3	100.0%	100.0%	99.3%	90.0%	83.0%	77.3%
YOLO V5	YOLO V5	100.0%	100.0%	100.0%	100.0%	98.0%	95.3%

Tab. 4 demonstrate the overall performance of HA and CA. It is indeed possible to achieve transfer attacks in the real world. For CA, the average $f_{succ}^{max(50)}$, $f_{succ}^{max(100)}$, and $f_{succ}^{max(150)}$, are 90.2%, 79.9%, and 71.6%, respectively, where the white-box ones are 100%, 96.0%, and 86.4% while the transferred ones are 85.3%, 71.8%, and 64.2%. For HA, the average of $f_{succ}^{max(150)}$ is 100% (white-box) and 98.3% (transferred), which has a better performance than CA. The possible reason is that the HA `TPatch` attached to the normal stop sign destroys the robust feature of the stop sign, which is agnostic to the recognition algorithm, which contributes to the blurring attacks. For comparison, we test the pure blur attack case whereby the stop sign has no patch and the camera is under similar levels of acoustic signal injection and found that the pure blur attack is difficult to hide the stop sign, where $f_{succ}^{max(150)} = 3.3\%$.

5.2.3 Attack Performance in Various Scenarios

The implementation of acoustic signal injection can vary for different attack scenarios, i.e., in-car, car-to-car, and roadside scenarios. We summarize the setups of different attack scenarios in Tab. 5.

- For an in-car scenario, an attacker may place the attack device near the target, whereby the attack device can be one ultrasonic transducer and may be left there during car maintenance. The attack device can be in a small size to be stealthy.
- For a car-to-car scenario, an attacker may drive a car to follow the victim's vehicle and emit acoustic signals with a transducer array. The attack distance is about 1 m.
- For a roadside scenario, an attacker may set up an ultrasonic array on the roadside, emitting acoustic signals to a driving-by victim vehicle. The farthest attack distance can be up to 6 m.

Note that in the roadside attack scenario, the attacker has to handle the Doppler effects since there may exist a relative speed between the ultrasonic devices and the victim's vehicle. Details of discussion on the Doppler effects are referred to Appendix B.

Table 5: Setups of three attack scenarios.

Scenario	Non-intrusive	Distance	Device Scale	Power
in-car	✗	5 cm	$1.6cm \times 1.6cm$	0.08W
car-to-car	✓	1 m	$15cm \times 15cm$	7W
roadside	✓	6 m	$30cm \times 30cm$	40W

Table 6: Effectiveness in three attack scenarios.

Attack Type	Scenario	$f_{succ}^{max(50)}$	$f_{succ}^{max(100)}$	$f_{succ}^{max(150)}$
HA	in-car	100.0%	100.0%	100.0%
HA	car-to-car	100.0%	99.0%	96.0%
HA	roadside	100.0%	99.0%	93.3%
CA	in-car	100.0%	99.0%	96.7%
CA	car-to-car	96.0%	91.0%	86.7%
CA	roadside	98.0%	97.0%	91.3%

To investigate the effectiveness of `TPatch` among various attack scenarios, we conduct experiments for each attack scenario and summarize the results in Tab. 6. For the in-car scenario, the $f_{succ}^{max(150)}$ is 100% for HA and 96.7% for CA, respectively. The attacks of an in-car scenario achieved the best performance because the transducer is close to the camera and remains relatively stationary to the camera, which results in imposing stable acoustic signals on the camera. Nevertheless, even with the car-to-car or roadside scenarios whereby the attack transducers are placed external to the victim vehicle, we can still achieve HA with a $f_{succ}^{max(150)}$ of over 93.3%, and achieve CA with a $f_{succ}^{max(150)}$ of over 86.7%.

To explore the upper bound of the attack distance, we conduct experiments with various attack distances. We utilize the roadside attack setup with a $30cm * 30cm$ ultrasonic transducer array at an apparent power level of 40W. Tab. 7 lists the results of $f_{succ}^{max(50)}$ at different attack distances from 2 m to 7 m. We find that we can achieve a $f_{succ}^{max(50)}$ of 100% for HA with an attack distance up to 6 m, and a $f_{succ}^{max(50)}$ of 98% for CA with an attack distance up to 5 m. The attack performance begins to decrease when the distance exceeds 5 m, which is because sounds attenuate with distances. Based on the inverse square law, the energy of sound received at 7 m is almost half of that at 5 m. The attenuated acoustic signals are more likely to be affected by environmental factors, leading to a degradation of the attack performance in consecutive frames.

Table 7: Impact of attack distances.

Distance	2 m	3 m	4 m	5 m	6 m	7 m
HA	100%	100%	100%	100%	100%	70%
CA	100%	100%	98%	98%	74%	48%

Table 8: Effectiveness with different cameras.

Attack Type	Camera	Freq.	$f_{succ}^{max(50)}$	$f_{succ}^{max(100)}$	$f_{succ}^{max(150)}$
HA	Samsung S20	20 kHz	100.0%	100.0%	100.0%
HA	iPhone7	27 kHz	100.0%	100.0%	100.0%
CA	Samsung S20	20 kHz	100.0%	98.0%	92.7%
CA	iPhone7	27 kHz	98.0%	93.0%	94.0%

To investigate whether the attacks can affect various cameras, we attack another camera equipped with image stabilization techniques, iPhone7, in addition to Samsung S20. Since iPhone7 mounts a different motion sensor from the one on Samsung S20, their resonant frequencies are different, i.e., 20 kHz for Samsung S20 and 27 kHz for iPhone7. But we can cause a similar image blur on different cameras via acoustic signals with different frequencies. As shown in Tab. 8, we evaluate both HA and CA on iPhone7 and find that our attacks can work with iPhone7 as well with a $f_{succ}^{max(150)}$ of 100% for HA and a $f_{succ}^{max(150)}$ of 94.0% for CA.

5.2.4 Impact of Environmental Factors

In this subsection, we investigate how those common environmental factors influence the attack performance of the TPatch. We mainly take three aspects of environmental factors into consideration, i.e., car movements, car vibrations, and lighting conditions. We conduct two series of experiments to evaluate the attack performance for positive triggers and the mis-triggering rate for negative triggers.

Car Movement. For car movements, we evaluated their impact by conducting experiments with three car speeds: (1) slow (5 km/h), (2) medium (10 km/h), and (3) fast (15 km/h). We limited the highest experimental speed to 15 km/h for safety reasons. As shown in Fig. 11, we can achieve a $f_{succ}^{max(50)}$ of 100% for HA and a $f_{succ}^{max(50)}$ of 96% for CA with any car speed. The $f_{succ}^{max(150)}$ of HA and CA at high speed are lower than the rest because the number of captured frames decreases when the vehicle drives fast. On the other hand, the TPatch is hardly mis-triggered as the car moves, and the average mis-triggering rate is 0.6% for HA and 1.1% for CA, w.r.t. $f_{succ}^{max(150)}$.

Car Vibration. For car vibrations, we conduct experiments on three roads with various conditions: (1) an even brick road, (2) a straight asphalt road, and (3) an asphalt intersection. We find that the impact of car vibrations is abrupt. Most of the time, the car is in relatively stable driving conditions. As shown in Fig. 11, the TPatch has a bit better performance on the brick road than the other roads. The car vibration is possible to trigger our TPatch as long as the motion blur is similar to the positive trigger, but it is occasional and has a limited mis-triggering rate, 0.2% for HA and 2.2% for CA, w.r.t. $f_{succ}^{max(150)}$.

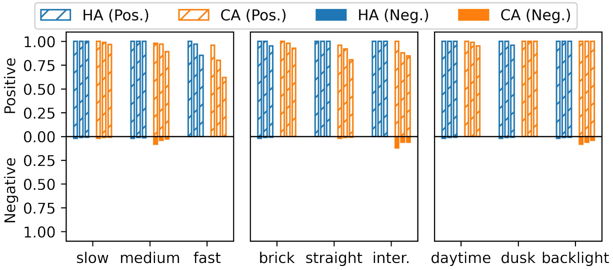


Figure 11: Impact of environmental factors on HA (blur bars) and CA (orange bars), including the speeds of vehicle (left), the road types (middle), and the lighting conditions (right). The three adjacent bars denote $f_{succ}^{max(50)}$, $f_{succ}^{max(100)}$, and $f_{succ}^{max(150)}$.

Lighting condition. Lighting conditions may change the color of the captured images and thus influence the effectiveness of TPatch. We investigate three lighting conditions, e.g., daytime (2000 lux), dusk (400 lux), and backlight (100 lux). As shown in Fig. 11, the TPatch is robust to different lighting conditions and achieves nearly 100% in all situations. It is because we have considered the possible color shifts in the EoT training of TPatch, including brightness, contrast, saturation, and hue.

6 Countermeasures and Limitation

6.1 Countermeasures

In this section, we discuss four potential defenses against TPatch from three levels, i.e., physical signal protection (sensor level), input-transformed defenses (algorithm level), model robustness improvement (algorithm level), and sensor fusion techniques (system level).

Physical Signal Protection. In this paper, we exploit the image blur caused by acoustic signals as the trigger of attacks, which is enabled by the vulnerability of MEMS inertial sensors to acoustic injection attacks. Microfiber metallic fabrics or MEMS fabricated acoustic metamaterials can be used to isolate the attacking sound. We can also design a secure low-pass filter to eliminate out-of-band analog signals, thus inhibiting the ability of the adversary to control the sensor output through signal aliasing [39].

Input-Transformed Defenses. A series of works intended to exploit the instability of adversarial examples to detect or mitigate this type of attack, typically via transformations on the input images, e.g., JPEG compression [13], bit depth reduction [43], Gaussian noise [46], median filter [43], and denoise autoencoder [27]. Due to their simplicity, these defenses are evaluated in recent relevant work like [30] and [6].

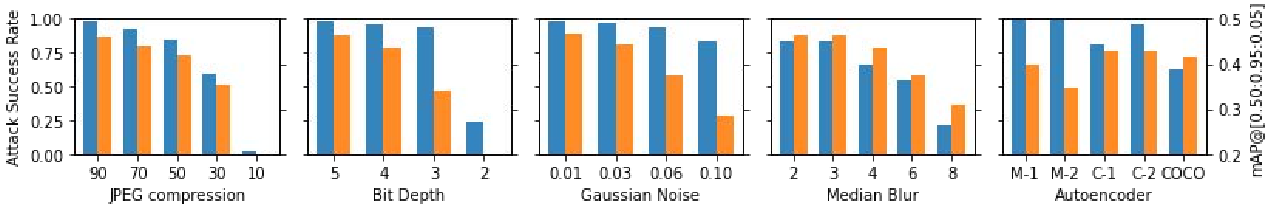


Figure 12: ASRs (blue bars) and mAPs in benign scenarios (orange bars) with the input-transformed defenses. The x-axis represents the strength of defenses for all defenses except the denoise autoencoders. The character of 'M' and 'C' denotes the MNIST and CIFAR-10, respectively, as the training dataset of autoencoders. The number '1' and '2' denotes the two different model architecture proposed in [27].

We use the ASR to evaluate the effectiveness of the defense and use the mean Average-Precision (mAP) to evaluate its impact on the performance of the detection model. As shown in Fig. 6.1, we find that with a large defense strength, the tested methods can defend our attacks to some extent but impair the performance of the detection model. For safety-related applications like self-driving, such performance drops of recognition may not be acceptable. Therefore, using the input-transformed defenses cannot easily defeat our attacks.

Model Robustness Improvement. The key factor that contributes to trigger-ability is the vulnerability of recognition models to the distortion. We can boost the resistance of the recognition model in two ways: (1) train a more robust model that tolerates the image distortion to an extent, and (2) detect the distortion via evaluating the image quality by metrics, e.g., PSNR and MS-SSIM with the last frame, and then, lower the importance of the recognition results of low-quality images.

Sensor Fusion Techniques. The general ways to cope with the abnormal situations that exist in the scenario of autonomous driving still work with TPatch, e.g., the fusion of 3d point cloud collected by Lidar and visual perception of the camera in obstacle detection, the reference of the HD maps that contain rich information about traffic signs, etc.

6.2 Limitation

TPatch attacks have the following limitations at present. First, our evaluation mainly focuses on studying the effects of TPatch attacks on the AI component level rather than the AV system level [34]. A system-level evaluation will help better understand the real-world impacts of our attacks. We will have a closed-loop simulation with the other self-driving components, e.g., prediction, planning, and control for evaluation in the future. Second, we only investigate the acoustic signal attacks as the trigger of TPatch. More evaluation on the other sensor attacks is demanded. Third, although we have enhanced the visual camouflage of TPatch with the content loss, it may still alter acute victims. Further improvement on the camouflage of TPatch is required. Fourth, the attack device for long range attacks can be noticeable at present. To improve its visual stealthiness, we can (1) employ a more compact array arrangement and (2) fit it with the surrounding environment. For instance, the scale of the attack device in the roadside scenario can be reduced into $20cm \times 20cm$ by simply eliminating gaps between the transducers. In addition, the adversary could spray the ultrasonic array in a color conform to the surroundings, e.g., green for the vegetation. We designate the aforementioned issues as our future work.

7 Related Work

In this section, we summarize the existing works on adversarial attacks from three aspects, i.e., digital noise attacks, physical adversarial attacks, and adversarial sensor attacks.

Digital Noise Attacks. It has been heavily studied how to generate adversarial examples against image classifiers by adding invisible noises, e.g., FGSM [17], PGD [24], I-FGSM [22], MI-FGSM [12] and CW attacks [7]. Other than image classification, Xie et al. [42] extend adversarial attacks to object detection and image segmentation.

Physical Adversarial Attacks. A bunch of works aim to explore the physical adversarial attacks. Kurakin et al. [22] first prove the feasibility of printed adversarial attacks. However, the subtle pixel modification limits the attack capability. Adversarial patch [5] replaces the additive noise with the placement of the patch. In [3], Expectation over Transformation (EoT) is proposed to boost the robustness of adversarial examples. Chen et al. [8] propose a robust attack, ShapeShifter, to Faster R-CNN in the real world. Song et al. [36] successfully attack both Faster R-CNN and YOLO V2. Zhao et al. [47] realize long-range attacks by adversarial example nested in the patch. Lovisotto et al. [23] render the stop sign with visible light to realize the adversarial attacks.

Adversarial Sensor Attacks. There exists another branch of researches that launch sensor attacks on the camera to achieve the adversarial goals. Yan et al. [44] demonstrate that the direct laser projection can make the camera blind or even destroyed. Man et al. [25] utilize the lens flare to manipulate the recognition of traffic signs. Sayles et al. [31] propose an optimization of color strips induced by the rolling shutter effect to mislead the image classifier. Ji et al. [20] propose adversarial blur attacks on object detectors by emitting acoustic signals to hijack the image stabilization system.

8 Conclusion

In this paper, we present TPatch, a physical adversarial patch triggered by specific physical signals to realize hiding, creating and altering attacks against the vision-based perception module in a targeted autonomous vesicle. Evaluations with three object detectors, YOLO V3/V5 and Faster R-CNN, and eight image classifiers demonstrate the effectiveness of TPatch in both the simulation and the real world. This work serves as the first attempt at the adversarial patches triggered by physical signals. Further directions include investigating other signals in addition to the acoustic ones as the triggers.

Acknowledgement

We thank our shepherd and the other anonymous authors for their valuable comments and suggestions. We would like to thank our colleagues Bo Yang and Zizhi Jin for help with the driving experiments; Kai Wang for useful discussions. This paper is supported by China NSFC Grant 62222114, 61925109, 62071428, 62271280 and China Postdoctoral Science Foundation Grant BX2021158.

References

[1] Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal for a standard default color space for the internet—srgb. In *Color and imaging conference*, number 1, pages 238–245. Society for Imaging Science and Technology, 1996.

[2] ApolloAuto. Apolloauto/apollo: An open autonomous driving platform.

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of ICML 2018*, pages 284–293. PMLR, 2018.

[4] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *Proceedings of ICLR 2018*, 2018.

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

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

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of IEEE SP 2017*, pages 39–57. IEEE, 2017.

[8] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Proceedings of ECML PKDD 2018*, pages 52–68. Springer, 2018.

[9] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *Proceedings of IEEE SPW 2020*, pages 48–54. IEEE, 2020.

[10] Common Objects in Context Dataset, 2018. https://cocodataset.org/.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of CVPR 2009*, pages 248–255. Ieee, 2009.

[12] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of CVPR 2018*, pages 9185–9193, 2018.

[13] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

[14] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH 2006 Papers*, pages 787–794. 2006.

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of CVPR 2016*, pages 2414–2423, 2016.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of CVPR 2012*, pages 3354–3361. IEEE, 2012.

[17] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of CVPR 2016*, pages 770–778, 2016.

[19] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *Proceedings of IJCNN 2013*, pages 1–8. Ieee, 2013.

[20] Xiaoyu Ji, Yushi Cheng, Yuepeng Zhang, Kai Wang, Chen Yan, Wenyuan Xu, and Kevin Fu. Poltergeist: Acoustic adversarial machine learning against cameras and computer vision. In *Proceedings of IEEE SP 2021*, 2021.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of ECCV 2016*, pages 694–711. Springer, 2016.

[22] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[23] Giulio Lovisotto, Henry Turner, Ivo Sluganovic, Martin Strohmeier, and Ivan Martinovic. {SLAP}: Improving physical adversarial examples with short-lived adversarial perturbations. In *Proceedings of {USENIX} Security 2021*, 2021.

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[25] Yanmao Man, Ming Li, and Ryan Gerdes. {GhostImage}: Remote perception attacks against camera-based image classification systems. In *Proceedings of RAID 2020*, pages 317–332, 2020.

[26] MarketsandMarkets. Automotive camera market by application, view type, technology, level of autonomy, vehicle & class, electric vehicle and region - global forecast to 2025.

[27] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of ACM CCS 2017*, pages 135–147, 2017.

[28] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2016.

[30] Takami Sato, Junjie Shen, Ningfei Wang, Yunhan Jia, Xue Lin, and Qi Alfred Chen. Dirty road can attack: Security of deep learning based automated lane centering under {Physical-World} attack. In *Proceedings of USENIX Security 2021*, pages 3309–3326, 2021.

[31] Athena Sayles, Ashish Hooda, M. Gupta, Rahul Chatterjee, and Earlence Fernandes. Invisible perturbations: Physical adversarial examples exploiting the rolling shutter effect. In *Proceedings of CVPR 2021*, 2021.

[32] Geary K Schwemmer. Doppler shift compensation system for laser transmitters and receivers, February 2 1993. US Patent 5,184,241.

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of CVPR 2017*, pages 618–626, 2017.

[34] Junjie Shen, Ningfei Wang, Ziwen Wan, Yunpeng Luo, Takami Sato, Zhisheng Hu, Xinyang Zhang, Shengjian Guo, Zhenyu Zhong, Kang Li, et al. Sok: On the semantic ai security in autonomous driving. *arXiv preprint arXiv:2203.05314*, 2022.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *Proceedings of {WOOT} 2018*, 2018.

[37] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of AAAI 2017*, 2017.

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[39] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks. In *Proceedings of IEEE EuroS&P 2017*, pages 3–18. IEEE, 2017.

[40] Ultralytics. Yolo v5.

[41] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking. In *Proceedings of {USENIX} Security 2021*, 2021.

[42] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of ICCV 2017*, pages 1369–1378, 2017.

[43] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[44] Chen Yan, Wenyuan Xu, and Jianhao Liu. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *Def Con*, 24(8):109, 2016.

[45] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of CVPR 2020*, pages 2636–2645, 2020.

[46] Yuchen Zhang and Percy Liang. Defending against whitebox adversarial attacks via randomized discretization. In *Proceedings of AISTATS 2019*, pages 684–693. PMLR, 2019.

[47] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of ACM CCS 2019*, pages 1989–2004, 2019.

A Design Details of Acoustic Signal Injection

A.1 Challenge

Previous work [39] has demonstrated that injecting acoustic signals into a MEMS motion sensor can induce a constant bias when the signal frequency is an integer multiple of the sampling frequency. Based on it, the attacker can form any signal through amplification or phase modulation. However, this method requires the attacker to monitor the sensor readings of the victim's car in real-time to synchronize the attack signals, which can be difficult to fulfill in real driving scenarios. Without signal synchronization, the sensor output may suffer from phase shifting and frequency shifting, degrading the performance of manipulation. In the following, we use the one-dimension sensor data as an example to demonstrate their effects.

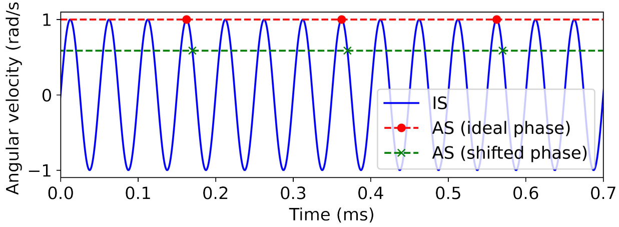


Figure 13: Phase shifting effects on the aliased signals. The blue line denotes the resonance of gyroscope induced by injected signals (IS). The red and green dot lines represent the aliased signals (AS) with different phase.

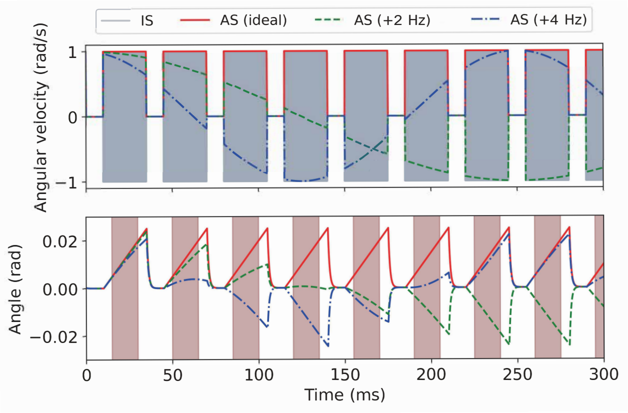


Figure 14: Frequency shifting effects on the aliased signals. The bottom figure shows the moving angles of camera which is the integral of angular velocity shown in the upper figure. The red line represents the ideal situation where the sampling frequency is constant, while the green and blue dot lines represent the consequences caused by different frequency shifts.

Phase shifting. Without synchronization, a random phase shift can be introduced during the aliasing process, as shown in Fig. 13. The random phase shift then introduces a random sensor output, and thus the motion blur with a random strength.

Frequency shifting. Similarly, when without synchronization, the sampling frequency of the MEMS sensor may drift slightly from time to time due to the temperature changes. As shown in Fig. 14, a small shift in frequency, e.g., 2 Hz or 4 Hz, can render the camera motion different from the ideal one and varied in continuous frames.

A.2 Our Approach

To address the aforementioned issues, we first model the image blur effect. A motion-blurred image can be viewed as the superposition of a series of translated clean images, which can be formulated in Eq. 13:

$$L(i,j,t) = \int_{t}^{t+t_e} g(i+\Delta i(\tau), j+\Delta j(\tau))d\tau \qquad (13)$$

where $L(i,j,t)$ denotes the j_{th} row and i_{th} column of the image L captured at the moment t. t_e is the exposure duration of the camera. $\Delta i(t)$ and $\Delta j(t)$ denote the translation of the X-axis and Y-axis of the image at the moment t. $g(i,j)$ denotes the j_{th} row and i_{th} column of the stationary image g.

Based on this formulation, we find the key to reducing the impact of phase/frequency shifting is to make the image blur independent of the time t. To achieve it, we propose to shift the frequency of the injected signal from the integer multiple of the sampling frequency nf_s to the frequency $nf_s \pm f_e$, where f_e represents the exposure frequency. In this way,

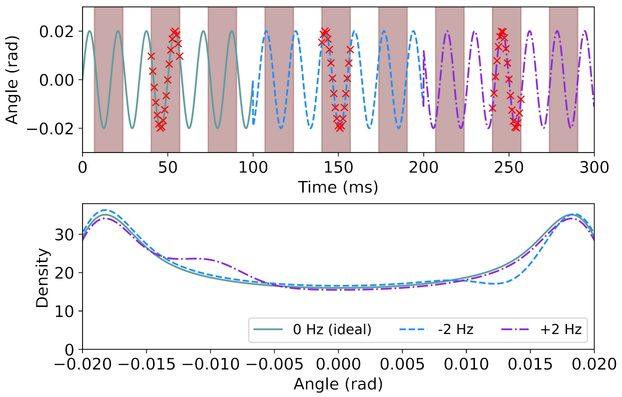


Figure 15: Aliased signals with frequency shifting. The upper figure shows the image translation caused by aliased sine signals, where the curves with different colors represent different frequency shifts and the red shadows represent the exposure duration. The bottom figure shows the density of image translation.

the frequency of the aliased signal can be given with $f_a = |f - nf_s| = |nf_s \pm f_e - nf_s| = f_e$, which equals the exposure frequency. Since the aliased signal determines the sensor output and thus the camera motion, the translation functions of the X-axis $\Delta i(t)$ and Y-axis $\Delta j(t)$ share the same period with the camera exposure duration t_e, as formulated in Eq. 14:

$$\Delta i(t) = \Delta i(t + t_e), \quad \Delta j(t) = \Delta j(t + t_e), \quad \forall t \tag{14}$$

In this way, we can inject a stable blur pattern that is time-independent as shown in Eq. 15, which eliminates the impact of phase shifting.

$$L(i, j, t_1) = L(i, j, t_2), \quad \forall t_1, t_2 \tag{15}$$

The proof of the property is provided as follows:

$$\begin{aligned}
L(i,j,t_1) &= \int_{t_1}^{t_1+t_e} g(i+\Delta i(\tau), j+\Delta j(\tau))d\tau \\
&= \int_{t_1}^{t_2} g(i+\Delta i(\tau), j+\Delta j(\tau))d\tau + \int_{t_2}^{t_1+t_e} g(i+\Delta i(\tau), j+\Delta j(\tau))d\tau \\
&= \int_{t_1}^{t_2} g(i+\Delta i(\tau+t_e), j+\Delta j(\tau+t_e))d\tau + \int_{t_2}^{t_1+t_e} g(i+\Delta i(\tau), j+\Delta j(\tau))d\tau \\
&= \int_{t_1+t_e}^{t_2+t_e} g(i+\Delta i(\tau), j+\Delta j(\tau))d\tau + \int_{t_2}^{t_1+t_e} g(i+\Delta i(\tau), j+\Delta j(\tau))d\tau \\
&= \int_{t_2}^{t_2+t_e} g(i+\Delta i(\tau), j+\Delta j(\tau))d\tau = L(i,j,t_2)
\end{aligned}$$

For frequency shifting, our approach cannot eliminate it but can reduce its impact. The reason is that the aliased signal generated by our approach has a much higher frequency than the frequency shift caused by the temperature drift, making its impact not significant. For instance, given a desired aliased signal of 60 Hz and a frequency shifting of $k = 2$ Hz, the actual aliased signal is 58 Hz or 62 Hz, which still has a high overlap with the idea one, as shown in Fig. 15

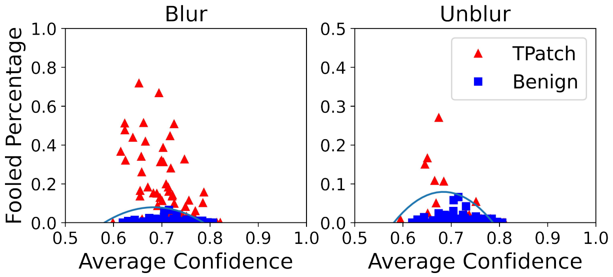


Figure 16: Visualization of SentiNet results. The red triangles and blue squares respectively denote `TPatch` and benign examples. The curve represents the decision boundary.

B Impact of Doppler Effects

In the roadside attack scenario, a relative speed can exist between the driving-by vehicle and the stationary attack device, resulting in a frequency shift of the injected signals due to the Doppler effects. Given a relative speed of 1 m/s and an emitted signal frequency of 20 kHz, the frequency shift of injected acoustic signal is 58.8 Hz. To address it, the attacker can compensate for the frequency shifting in real-time by externally measuring the speed of the vehicle via a certain common approach, e.g., LiDAR [32]. For the reason of engineering, we configure the car at a constant speed to simulate the scenario where the adversary has obtained the vehicle speed from real-time measurements.

C Explainable-AI based Detection Techniques

Recently, a few approaches are proposed to defend the attack of adversarial patches, e.g., *SentiNet* [9], *PatchGuard* [41], etc. The common methodology among existing defenses is to utilize the explainable-AI techniques, e.g., Grad-CAM [33] in *SentiNet* and BagNet [4] in *PatchGuard*, to localize the potential adversarial patch, whose saliency is abnormally high. Once the `TPatch` is triggered, it would work like the normal adversarial patch; hence, it is feasible to tackle our attack with similar approaches. We report the defense results of *SentiNet* in Fig. 16, where the accuracy of *SentiNet* is relatively low when `TPatch` is untriggered. We suppose that there are pilot systems that are equipped with these state-of-the-art detection techniques. We want to investigate whether it is possible to detect the `TPatch` via feature analysis.

We use Grad-CAM and BagNet to extract the feature saliency map of the target class. Here we make a strong assumption that the defender knows the target class of attack. If the defender is still unable to detect the presence of the `TPatch` despite having additional prior conditions, it means that this detection method is also difficult to succeed with zero prior. We follow the methodology of original papers to process the saliency map. In *SentiNet*, the defender uses a threshold to generate the binary mask and then crops the

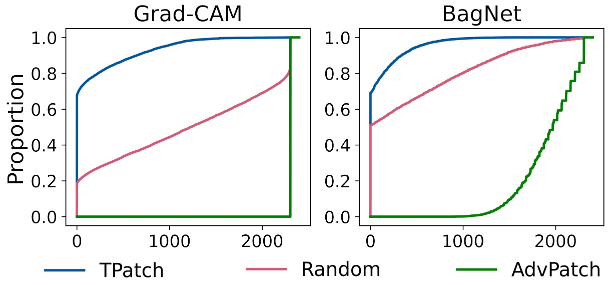


Figure 17: CDF of the number of masked pixels. We use ResNet-50 and BagNet-33 as the recognition model respectively for the experiments of Grad-CAM and BagNet.

masked part of the image, which potentially contains the adversarial patch. We keep 30% of the pixels with the highest saliency for evaluation. In *PatchGuard*, the defender uses BagNet as the recognition model, whose final convolutional layer can generate a local feature map of each class, and utilizes a sliding window to detect the most salient area. We design the TPatch as 48×48 pixels and the corresponding best window size is 6×6. After finding the most salient window, we can map the window in the feature space to the mask in the image pixel space.

We define the masked pixel numbers of the TPatch as the metric of evaluation, where zero stands for the complete failure of detection and the total pixel number of the patch stands for the success of detection. We conduct experiments on three types of patches, i.e., random patch, normal adversarial patch, and our TPatch, and plot the overall result with the format of cumulative distribution function (CDF). As shown in Fig. 17, both methods have an excellent performance in detecting the normal adversarial patch. However, both methods perform badly on detecting the TPatch and never mask the whole patch. Surprisingly, the performance of detecting the TPatch is even worse than the one of detecting the random patch, which represents the probability of accidentally masking the irrelevant objects.

D Physical Experiments on Altering Attacks

We conduct indoor physical experiments to investigate the effectiveness of altering attacks on classifiers. We place the TPatch with a mouse (an existed class in ImageNet [11]) in the view of camera and use a switch to control the acoustic signal injection. When the switch is off, the classifiers correctly recognize the mouse, and the target class, traffic light, is not within the top-5 recognition results. When the switch is on, the top-1 recognition result is altered into the traffic light. All the TPatches achieve $f_{succ}^{max(150)}$ of 100% on three tested classifiers, i.e., VGG16, Res50, and Incv3.

E Impact of Other Acoustic Signals

We conduct experiments with three representative types of acoustic signals: (1) in-car music, (2) road noises, and (3) traffic horn sounds. During attacks, we played each of them at 110dB using a speaker at a distance of 5cm from the camera. From the results, we find that none of them has effects on either the captured images or the motion sensor readings. This is because common background noises are in a lower frequency band (<10kHz) than the resonant frequencies of the motion sensors tested in our experiments, i.e., higher than 20kHz.

F Impact of Phase Differences

With the same strength A and orientation ψ of motion blur, different phase differences can make the PSF kernel various shapes, as illustrated in Fig. 18, while the phase differences have few impacts on the ASRs of TPatch, as shown in Fig. 19.

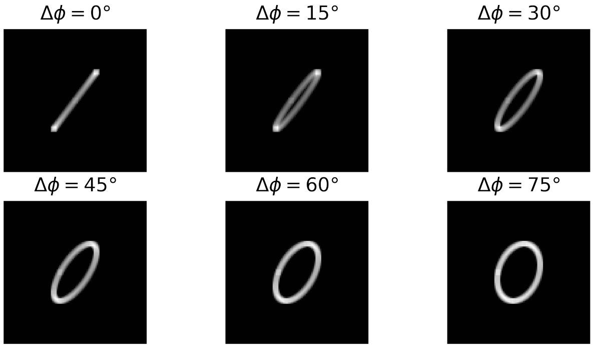


Figure 18: PSF kernels of different phase differences.

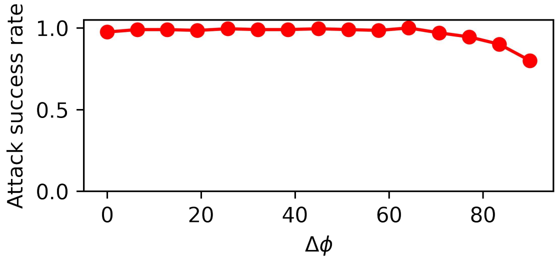


Figure 19: ASRs with different phase differences.