Artifact Evaluated
NDSS
Functional

# Targeted Physical Evasion Attacks in the Near-Infrared Domain

Pascal Zimmer
Ruhr University Bochum
pascal.zimmer@rub.de

Simon Lachnit
Ruhr University Bochum
simon.lachnit@rub.de

Alexander Jan Zielinski
Ruhr University Bochum
alexander.zielinski@rub.de

Ghassan Karame
Ruhr University Bochum
ghassan@karame.org

***Abstract*—A number of attacks rely on infrared light sources or heat-absorbing material to imperceptibly fool systems into misinterpreting visual input in various image recognition applications. However, almost all existing approaches can only mount untargeted attacks and require heavy optimizations due to the use-case-specific constraints, such as location and shape.**

**In this paper, we propose a novel, stealthy, and cost-effective attack to generate both *targeted* and *untargeted* adversarial infrared perturbations. By projecting perturbations from a transparent film onto the target object with an off-the-shelf infrared flashlight, our approach is the first to reliably mount laser-free *targeted* attacks in the infrared domain. Extensive experiments on traffic signs in the digital and physical domains show that our approach is robust and yields higher attack success rates in various attack scenarios across bright lighting conditions, distances, and angles compared to prior work. Equally important, our attack is highly cost-effective, requiring less than $50 and a few tens of seconds for deployment. Finally, we propose a novel segmentation-based detection that thwarts our attack with an F1-score of up to 99%.**

## I. Introduction

Deep neural networks are known to be susceptible to malicious inputs, which is especially relevant in safety-critical use cases, such as traffic light/sign recognition and facial recognition systems for access control and surveillance. Different attack strategies exist, some of which assume direct model access and enable direct gradient computations, i.e., white-box model [13], while others are limited to oracle access to a model (black-box model). In the digital domain, the generation of these perturbations is often constrained with an $L_p$ norm, which captures the difference between a benign and malicious image on a pixel level and is an imperceptibility measure for a (human) observer.

Recent real-world attacks exploit the specificities of camera hardware or hide perturbations in inconspicuous phenomena. Popular methods to conceal perturbations consist of the reliance on so-called adversarial patches; these have been shown to be particularly harmful in traffic sign detection [8], [10], [54], facial recognition systems [31], [37], and person detection [40], [41]. Adversarial patches are typically static, leave a visible trace behind, and are constrained in their degrees of freedom. Other methods, such as projector-based attacks, exploit the full RGB color space due to their ability to project arbitrary images with close to pixel-wise precision onto a target. Projector-based attacks, however, suffer from a major shortcoming, as the illumination required to be projected onto an object is far from being stealthy.

Other, more recent, attacks exploit the fact that the sensitivity of CMOS sensors often stretches further into the infrared part of the optical light spectrum compared to human vision, opening the door for exploitation in image recognition applications [35], [42], [55]. This problem is further exacerbated by the fact that spectral filters are often not installed in modern vehicles due to their high cost and performance overhead [42]. In these settings, infrared projectors emerge as a robust and precise means to introduce inconspicuous pixel-wise modifications. Such projectors are costly, requiring investments of tens of thousands of USD. As an alternative, several recent contributions have overcome the high cost associated with infrared projectors using infrared lasers [35], albeit at the expense of precision. More specifically, even though these approaches can cut down costs to just thousands of USD, *they cannot mount attacks targeting specific classes* since their optimization space is limited and can only reduce the model's confidence in correctly classifying input. As such, *they often result in disruptions of service (e.g., flipping the prediction to any different class) but cannot be used to mount sophisticated attacks*, i.e., precise label-flipping. In comparison, the misclassification of a stop sign as a speed limit 50 sign or vice-versa by an autonomous vehicle poses a greater safety hazard than a simple service disruption.

In this paper, we propose the first practical and robust infrared perturbation approach to mount inconspicuous *targeted* and *untargeted* attacks in the physical world. Our laser-free approach bridges the gap between powerful projector-based attacks and existing solutions by significantly reducing the complexity of the underlying optimization problem. To ensure real-world robustness, we opted to account for the spectral shift into the infrared domain (since we cannot exploit the full RGB color space). We incorporated the use of *expectation over transformation*, i.e., EOT [5], to adapt to various real-world limitations, e.g., stemming from brightness changes, perturbation misalignment, and spatial transformations. Unlike previous work [35], [43], [45], [56], [57], our approach

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.231568
www.ndss-symposium.org

considerably reduces the real-world constraints on shape and location by mimicking an infrared projector. This allows us to exploit additional degrees of freedom as a means to generate more robust, targeted, and successful perturbations compared to existing approaches. Moreover, contrary to [53], our model is not restricted to a single (artificial) light source. This particularly allows us to capture realistic deployment environments with varying lighting conditions and to realize high-accuracy targeted attacks (in addition to the standard untargeted attacks) with a negligible overhead. Namely, our attack is highly cost- and time-effective—incurring an equipment cost of less than US$50 and only tens of seconds to deploy. In summary, our contributions are as follows:

**Novel attack:** We propose a novel approach to generate adversarial infrared perturbations that alleviates many practical constraints in current proposals and can accurately mount both targeted and untargeted attacks (cf. Section IV).

**Thorough evaluation:** We evaluate and verify our adversarial infrared perturbations in both targeted and untargeted settings in the use cases of traffic sign recognition, i.e., object detection and image classification, in both digital and physical domains. Real-world experiments show that our approach results in attack success rates of up to 100% in various lighting conditions across varying distances and angles, and in a moving vehicle (up to 30 km/h), underlining the impact on real-world safety in both two-stage (cf. Section V) and single-stage architectures (cf. Section VI). For instance, our proposal improves the attack success rate by up to 20.47% compared to [43], [45], even though [45] is a white-box method with direct access to model gradients. We achieve this while requiring a considerably lower number of queries, by up to 65% (cf. Section V).

**Countermeasures:** We show that our proposal exhibits significant robustness against state-of-the-art defensive schemes (cf. Section VII). To remedy this, we propose a novel segmentation-based detection scheme that is specifically designed to address infrared perturbation attacks on traffic signs. Our experiments show that our defense can thwart infrared perturbation attacks with an F1-score of up to 99%.

**Open science:** To aid researchers in conducting real-world evaluations in the near-infrared spectrum, our source code and the first open-source infrared traffic sign dataset, which we dub *GTSRB-IR-100* (cf. Appendix B), is publicly available[1]. We also responsibly disclosed our findings to Mercedes, Mobileye, Tesla, Sony, and OnSemi.

## II. Background and Related Work

**Vision-based System Architectures:** Image recognition architectures generally fall into two categories: single-stage and two-stage [9]. Single-stage models perform object detection and classification jointly, offering efficiency for tasks with a limited number of classes, but suffer in performance as the number of classes increases. In contrast, two-stage pipelines first detect objects using a single-class detector and then classify each detected region, making them more suitable for handling a large number of classes.

[1]https://github.com/RUB-InfSec/infrared_perturbations

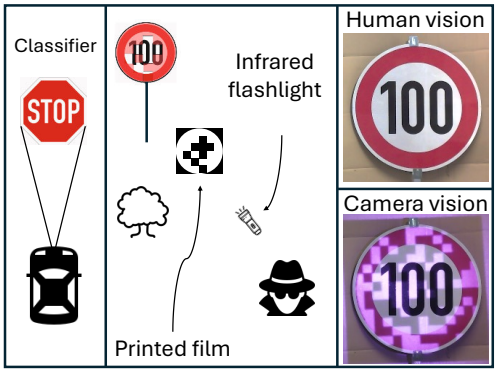



Fig. 1. Overview of our system (icons from [1]).

**Real-World Adversarial Attacks:** Adversarial patch attacks have been adapted to real-world settings by embedding visible perturbations that mimic plausible scenarios—e.g., shadows, snow, stickers, or weathered signs [8], [53], [54]. However, these attacks are often easy to spot due to the unnatural appearance of the patterns.

To increase stealth, newer attacks exploit human perceptual limitations and the characteristics of camera sensors. These include perturbations invisible to humans but detectable by cameras, or those injected via the camera pipeline, such as with modulated lighting [36], laser interference [50], ultrasonic signals [17], or EM interference [20]. Others exploit visual illusions, projecting images too briefly for human detection [30].

A particularly stealthy class of attacks leverages the camera sensor's sensitivity to infrared (IR) light. These include hidden IR patterns for evading facial recognition [55] and spoofing traffic signals with IR LEDs [42]. More recent work targets traffic sign detection using large, invisible IR laser spots [35], combining techniques from visible-light and laser-based attacks [16], [22]. Unfortunately, due to the limited solution space of possible perturbations, *all existing works can only be effective in the untargeted attack setting*.

An attacker might also resort to jamming the camera to attack the vision system with (in-)visible light. Jamming attacks are less fine-grained and disable the entire vision system. These attacks require precise, real-time targeting of the camera of a moving vehicle, making it challenging to execute.

## III. Design Goals & Approach

### A. System & Threat Model

We consider an adversary that is interested in causing vision-based recognition systems used in environments such as autonomous vehicles to output an incorrect prediction by placing an adversarial perturbation on a target object. The adversary is interested in keeping any introduced perturbation invisible to a human observer but clearly observable by a CMOS camera, thereby impacting the image processing pipeline.

Unless otherwise specified, we focus on the main use case of traffic sign recognition in this work. However, we emphasize that our approach is equally applicable to other use cases, such as facial recognition systems. More specifically, we target both single- and the more challenging two-stage pipelines to ensure that our approach broadly applies to many existing production systems. The reason that we consider this use case is that it presents an unexplored opportunity for the adversary; in most modern vehicles (e.g., Tesla Model 3), spectral filters are often not installed due to their additional cost, performance overhead, and functional limitations [21], [42]—preventing the masking of infrared perturbations in modern cars. For instance, spectral filters reduce the perception of a vision system when dealing with low-light conditions, hamper the detection of lane markings [21], etc. Therefore, many vendors explicitly opt for cameras that remain sensitive to NIR, e.g., sensing systems by Mobileye or Mercedes, or are actively developing patents that explicitly consider infrared-sensitive camera systems [6], [28]. Moreover, (i) recent camera modules, e.g., e-con Systems STURDeCAM88, are also *not* equipped with infrared filters, and (ii) cutting-edge sensor designs for day-night imaging promise improved color fidelity (mimicking the effect of infrared filters), while maintaining sensitivity to infrared light at night [26].

Throughout this paper, we assume the camera of the attack target lacks infrared filtering and the system relies solely—like recent Tesla models—on vision-based sensing. While some systems may use additional map data, this data is often missing/outdated for new roads, indoor sites, or construction zones. Even when maps are accurate, if a vehicle detects contradicting inputs, e.g., a 20 km/h sign but the map shows 80 km/h, it will likely brake.

Arguably, in this setting, the main goal of the adversary is to manipulate the output of the vision-based recognition system, e.g., by hiding signs or by classifying a stop sign as a speed sign or vice versa (e.g., to target specific manufacturers' processing pipelines [32], or to cause harm). To achieve these goals, an adversary might resort to three types of attacks: (1) a hiding attack, for which the object detector is fooled to ignore a given object; (2) a weak untargeted attack, for which an image classifier is fooled to output *any* class different from the ground truth, or (3) sophisticated targeted attacks that purposefully intend to fool the classifier into outputting a *specific* target class that is different from the ground truth (e.g., misclassify all speed signs as a stop sign).

Unlike previous work in this area [35], [43], [45], [52], we primarily focus on the challenging targeted attack setting; we, nevertheless, also analyze and evaluate the effectiveness of our approach in the untargeted setting. In contrast to camera jamming, our infrared perturbation selectively misclassifies specific signs without disrupting functionality.

To achieve these goals, we assume that the adversary has black-box score-based oracle access to the target image classifier—in which an adversary can only observe the output probabilities after supplying a controlled input. That is, we assume that the adversary neither knows the model weights, architecture, nor has access to the exact training data. This mimics a realistic setting where the adversary can only interact, e.g., with the classifier of a vehicle through a debug interface, but does not have access to the full classifier [16], [17], [22], [35], [43], [53][2].

### B. Design Criteria

To make the physical adversarial perturbations practical, our design seeks to achieve the following criteria.

**Targeted, Untargeted & Hide Attacks.** Adversarial perturbations should effectively realize targeted, untargeted, and hide attacks. In most practical deployments, hide and untargeted attacks result in service disruptions (e.g., by hiding an object or preventing correct classification). Targeted attacks, on the other hand, are more powerful as they enable the adversary to cause specific damage, such as causing autonomous vehicles to increase their speed at stop signs.

**System and Camera-Agnostic.** Since it is not feasible for an attacker to obtain the images captured from the camera of an approaching vehicle in real-time, the crafted adversarial perturbations should not be specific to a given (camera) system and should be transferable across various target classifiers.

**Scene-Agnostic.** The adversary may have to conduct the attack at dynamically changing scenes. This includes varying lighting conditions, varying distances, e.g., between a sign and an approaching vehicle, and sometimes in the presence of motion blur induced by the vehicle's movement.

**Cost-effective deployment.** The adversary is clearly interested in minimizing the amount of time and the cost required to mount such attacks. Namely, the generation of physical adversarial perturbations and their deployment should require minimal time and resources.

### C. Approach

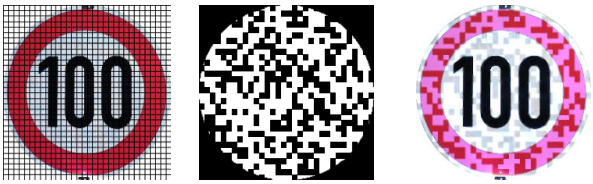
A strawman solution to create inconspicuous adversarial examples would be to rely on infrared projectors (e.g., Barco FS70-W6). Such projectors are widely used in military applications and are notorious for their ability to introduce precise pixel-wise projections. Infrared projectors are, unfortunately, costly (in the order of tens of thousands of USD).

In this work, we opt for a more efficient alternative to infrared projectors. Namely, we explore using a transparent film on which we print the perturbation combined with the mask using an off-the-shelf printer. Our primary intuition is to discreetly place the film in front of an infrared light source, allowing it to project the perturbation onto our target object (cf. Figure 1). Notice that this process precisely mimics the projection of small squares onto the target image. Our setup consists of a compact device that integrates both an infrared lamp and transparent film (approx. $10\text{cm} \times 10\text{cm} \times 20\text{cm}$) which can be *easily concealed behind bushes or junction boxes*. This is considerably more stealthy than setups used

[2]The adversary can be a user themselves to interact with the classifier to directly observe how a manipulated sign is perceived on a car's dashboard.

TABLE I
OVERVIEW OF MP PARAMETERS.

| Parameter | Description |
|---|---|
| $w$ | Width of the input image. |
| $h$ | Height of the input image. |
| $k$ | Maximum number of MP used in the perturbation. |
| $l$ | Side length of an MP in pixel. |
| $\mathcal{I}$ | Set of MP positions in the reduced coordinate space. |
| $\tilde{\mathcal{I}}$ | Set of pixel positions in the pixel coordinate space. |
| $\mathcal{M}$ | Mask to locate the target object. |
| $\mathcal{P}$ | Projection mask consisting of the mask and MP . |



(a) MP positions. (b) Perturbation mask. (c) Infrared perturbation.

Fig. 2. Definition of pixel positions ($32 \times 32$) and a concrete perturbation for $l = 7$ and an image of $w = h = 224$.

in prior works, which, for example, require bulky video projectors [25] or infrared lasers [35], that can also pose a safety hazard due to the use of lasers.

In this setting, several challenges arise to ensure that our design is scene- and system-agnostic. Namely, while the reliance on the infrared domain presents opportunities to the adversary, it effectively limits the available color space that can be utilized when creating perturbations (and increases the difficulty of successfully generating adversarial examples). Moreover, one needs to ensure that adversarial examples are efficiently created to be effective in real-world deployments practically and to adjust to environmental changes quickly; for instance, we need to cater to the fact that the CMOS camera is constantly moving (and is not fixed when compared to other use cases) and, thus, lighting and position would also vary as the vehicle approaches the traffic sign.

## IV. DESIGN

We now present our methodology for generating efficient and workable adversarial perturbations in the infrared domain.

### A. Modeling Infrared Perturbations

Unlike traditional perturbation attacks, our infrared perturbations must be created while paying special attention to the fact that there might be multiple light sources involved (i.e., an ambient and an infrared light source); this complicates the modeling process significantly. We summarize the various notations used in this paper in Table I.

**Shape.** Due to the imperfect nature of the perturbation process, we opted to move away from pixel-wise perturbations to so-called *manypixel* (MP), a grouping of several neighboring pixels. For simplicity and without loss of generality, we assume that a MP can be approximated by a square whose side length $l$ divides both the height and width of the input image, i.e., $l|h \wedge l|w$. This effectively reduces the pixel coordinate space from $w \times h$ pixels to a reduced coordinate space of $h/l \times w/l$ MP for our subsequent optimizations, as seen in Figure 2a. This matches our real-world experiments in Section V-C, where a square MP is output by projecting small pixel perturbations from a transparent film onto the target.

**Location.** We define the location of our adversarial perturbation by a set of MP positions $\mathcal{I}$. For instance, when the MP corresponds to a square, $\mathcal{I} \subseteq [0, w/l] \times [0, h/l]$. The amount of MP is denoted by $k$, i.e., $|\mathcal{I}| = k$. A mask $\mathcal{M}$ is used to locate the target object, e.g., the shape of a traffic sign and the region of the adversarial perturbation. It might happen that an MP is drawn directly on the mask. To enable a partial drawing of the MP along the contour of the mask, we define a transformation from the reduced coordinate space used for the MP back to the pixel coordinate space of the original input image. In the particular case of a square, this is achieved by adding the positions of all $l^2$ pixels within a single MP to our position set. We define the transformation $\varphi$ as follows:

$$\varphi(x, y) := [x, x + l] \times [y, y + l] \tag{1}$$

and construct the transformed coordinates $\tilde{\mathcal{I}}$ as

$$\tilde{\mathcal{I}} := \bigcup_{(x,y)\in\mathcal{I}} \varphi(x, y) \tag{2}$$

Using the aforementioned transformation, the function $\texttt{ModelPerturbation}(\mathcal{I})$ returns the projection mask $\mathcal{P}$ that corresponds to the intersection of the MP locations in pixel space and the mask, i.e., $\mathcal{P} = \texttt{ModelPerturbation}(\mathcal{I}) = \tilde{\mathcal{I}} \cap \mathcal{M}$ as shown in Figure 2b.

**Perturbation color.** We need to model the impact of the infrared light source on an object that is already lit by ambient light. Our perturbation masks the infrared light source, making the covered area appear as ambient-lit. As a result, the area outside the perturbation exhibits a color shift while the perturbed area remains unaffected. Unlike prior work that relies on a single visible light source [53], we cannot assume that perturbations can be modeled simply as brightness reductions.

Note that, in ideal conditions, cameras adjust their exposure and white balance to obtain similarly bright images, albeit being taken under different lighting conditions. However, many real-world traffic sign datasets contain overexposed or underexposed images due to complex lighting scenarios, which obscures the impact of an infrared light source. Before introducing a brightness-dependent infrared transformation, we normalize the data to ensure that we obtain similarly exposed images. This can be modeled using the three-dimensional CIELAB color space [4], which covers the entire gamut of human color perception. More concretely, the lightness channel $L$ correlates with the perceptual lightness, and we assume that an exposure adjustment only changes this channel. The $A$ and $B$ channels model the four unique colors of human perception, i.e., red, green, blue, and yellow, and remain unchanged.

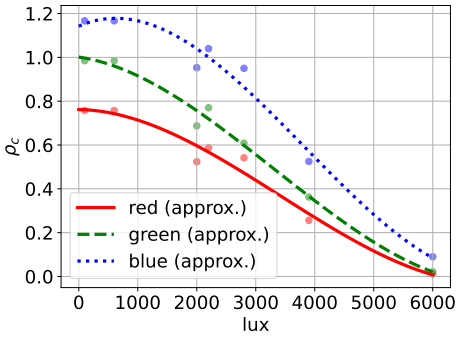



Fig. 3. Estimated channel-specific scaling factors $\rho_c$ for various ambient lighting intensities.

We define the transformation from RGB to CIELAB space as follows:

$$\mathbf{LAB} : [0, 255]^3 \rightarrow [0, 100] \times [-128, 127]^2 \tag{3}$$

$$\mathbf{LAB}(x) = [\mathbf{L}_x \ \mathbf{A}_x \ \mathbf{B}_x] \tag{4}$$

The resulting normalization adjusts the average lightness value $\bar{\mathbf{L}}_x$ of an image $x$ to the average lightness value of the data set $\bar{L}_{\text{data}}$ as follows:

$$\texttt{Normalize}(x) = \mathbf{LAB}^{-1}\left(\left[\frac{\mathbf{L}_x \bar{\mathbf{L}}_x}{\bar{L}_{\text{data}}} \ \mathbf{A}_x \ \mathbf{B}_x\right]^T\right)$$

Considering the fixed position and power of an infrared light source, the largest impact is observed on the red channel, with less pronounced effects on the green and blue channels. Due to the fixed power of the infrared light source, its effect is attenuated as the ambient lighting increases. To model the color channel of an infrared image ($\mathbf{IR}_c$) based on these effects, we take the visual color channel ($\mathbf{VIS}_c$) and apply a channel and ambient lighting-specific scaling ($\rho_c$) of the red color channel ($\mathbf{VIS}_r$). This relationship is captured in the $\mathbf{IR}$ transformation as follows:

$$\mathbf{IR} : [0, 255]^3 \rightarrow [0, 255]^3 \tag{5}$$

$$\mathbf{IR} = [\mathbf{IR}_r \ \mathbf{IR}_g \ \mathbf{IR}_b] \tag{6}$$

$$\mathbf{IR}_c = \mathbf{VIS}_c + \mathbf{VIS}_r * \rho_c \tag{7}$$

To estimate the scaling parameter $\rho_c$ for various ambient lighting intensities, we rely on empirical measurements. Here, we conducted experiments in a range of $100-6000$ lux on the surface of a traffic sign. Based on the resulting pairs of images with only the ambient lighting and an additional infrared light source, we estimated the scaling parameter as follows:

$$\rho_c = \frac{\mathbf{IR}_c - \mathbf{VIS}_c}{\mathbf{VIS}_r} \tag{8}$$

We used these data points to fit channel-specific functions, as shown in Figure 3 to perform the transformation digitally. An example of a successful transformation is shown in Figure 4. In practical scenarios, we find that the scaling parameter $\rho_c$ depends on various factors, most prominently ambient lightning. To ensure the robust generation of physical adversarial


Fig. 4. Comparison of a real-world infrared light source (right), a simulated infrared light source (center) for a traffic sign (left) from the GTSRB dataset.

examples, we account for some variation of this parameter with EOT [5] (cf. Section IV-E).

We define a function `ApplyIR` that takes an input image $x$ and the projection mask $\mathcal{P}$ and applies the infrared transformation (cf. Equation (7)) only to the parts of the image that are *not* covered by an MP as a means to prevent the infrared light from reaching the surface of the traffic sign. This is achieved using the Hadamard product as follows:

$$x' = \texttt{ApplyIR}(x, \mathcal{P}) = x \odot \mathcal{P} + \mathbf{IR}(x) \odot (1 - \mathcal{P}) \tag{9}$$

and is shown for an example in Figure 2c.

### B. Optimization for Two-Stage Architectures

We start by describing our optimization strategy for the challenging two-stage architecture. In Section IV-C, we also outline the optimization strategy for the single-stage architecture. For image classification, let $f_\theta \colon \mathbb{R}^d \rightarrow \Delta^n$ denote a DNN model, parameterized by $\theta$, assigning $d$-dimensional inputs to $n$ classes, where $\Delta^n$ is the probability simplex of $n$ classes, and let $C \colon \mathbb{R}^d \rightarrow [n]$ refer to the associated classifier defined as $C(x) := \arg\max_{i \in [n]} f_i(x)$. The dimension is equivalent to the number of pixels, i.e., $d = h \times w \times c$, with width $w$, height $h$, and number of color channels $c$. Given a genuine input $x \in \mathbb{R}^d$ predicted as $C(x) = s$ (source class), desired target class $t$, and an adversarial perturbation $\delta \in \mathbb{R}^d$, $x' = x + \delta$ is considered an *adversarial example* of $x$ if the following criterion is fulfilled:

$$\mathcal{A}(x') := \begin{cases} C(x') \neq s & \text{(untargeted attack)}, \\ C(x') = t & \text{(targeted attack)}. \end{cases} \tag{10}$$

The objective of the adversary is then expressed with the following margin loss function [7]:

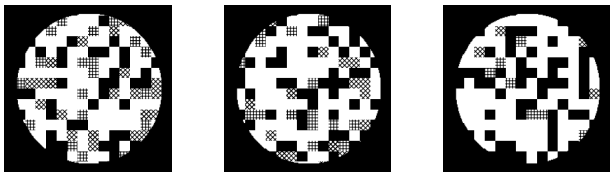
$$\mathcal{L}_{adv}(x) := \begin{cases} f_s(x) - \max\limits_{i \neq s} f_i(x) & \text{(untargeted attack)}, \\ \max\limits_{i \neq t} f_i(x) - f_t(x) & \text{(targeted attack)}. \end{cases} \tag{11}$$

For an adversarial example $x'$ to be successful, we require that there is a perturbation $\mathcal{P}$ that satisfies $\mathcal{L}_{adv} < 0$ to achieve an (un)targeted misclassification. The optimization problem is defined as follows

$$\min \mathcal{L}_{adv}(\texttt{ApplyIR}(\texttt{Normalize}(x_{\text{input}}), \mathcal{P})) \tag{12}$$

$$\text{s.t.} \quad \mathcal{P} = \texttt{ModelPerturbation}(\mathcal{I}) \tag{13}$$

to find an optimal set of MP positions $\mathcal{I}$ for a given input image $x_{\text{input}}$. Based on Equation (10) and the subset of a



(a) It. 0, $k = 64$ (b) It. 20, $k = 52$ (c) It. 160, $k = 13$

Fig. 5. Selected iterations from our approach. Crossed hatches indicate the addition of a MP from a previous iteration, while diagonally crossed hatches indicate a removal of a MP from a previous iteration.

given dataset $\mathcal{S}$, we define the attack success rate (ASR) for the digital and physical experiments as follows:

$$\text{ASR} := \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \mathbf{1}(\mathcal{A}(x')) \tag{14}$$

with output $x'$ of $\texttt{ApplyIR}(x_{\text{input}})$, adversarial criterion $\mathcal{A}$, and the indicator function defined as $\mathbf{1}(x) = 1$ if $x$ is true and $\mathbf{1}(x) = 0$, otherwise.

In contrast to existing works that need to perform complex modeling of adversarial shapes and locations [35], [43], [45], [56], [57] to remain inconspicuous and/or easily manufacturable, due to our large perturbation space, we can resort to the well-known random search algorithm [33] for the derivative-free optimization of our problem. Since, in our use case, it may not be possible to simply extract a model from the automotive system for more powerful attacks, we opt to investigate a more realistic threat model, treating our system as a black box, which is generally considered more applicable to real-world deployments.

As mentioned earlier, we approximate a MP with a square (see Section IV-A for the justification) and proceed to find a (locally) optimal set of MP positions that cause an (un)targeted misclassification. To this end, we randomly perturb a total of $k$ different MP , i.e., $\mathcal{I} \leftarrow \mathcal{U}(0, {}^{h}/_{l}) \times \mathcal{U}(0, {}^{w}/_{l})$, throughout a maximum of $Q$ queries to the classifier, with $\mathcal{U}$ denoting the uniform distribution. Naively perturbing the MP until we converge to a successful adversarial example is intractable, especially in the targeted attack setting. Instead, we only update the current best set of MPs if the new set of MPs results in a lower loss. To improve convergence, we exponentially decrease the number of changed MP based on the current iteration. Initially, we change the largest number of MP to significantly reduce the loss (cf. Equation (11)), while subsequent queries with fewer changed MP are intended to refine the loss. The schedule for deriving the number of perturbed MP for a given iteration $i$, query budget $Q$, and maximum number of perturbed MP $k$ is defined as follows:

$$k(i) = \left\lceil \frac{k}{2} e^{\frac{\ln \frac{2}{k}}{Q} \cdot i} \right\rceil$$

This schedule is used to randomly draw $k(i)$ new MP from the image. Before adding them to the set of indices $\mathcal{I}$, we randomly remove $k(i)$ MP from this set. Once we obtain a negative loss, the goal of (un-)targeted misclassification is achieved. The overall algorithm for generating infrared perturbations is shown in Algorithm 1. Given an initial image, this algorithm outputs the perturbation mask $\mathcal{P}$ and the resulting infrared adversarial image $x'$. The process of optimizing the MPs is shown in Figure 5.

**Algorithm 1:** Generating infrared perturbations

**Data:** input $x$, loss $\mathcal{L}$, max query $Q$, number of MP $k$, MP size $l$
**Result:** candidate for minimizing $\mathcal{L}$

```
/* Initialize first candidate                          */
I ← U(0, h/l) × U(0, w/l); |I| = k
P = ModelPerturbation(I), x' = ApplyIR(x, P)
L* ← L(x'), i ← 0
while i < Q do
    /* Update positions of perturbation                */
    I' ← U(0, h/l) × U(0, w/l); |I'| = k(i)
    I'' ← randomly select k(i) indices from I
    I' ← (I \ I'') ∪ I'
    P' = ModelPerturbation(I'), x' = ApplyIR(x, P')
    /* Upon improvement, update solution               */
    if L(x') < L* then
        L* ← L(x'), I ← I'
    end
    /* Negative loss: success                          */
    if L* < 0 then
        break
    end
    i ← i + 1
end
P = ModelPerturbation(I), x' = ApplyIR(x, P)
return P, x'
```

**Comparison of Optimization Strategies.** To confirm the superiority of our optimization, we now compare the ASR and average queries achieved by various popular optimization strategies, i.e., local random search (LRS), particle swarm optimization (PSO), genetic algorithms (GA), and evolution strategies (ES) for an ambient light setting of 10 lux and ablate the number of MP $k$ of size $l = 1$ in an untargeted attack setting. We compare our results against a baseline consisting of a naive random strategy (RND) that randomly places up to $k$ MP . Our results are depicted in Figure 6 for GTSRB [39] with 25 samples for each of the 43 classes.

We observe that *local random search* results in the highest ASR over all perturbation counts $k$ with an average of $90.6\%$ (i.e., twice as high compared to a random positioning of perturbations) and with as few as $123.4$ queries.

### C. Optimization for Single-Stage Architectures

To detect objects, let $f_\theta \colon \mathbb{R}^d \to \{(\texttt{bbox}, \Delta^n)\}^m$ denote a DNN model, parameterized by $\theta$, assigning $d$-dimensional inputs to $m$ bounding boxes bbox, each with a probability simplex $\Delta^n$ of $n$ classes. Without loss of generality, we focus on an image with one bounding box. We let $D \colon \mathbb{R}^d \to \{(\texttt{bbox}, [n])\}$ refer to the associated detector which is defined as $D := \{(\texttt{bbox}, \arg\max_{i \in [n]} f_i(x))\}$ for each bounding box for which the maximum probability is above the detection threshold $\tau$, i.e., $\max_{i \in [n]} f_i(x) > \tau$.

We consider a genuine input $x \in \mathbb{R}^d$ predicted as $D(x) = \{(\texttt{bbox}, s)\}$ (source class) and define the adversarial criterion

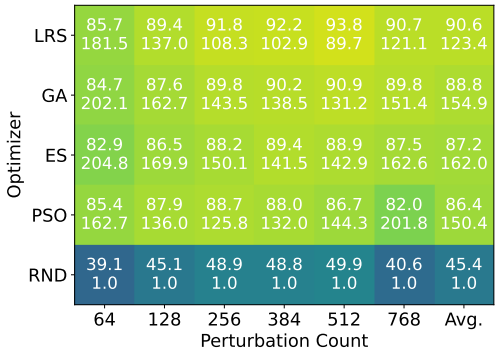



Fig. 6. Optimization results for a two-stage architecture on the GTSRB dataset. Each cell contains the ASR and query count. The last column includes the averages.

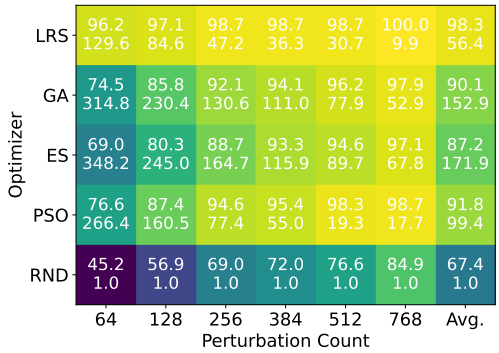



Fig. 7. Optimization results for a single-stage architecture on the Mapillary dataset. Each cell contains the ASR and query count. The last column includes the averages.

and objective for a hide attack, i.e., not recognizing the bounding box of the source class, as follows:

$$\mathcal{A}(x') := \begin{cases} \{(\texttt{bbox}, s)\} \notin D(x') \end{cases} \tag{15}$$

$$\mathcal{L}_{adv}(x) := \begin{cases} \max_{i \in [n]} f_i(x) - \tau \end{cases} \tag{16}$$

For an adversarial example $x'$ to be successful, we require that there is a perturbation $\mathcal{P}$ that satisfies $\mathcal{L}_{adv} < 0$ to achieve a hide attack. The optimization problem is then identical to the previous case of image classification (cf. Equation (12)). We use Equation (14) to compute the attack success rate.

To evaluate the impact of the optimizer selection, we evaluated the ASR and average consumed queries achieved with various popular optimization strategies on the YOLOv8 model [11] trained on the Mapillary [9] dataset. Our results are depicted in Figure 7. Here, we use 25 samples for each of the 9 classes, i.e., speed limits and stop, with a total of 225 images (cf. Section VI). In line with our previous results, the local random search optimization algorithm performs best, with an average attack success rate of 98.3% and an average of 56.4 consumed queries, outperforming the second-best optimization procedure, particle swarm optimization, with an average ASR of 91.8% and an average of 99.4 queries.

### D. Validating our Infrared Model

To validate our model, we created an (open-source) infrared traffic sign dataset. Our dataset comprises images of traffic signs with varying levels of ambient lighting, both with and without an additional infrared light source. We include additional details about our dataset in Appendix B. We compare the success of our approach on (1) the real infrared images and (2) the emulated infrared light stemming from our digital transformation in Section IV-A. We conducted our experiments with $l = 14$, in line with Section V-C.

As shown in Table II, our results on the real-world dataset show an across-the-board ASR of 100% with an averaged consumed queries as low as 21.7 for $k = 96$. For our simulated infrared light source, we observe the highest ASR at $k > 128$ with 94%, while the lowest observed rate ranks at 88%. Our results, however, confirm that our infrared transformation provides a *tight worst-case emulation of the real-world*. This also means that we expect our approach to yield, on average, better success rates in the real world when compared to the digital world.

### E. Real-World Physical Perturbations

To ensure robustness of adversarial examples under real-world conditions, we rely on expectation over transformation (EOT) [5] that finds a perturbation over the expected value of all transformed inputs over the set of transformations $\Omega$:

$$\begin{aligned} &\min \mathbb{E}_{\omega \sim \Omega}[\mathcal{L}_{adv}(\omega(\texttt{ApplyIR}(x_{\text{input}}, \mathcal{P})))] \\ &\text{s.t.} \quad \mathcal{P} = \texttt{ModelPerturbation}(\mathcal{I}) \end{aligned}$$

We model only reasonable effects with justified value ranges, as overly complex transformations result in difficult convergence towards a suitable adversarial example. More concretely, $\Omega$ includes transformations for the following effects [25]:

**Perspective.** Traffic signs are typically placed on the right side of a street (in countries with right-hand traffic) at a typical height of 2m in Europe and 5-7 ft in the US. We assume that the camera is placed at an average height of a European vehicle of 1.5m. As a result, we consider both an $x$-axis and $y$-axis perspective transformation of $\pm 35\,\text{deg}$.

**Distance.** As a vehicle approaches a traffic sign, the initially small sign gets larger over time in the captured images. This results in an initial upsampling, followed by a subsequent downsampling once the sign is too large for the network to process. For the lower bound on distance, we determine the minimum sign size for which the DNN can still correctly classify a given sign. As a result, we determine the minimum size to be $18 \times 18$ pixel.

**Rotation.** Traffic signs are typically mounted straight, i.e., the horizontal sign axis is perpendicular to the street. Due to imperfect mounting, we tolerate rotations of $\pm 6°$.

**Brightness.** To account for the slight over-/underexposure of a camera, we also utilize the LAB color space to model a brightness change (cf. Equation (3)). We consider a value range of $\pm 20\%$ in the lightness $L$ channel of the image.

TABLE II
COMPARISON OF ASR AND AVERAGE CONSUMED QUERIES $Q$ ON REAL-WORLD AND SIMULATED INFRARED PERTURBATIONS BASED ON OUR MODEL IN SECTION IV-A.

| $k$ (#MP ) | Real-World ASR | Real-World $Q$ | Simulated ASR | Simulated $Q$ |
|---|---|---|---|---|
| 16 | 100.0 | 36.94 | 88.0 | 164.38 |
| 32 | 100.0 | 41.28 | 88.0 | 144.84 |
| 64 | 100.0 | 21.70 | 90.0 | 117.44 |
| 96 | 100.0 | 11.20 | 90.0 | 125.38 |
| 128 | 100.0 | 41.98 | 94.0 | 77.32 |
| 192 | 100.0 | 17.66 | 94.0 | 97.38 |

**Backgrounds.** In single-stage pipelines, the background can significantly influence the model's output. To ensure robustness across various settings, we place the sign against a variety of backgrounds.

**Alignment and Motion Blur.** Due to the required alignment of the perturbation onto the traffic sign, we consider a shift in the $x$- and $y$-axis of $\pm 5$ pixel. We also introduce motion blur to mimic blur on frames of a moving camera.

To implement our setup in Figure 1, the film must be carefully aligned with the light source, which can be efficiently done using an infrared camera as a viewfinder. We also added a 3D-printed magnetic frame to prevent film bending and projection distortions.

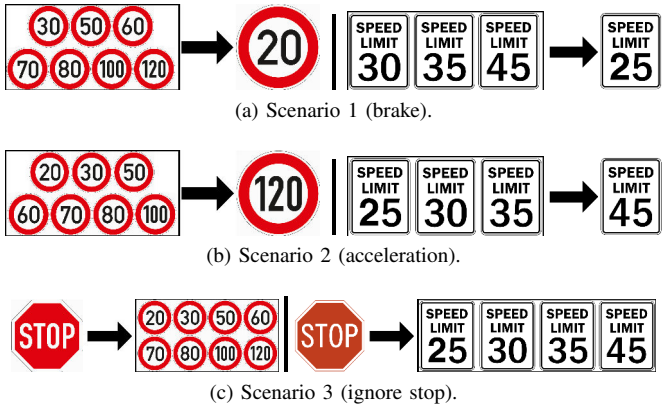
## V. EXPERIMENTS ON TWO-STAGE ARCHITECTURES

In this section, we empirically evaluate our approach for traffic sign recognition in the digital and physical domains.

**Datasets and Models:** We conducted our experiments on established datasets for traffic sign recognition for two-stage architectures: we rely on GTSRB [39] for German traffic signs and LISA [29] for American traffic signs. For the underlying model architectures, we use a simple CNN [49] for GTSRB and LISA-CNN (taken from the cleverhans library [12]), which is in line with previous works in this field [10], [25], [53]. In our normalized test sets, we report a clean accuracy (CA) of $98.76\%$ and $99.63\%$ for GTSRB and LISA, respectively.

### A. Targeted Attacks in the Digital Domain

Our evaluation in the digital domain emulates physical attacks in the real world using the GTSRB and LISA datasets. We start by evaluating our approach in the more challenging targeted attack scenario, where an adversary seeks to ensure that the prediction only flips to a *specific* class. In Section V-B, we also discuss the effectiveness of our approach in the untargeted setting. For a meaningful evaluation of a (semi-)autonomous vehicle, we define the following three driving scenarios that result in prominent safety hazards, especially when triggered in a *targeted* manner. Concretely, they result in a reduction of speed, i.e., braking, acceleration, or the ignoring of a stop sign, because a speed sign is recognized. We show them in Figure 8.



(a) Scenario 1 (brake).

(b) Scenario 2 (acceleration).

(c) Scenario 3 (ignore stop).

Fig. 8. Overview of the targeted class flips in our scenarios with European traffic signs on the left and North American traffic signs on the right.

**Scenario 1 (Brake).** We use all speed sign classes (except the lowest one) as source images and generate an adversarial example for each that classifies it as the lowest speed.

**Scenario 2 (Acceleration).** We use all speed sign classes (except the highest) as source images and generate an adversarial example for each that classifies it as the highest speed.

**Scenario 3 (Ignore stop).** We use the stop sign class as source images and generate adversarial examples for the eight speed signs that we consider in this work (cf. Figure 8a).

To avoid bias toward specific geographic regions, we included traffic signs from both Europe (via GTSRB) and North America (via LISA). This approach ensured our analysis captured a diverse range of signs—with varying shapes, colors, and sizes—across both digital and physical experiments. More precisely, for GTSRB, we relied on the stop sign and speed limits of 20, 30, 50, 60, 70, 80, 100, and 120 km/h, with 150 samples for each class. Analogously for LISA, we opted to take all samples for the selected classes due to the generally smaller dataset—here, we used speed limits 30, 35, and 45 to map to limit 25, speed limits 25, 30, and 35 to map to the highest limit of 45, and mapped the stop sign to speed limits 25, 30, 35, and 45, for the three scenarios, respectively.

Our results (cf. Table III) indicate that our proposal consistently obtains a high ASR across all three targeted attack scenarios, datasets, number of MP $k$, and strength of the ambient lighting, with up to $96.1\%$ and $100\%$ in Scenario 1, $99.9\%$ and $100\%$ in Scenario 2, and up to $76.42\%$ and $98.49\%$ in Scenario 3, for GTSRB and LISA, respectively. For LISA, we observe a slower decline in ASR , which we attribute to its fewer classes and their size in contrast to GTSRB.

Our results also suggest that the targeted class flip from a stop sign to any speed sign, i.e., Scenario 3, is the most challenging to achieve, likely due to the dissimilarity between the two sign types. In contrast, the scenarios involving similar signs, i.e., Scenarios 1 and 2, appear to be easier to realize in terms of higher ASR and fewer required queries. For GTSRB and Scenario 2, we obtain a high ASR $> 99\%$ and a low query count of 28.3 queries on average. For Scenario 1, we also observe a high ASR of up to $96.1\%$ at 127.9 queries.

TABLE III

RESULTS FOR THE FIVE ATTACK SCENARIOS. ATTACK SUCCESS RATE AND THE AVERAGE QUERIES $Q$ UNDER VARYING BRIGHTNESS CONDITIONS FOR FIXED $k = 192$ MPS AND FOR VARYING NUMBER $k$ OF MPS FOR A FIXED BRIGHTNESS OF 2000 LUX. HERE, $l = 2$.

| | | **Two-Stage Architecture** | | | | | | | | | | | | | | **Single-Stage Architecture** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Targeted** | | | | | | | | | | | | **Untargeted** | | | | **Hide** | | | |
| | | **Scenario 1** Any speed → Lowest speed | | | | **Scenario 2** Any speed → Highest speed | | | | **Scenario 3** Stop → Any speed | | | | **Scenario 4** Any sign → Any sign | | | | **Scenario 5** Sign → No sign | | | |
| | | **GTSRB-CNN** | | **LISA-CNN** | | **GTSRB-CNN** | | **LISA-CNN** | | **GTSRB-CNN** | | **LISA-CNN** | | **GTSRB-CNN** | | **LISA-CNN** | | **YOLOv8** | | **Faster-RCNN** | |
| | | ASR | $Q$ | ASR | $Q$ | ASR | $Q$ | ASR | $Q$ | ASR | $Q$ | ASR | $Q$ | ASR | $Q$ | ASR | $Q$ | ASR | $Q$ | ASR | $Q$ |
| **Lux** | 10 | 96.10 | 127.90 | 100.0 | 16.33 | 99.90 | 28.31 | 100.0 | 31.11 | 76.42 | 376.13 | 98.49 | 91.05 | 95.07 | 68.31 | 97.95 | 71.56 | 100.0 | 3.23 | 100.0 | 4.97 |
| | 1000 | 95.62 | 135.36 | 100.0 | 15.02 | 99.79 | 28.92 | 100.0 | 31.33 | 73.42 | 400.05 | 98.42 | 103.76 | 94.88 | 70.62 | 97.81 | 77.68 | 100.0 | 3.21 | 100.0 | 4.38 |
| | 2000 | 92.48 | 177.03 | 100.0 | 22.80 | 99.58 | 32.10 | 100.0 | 45.41 | 69.17 | 459.4 | 97.32 | 152.76 | 93.58 | 86.32 | 96.42 | 103.58 | 100.0 | 8.79 | 100.0 | 11.18 |
| | 3000 | 81.43 | 300.15 | 100.0 | 39.13 | 95.94 | 89.87 | 99.51 | 70.58 | 57.17 | 589.18 | 90.66 | 297.81 | 89.02 | 146.33 | 93.86 | 167.98 | 99.16 | 19.33 | 98.53 | 39.07 |
| | 4000 | 41.05 | 671.71 | 100.0 | 127.47 | 62.71 | 456.30 | 97.57 | 177.30 | 15.08 | 907.48 | 59.34 | 576.84 | 67.91 | 367.57 | 81.71 | 355.62 | 95.40 | 79.34 | 91.91 | 167.60 |
| | 5000 | 4.67 | 964.87 | 62.80 | 585.45 | 10.42 | 922.86 | 56.80 | 644.25 | 1.25 | 995.47 | 25.76 | 876.95 | 28.56 | 747.15 | 40.75 | 725.77 | 84.52 | 301.02 | 56.25 | 596.34 |
| **Patches ($k$)** | 16 | 57.14 | 559.79 | 99.39 | 70.66 | 87.19 | 260.26 | 99.03 | 109.45 | 59.08 | 490.19 | 87.71 | 287.74 | 86.88 | 172.60 | 87.93 | 234.34 | 98.33 | 55.33 | 96.69 | 79.44 |
| | 32 | 76.76 | 363.14 | 100.0 | 35.70 | 95.10 | 135.89 | 100.0 | 58.03 | 63.50 | 441.24 | 95.67 | 175.91 | 90.60 | 121.12 | 94.59 | 150.15 | 99.16 | 22.15 | 100.0 | 23.36 |
| | 64 | 88.67 | 218.16 | 100.0 | 24.06 | 98.44 | 71.78 | 100.0 | 39.42 | 67.00 | 418.08 | 97.53 | 127.67 | 92.84 | 96.76 | 96.56 | 105.23 | 99.16 | 13.83 | 100.0 | 6.37 |
| | 96 | 90.95 | 192.13 | 100.0 | 20.01 | 98.65 | 55.78 | 100.0 | 32.84 | 68.25 | 427.06 | 97.87 | 120.78 | 93.67 | 87.65 | 96.63 | 97.19 | 100.0 | 9.81 | 100.0 | 5.83 |
| | 128 | 92.57 | 172.00 | 100.0 | 17.71 | 98.85 | 47.11 | 100.0 | 37.61 | 68.33 | 438.48 | 97.87 | 128.01 | 93.67 | 86.66 | 97.15 | 91.65 | 100.0 | 9.15 | 100.0 | 5.21 |
| | 192 | 92.48 | 177.03 | 100.0 | 22.80 | 99.58 | 32.10 | 100.0 | 45.41 | 69.17 | 459.40 | 97.32 | 152.76 | 93.58 | 86.32 | 96.42 | 103.58 | 100.0 | 8.79 | 100.0 | 11.18 |

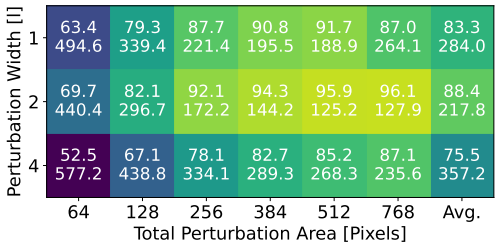



Fig. 9. Perturbation width $l$ vs. amount of perturbed pixels on GTSRB for Scenario 1 with the average in the last column. Each cell contains the ASR and query count.

TABLE IV

DETAILED RESULTS FOR SCENARIO 1 FOR GTSRB WITH 2000 LUX AND $k = 192$ USED TO COMPUTE THE AVERAGE ASR IN TABLE III.

| Sign mapping | ASR |
|---|---|
| $30 \rightarrow 20$ | 99.30 |
| $50 \rightarrow 20$ | 91.33 |
| $60 \rightarrow 20$ | 81.33 |
| $70 \rightarrow 20$ | 99.33 |
| $80 \rightarrow 20$ | 92.00 |
| $100 \rightarrow 20$ | 88.67 |
| $120 \rightarrow 20$ | 95.33 |

**Impact of ambient light intensity:** We now evaluate the performance of our proposal on the GTSRB and LISA datasets in the presence of a light source of varying intensity, ranging from 10 to 5000 lux, simulating a range from a dark to a brightly lit outdoor environment. Our results for this ablation are included in the upper half of Table III. Here, we measure the ASR and the average consumed queries required in our approach for a combination of lux $\in \{10, 1000, 2000, 3000, 4000, 5000\}$ and $k = 192$ (for the reasoning why, see next paragraph).

In the case of GTSRB, we mostly observe ASR of more than $90\%$ for lux values below 2000, except for the most challenging Scenario 3, where we reach an ASR around $70\%$. In contrast, LISA exhibits high success rates of more than $\sim 90\%$ for values lower than 3000 lux. For all three scenarios, we find the best lux setting for both datasets at 10 lux for an ASR of $96.1\%/99.9\%/76.42\%$ at an average of $127.9/28.31/376.13$ consumed queries for GTSRB. In the case of LISA, we observe $100\%/100\%/98.49\%$ at $16.33/31.11/91.05$ average consumed queries for LISA. The higher complexity for optimizing on GTSRB is also evidenced by the generally lower ASR and higher number of consumed queries compared to LISA, which confirms our previous observations. A core strength of our proposal lies in the modest number of required queries for convergence toward a successful *targeted* adversarial example, e.g., 127.9 queries for 10 lux in Scenario 1 in GTSRB.

**Impact of number of MP k:** In the lower half of Table III, we vary the number $k$ of MPs between $16 - 192$ (out of the maximum of 256 MPs) on the three proposed scenarios in a bright environment of 2000 lux. Recall that $k$ impacts the area covered by the perturbation. Generally, we observe that a varying $k$ can significantly boost the ASR by $\sim 35\%$ and reduce the consumed queries by $\sim 70\%$. In contrast, the attack success in the second and third scenarios is boosted by up to $12\%$. In the case of GTSRB, we observe that a minimum of $k = 96$ is required to obtain an ASR of $> 90\%$ for the first two scenarios, reaching its maximum at $k = 192$ at approximately $93\%/100\%$ for Scenarios 1 and 2, while reaching $70\%$ in Scenario 3. Generally, we observe that $k = 192$ strikes a strong tradeoff between ASR and the query budget in all scenarios.

**Impact of size of MP $l$:** Recall that the size $l$ of a MP is directly proportional to the maximum number of $k$ MP we can perturb. Since the number of MPs is bounded by the constant size of our samples $w = h = 32$, changing the size of an MP results in an upper bound on the total number of MP. For instance, consider the following configurations that all have the

same number of underlying pixels while exhibiting a different number of MPs:

- $l = 1, k = 256 \rightarrow \frac{32}{1} \times \frac{32}{1} = 1024$ MP positions. $l^2 \times k = 1^2 \times 256 = 256$ perturbed pixels.
- $l = 2, k = 64 \rightarrow \frac{32}{2} \times \frac{32}{2} = 256$ MP positions. $l^2 \times k = 2^2 \times 64 = 256$ perturbed pixels.
- $l = 4, k = 16 \rightarrow \frac{32}{4} \times \frac{32}{4} = 64$ MP positions. $l^2 \times k = 4^2 \times 16 = 256$ perturbed pixels.

Therefore, we opted not to evaluate against $k$ but instead to benchmark against the number of perturbed pixels (see the above example). Our results are shown in Figure 9 for a targeted attack on GTSRB for Scenario 1. We observe an average ASR of $83.3\%$ at $l = 1$, which increases to $88.4\%$ for $l = 2$ and $75.5\%$ at $l = 4$. For the standard case of $l = 1$, i.e., in the case of pixel-wise perturbations, we observe the highest performance at $k = 512$ with an ASR of $91.7\%$, which again decreases for a larger number of perturbed pixels, i.e., $768$, down to $87.0\%$. In contrast, we observe that a larger size $l$ is favorable due to the better performance of the attack. Particularly, we see that $l = 2$ strikes the best tradeoff between the size of a MP and the resulting available number of MP $k$, where we observe the highest average ASR of $88.4\%$ and the highest overall ASR of $96.1\%$ for a total of $768$ perturbed pixels. Analogously to the ASR , we also observe a minimum of $217.8$ consumed queries on average for this configuration. This number of perturbed pixels results in a value of $k = 192$ for $l = 2$ (see previous paragraph).

**Impact of sign choice:** Our approach is inherently general and does not exploit specific traffic sign shapes, colors, or textures. Notably, the choice of sign pairs has only a minor effect on ASR , as detailed in Table IV.

To confirm this intuition, we further extended our experiments beyond Scenarios 1–3—which already feature a diverse range of signs—to include several more dissimilar pairs. Specifically, we performed a targeted attack on the GTSRB priority road sign, aiming to misclassify it as a yield sign, a road construction sign, and a speed limit sign (30/120 km/h). As shown in Table V, our method achieved a strong ASR of up to 98% and an average of around 300 consumed queries. In addition, we consider a yield sign as a source and aim to classify it as a priority road, a road construction, and a speed limit sign (30/120 km/h) and obtain an ASR of up to 85% with an average of 500 consumed queries.

### *B. Untargeted Attacks in the Digital Domain*

We now move our focus towards an untargeted scenario.

**Scenario 4 (Service disruption).** We use "any" sign as the source class and "any" sign as the target class. A class flip here can lead to a sudden stop, acceleration, or any other behavior triggered by a specific sign.

Our results for Scenario 4 are shown in the fourth column of Table III for various brightness levels and a varying number of MP . When compared to its targeted counterpart (cf. Section V-A), we observe a less steep decline in ASR for the untargeted setting, combined with a slower increase in the number of queries for GTSRB, while both the ASR and number of queries for LISA are relatively similar. This trend highlights the increased difficulty in mounting targeted attacks compared to their untargeted counterparts.

In the case of GTSRB, we observe that a minimum of $k = 32$ is required to obtain an ASR of $\sim 90\%$. Subsequent increases to $k = 64$ result in a further boost of ASR by $3\%$, which only marginally increases beyond that for larger values of $k$. We consistently observe ASRs of more than $\sim 90\%$ for lux values below 3000, reaching $95.07\%$ at the lowest ambient lighting of 10 lux at just $68.31$ consumed queries for GTSRB. In contrast, we observe an ASR of $97.95\%$ and $71.56$ consumed queries for LISA. With a comparable ASR , we note that the number of queries required in LISA is slightly higher than that required in GTSRB. We contrast this to the previous trend, for which LISA performed better in terms of ASR and consumed queries, and attribute this to the fact that our sample size for this untargeted scenario is larger than the targeted scenarios before.

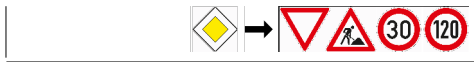
**Blackbox transferability:** To confirm that our approach is also effective on other architectures and to model an adversary without oracle access to the model, we now assess the transferability of our scheme to different architectures on the GTSRB dataset. Here, we use models of increasing complexity as surrogate models and generate the adversarial perturbations, which we subsequently evaluate on the target architecture for Scenario 4. We consider the following architectures (with the respective number of weights): GTSRB-CNN ($\sim 16.5$M), ResNet-50 [15] ($\sim 25.5$M), SwinTransformer [23] ($\sim 87.7$M), and ConvNeXt [24] (88.5M). Our experiments are conducted for 10 lux, $k = 192, l = 2$, and a query budget of $Q = 1000$. Unlike our previous experiments, where we stopped the attack once an adversarial example was found, we utilized the entire query budget here to more accurately assess the robustness of the perturbation. Our results, summarized in Table VI, show that the success of our attack is independent of the underlying model architecture. Specifically, for the same surrogate and target models, we consistently achieve success rates of over $95\%$. We observe higher transferability rates from the more complex architectures towards the simpler ones, i.e., the first column shows an average transferability of $\sim 72\%$ towards the simplest architecture GTSRB-CNN. On the other hand, when using GTSRB-CNN as the surrogate model, we observe a transferability of $\sim 63\%$ to the more complex architectures.

**Comparison with Related Work:** We now compare our approach against the state-of-the-art methods of [25], [43], [45] using the GTSRB dataset. The former two attacks are black-box methods based on transferability and gradient-free particle swarm optimization [18], respectively, while the latter is a white-box method with direct model access and, as such, requires the availability of model gradients.

We adapt [43], [45] to a two-stage pipeline and optimize our loss functions to evaluate the effectiveness of their shape-generation strategies in Scenario 4. We instrument our approach with $k = 192$ and $l = 2$, as determined in the

TABLE V

RESULTS FOR A TARGETED ATTACK ON A YIELD AND PRIORITY ROAD TRAFFIC SIGN. ASR AND AVERAGE QUERIES $Q$ UNDER VARYING BRIGHTNESS CONDITIONS FOR FIXED $k = 192$ MPS AND FOR VARYING NUMBER $k$ OF MPS FOR A FIXED BRIGHTNESS OF 2000 LUX. RESULTS FOR REAL-WORLD EXPERIMENTS ARE SHOWN IN SECTION V-C.




| | Lux | | | | | | Patches ($k$) | | | | | | Lux | | | | | | Patches ($k$) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | 1000 | 2000 | 3000 | 4000 | 5000 | 16 | 32 | 64 | 96 | 128 | 192 | 10 | 1000 | 2000 | 3000 | 4000 | 5000 | 16 | 32 | 64 | 96 | 128 | 192 |
| ASR | 96.0 | 98.0 | 95.0 | 79.0 | 38.0 | 20.0 | 66.0 | 89.0 | 94.0 | 95.0 | 94.0 | 95.0 | 83.0 | 85.0 | 81.0 | 70.0 | 46.0 | 12.0 | 39.0 | 58.0 | 81.0 | 76.0 | 77.0 | 81.0 |
| $Q$ | 137.2 | 155.9 | 193.9 | 358.4 | 693.5 | 822.2 | 476.8 | 231.4 | 161.3 | 141.3 | 155.6 | 193.9 | 330.8 | 313.1 | 365.1 | 493.6 | 690.6 | 918.2 | 740.1 | 588.0 | 380.7 | 404.7 | 388.0 | 365.1 |

TABLE VI

ASR FOR VARIOUS SURROGATE AND TARGET ARCHITECTURES OF VARYING COMPLEXITY. THE BOLD DIAGONAL ELEMENTS INDICATE THE ASR WHEN SURROGATE AND TARGET ARCHITECTURES ARE IDENTICAL.

| | | Two-Stage | | | | Single-Stage | | |
|---|---|---|---|---|---|---|---|---|
| | **Target** → | CNN | Res-Net50 | Swin-Trans. | Conv-NeXt | **Target** → YOLOv8 | Faster-RCNN |
| **Surrogate** | CNN | **95.16** | 74.88 | 57.77 | 57.58 | YOLOv8 **100.00** | 93.72 |
| | ResNet50 | 71.07 | **97.58** | 55.81 | 56.37 | | |
| | SwinTrans. | 72.56 | 75.63 | **97.12** | 66.98 | Faster-RCNN 93.38 | **100.00** |
| | ConvNeXt | 72.84 | 77.40 | 68.84 | **97.21** | | |

TABLE VII

COMPARISON AGAINST STATE-OF-THE-ART W.R.T. ASR, AVERAGE QUERIES $Q$ ON GTSRB, AND TIME IT TAKES FOR DEPLOYMENT.

| | Shapes & Location [45] | HotNCold [43] | **Ours** |
|---|---|---|---|
| ASR | 74.6 | 82.6 | **95.07** |
| $Q$ | 200.0 | 200.0 | **68.31** |
| Time | $\sim$ 5 min | 30 min | $\sim$ **50s** |

aforementioned ablation study, and apply our infrared transformation with a brightness of 10 lux.

Our results are depicted in Table VII[3]. We find that our proposal results in a remarkably higher ASR by at least 12.5% and a lower amount of queries, by up to 65%, compared to [43], [45], even though [45] is a white-box method with direct access to model gradients.

To compare against [25]—a projector-based attack in the *visible* light spectrum, we generate perturbations at 120 lux for 100 stop signs[4] sampled from GTSRB and obtain an ASR of 100% at an average of just 2.42 queries. Our results are on par with the results in [25]; however, our scheme does not require the generation of individual projection models and saves considerable effort in generating adversarial examples.

### C. Perturbation Attacks in the Physical World

We now proceed to evaluate our approach in the real world. In our experiments, we directly perturb the $w = h = 224$ large images using a square MP with $l = 14$ and apply the aforementioned EOT transformations (cf. Section IV-E) while

[3]We could unfortunately not compare with [35], [44] due to the unavailability of their source code.

[4]Note that a comparison with the full GTSRB was not feasible as the projection model of [25] is sign-specific and is only available for a stop sign.

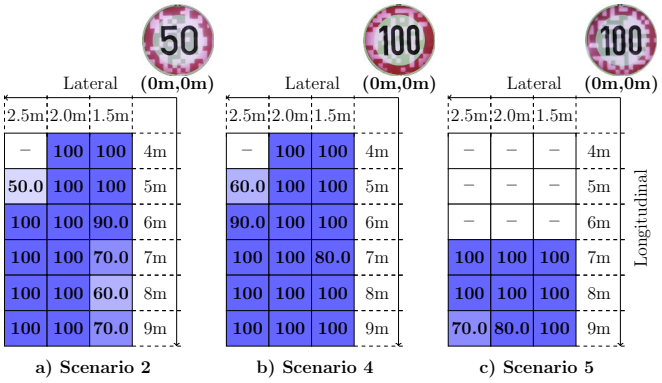

(a) Indoor setting. (b) Outdoor setting.

Fig. 10. Experimental environments with 1000 lux (avg.) on the sign surface.

enforcing a query budget of $Q = 2500$ to make the perturbations more robust. In practice, generating a single perturbation takes approximately four minutes and needs to be performed only once before deployment. Because these attacks transfer effectively across classifiers (cf. Section V-B), the adversary does not need to interact directly with the target classifier in the vehicle. Due to the larger image size, we opted to rely on this (large) MP size to facilitate the recognition by a camera. Here, we select one representative class mapping for each introduced scenario and devise ten dedicated perturbations, i.e., we average the success of each scenario over ten different perturbations. An example of Scenario 1 is the targeted class flip from speed limit 100 to speed limit 30.

**Setup & Hardware:** We performed our experiments using two different cameras with CMOS sensors, which are commonly found in product families used in autonomous driving or commercial traffic sign recognition systems, such as Baidu Apollo. For most experiments, we use (1) Raspberry Pi Camera Module 3 without infrared filters based on a Sony IMX708 sensor with a focal length of 4.74mm (similar to Leopard Imaging `LI-USB30-IMX728-GMSL3-070H`). In another dedicated test, we also relied on (2) Leopard Imaging `LI-USB30-AR023ZWDR` (using an OnSemi `AR023ZWDR` sensor) with a focal length of 6mm, which has also been used in other works [35]. Both cameras have been connected to a Raspberry Pi Model 4. To broaden the range of ambient light intensity conditions, we used a powerful 12W 808nm infrared light source in all our experiments[5]. This allows us to produce clearly visible perturbations even under high ambient light levels of up to 1100 lux. Ambient light intensity is measured directly on the surface of the sign using a lux meter.

[5]In our initial tests, we used a 5W 850nm infrared lamp to successfully mount attacks up to an ambient light intensity of 300 lux.




a) Scenario 2 b) Scenario 4 c) Scenario 5

Fig. 11. Attack success rate for various scenarios at different camera positions in an indoor setting at a brightness of 1000 lux. We omitted the datapoint at (4m, 2.5m) because the sign was not fully within the camera's field of view.

We printed the previously generated perturbations on transparent off-the-shelf overhead projector film made from PET, costing around US$0.1 per perturbation. This process is sufficiently precise for our purpose, as initial experiments on different versions of the same perturbation did not show any impact on the attack success rate.

**Placement:** For placing the camera and infrared light source in our experiments, we assume a real-world setting mimicking a traffic setting (cf. Figure 1), in which we place our sign on the right side of the road. We place the infrared light source at a fixed position opposite the sign at a distance of 2 meters, considering one-shot attackers. This choice is reasonable, as traffic signs are typically located on the side of the street, which is also the only practical place to position a light source (e.g., on bridges, alternative side placements may not be feasible). The projection was manually aligned once using the live camera feed prior to any experiment and remained unchanged throughout experiments. The default position of our camera is located in the middle of the right driving lane at a distance of 4 meters (longitudinal) and 2 meters (lateral) to the left of the sign, at a default viewing angle of $\sim 25°$.

We first evaluate the success of our approach in a controlled and artificially lit indoor environment, i.e., a basement with bright natural video lighting (Figure 10a), and then move into a more diverse outdoor scenario, i.e., a parking lot (Figure 10b). In both settings, we measure an average ambient lighting of 1000 lux. At all times, we verified the correct classification even in the presence of an infrared light spot (without a perturbation).

**General Success:** We evaluate the performance of our approach in the previously introduced scenarios in both indoor and outdoor environments (cf. Table VIII). To this end, we place the camera at the previously described distance and determine the ASR over ten different perturbations. In the indoor setting, we obtain an ASR of 100% for Scenarios 1, 2, 3, and 4. In the outdoor environment, we observe success rates of 90% and 80% for the first two scenarios, respectively, while the last two scenarios maintain a success rate of 100%.


(a) Raspberry Pi Camera 3 (b) Leopard Imaging AR023ZWDR

Fig. 12. Infrared perturbation captured with two different camera sensors.

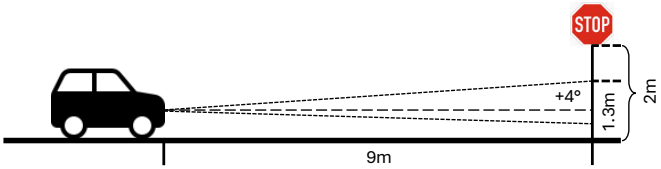



Fig. 13. Impact of headlights (cf. ECE-R112 [3]) at a distance of 9m on a traffic sign mounted at a height of 2m. The maximum permitted brightness on the surface of the sign is $\sim$ 22 lux. Figure is to scale (icon acquired from [1]).

In a separate experiment (conducted outdoors) on a yield sign flipping to a priority road sign, we observe an ASR of 70%.

**Different Angles, Distances, Cameras:** To assess the impact of real-world environments, such as spatial transformations introduced by angle and distance on the robustness of the perturbed signs and the success of our EOT transformations (cf. Section IV-E), we conduct the following experiments: we place the camera at the default distance of 4 meters of longitudinal and 2 meters of lateral distance away from the sign and verify the success for various camera positions. Namely, we simulate different lane positions of the vehicle on the road by moving the camera laterally to the left and right by 0.5 meters. We combine this with longitudinal distances between 4 and 9 meters in one-meter increments to verify the robustness of our approach, which results in diverse viewing angles between 27° and 10°. These positions faithfully capture various real-world positions across different driving lanes and are also limited by the visibility of the sign on the camera. As shown in Figure 11, we observe consistently high ASR , averaging over 90% in almost all configurations. However, we observe a reduction in ASR for (5m, 2.5m) due to the slightly steeper viewing angle and increased distance (resulting in a less visible reflection). Another decrease in ASR is measured when increasing the longitudinal distance across a lateral distance of 1.5m due to the reflections of the infrared light source becoming more prominent on the sign (especially for the targeted Scenario 2).

To evaluate the transferability of our approach across different camera sensors, we also instrumented an additional camera, namely Leopard Imaging AR023ZWDR (cf. Figure 12), in the indoor setting. In this setting, we tested ten different perturbations for Scenario 1 and observed a high transferability rate of 90%. As the spectral sensitivity curves of CMOS camera sensors in the near-infrared part of the spectrum (800-1000nm) are highly similar, we expect our approach to also be effective against other sensors.

TABLE VIII
ASR IN THE PHYSICAL WORLD IN THE INDOOR AND OUTDOOR ENVIRONMENT FOR A GIVEN SCENARIO.

| | Scenario | | | | |
|---|---|---|---|---|---|
| Environment | #1 | #2 | #3 | #4 | #5 |
| Indoor | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Outdoor | 90.0 | 80.0 | 100.0 | 100.0 | 100.0 |
| Moving (10 km/h) | 99.4 | 93.7 | 96.3 | 84.8 | 79.8 |
| Moving (30 km/h) | 98.0 | 90.0 | 84.5 | 84.4 | 85.7 |

TABLE IX
IMPACT OF CURRENT DEFENSES ON THE CA AND ASR ON GTSRB (DIGITAL DOMAIN) AND ON OUR EXPERIMENTAL DATA (PHYSICAL DOMAIN). ↑ (RESP. ↓) INDICATES THAT VALUES CLOSE TO 100 (RESP. 0) PROVIDE BETTER RESULTS.

| | Digital | | | | Physical |
|---|---|---|---|---|---|
| | No defense | Spatial Smooth. (non-local) [47] | Spatial Smooth. (local) [47] | Adv. Training [13] | **Ours** Segment. -based |
| CA ↑ | 98.76 | 95.35 | 96.56 | 98.67 | **96.63** |
| ASR ↓ | 95.16 | 67.72 | 61.77 | 62.89 | **25.3** |

**Driving vehicle:** We now conduct moving vehicle experiments by driving past the perturbed sign in an outdoor environment. In our setup, the camera is mounted on the rear-view mirror, which corresponds to the typical height for front-facing camera systems in modern vehicles. Starting from a distance of 30 meters, we approach the sign at two different speeds: 10 km/h and 30 km/h. For safety reasons, we were unable to experiment at higher speeds. For each perturbation and across all five scenarios, we record a video and compute the ASR over all cropped frames (cf. Table VIII). Our results demonstrate the practical effectiveness of our approach, achieving an ASR of up to $98\%$ and as low as $79.8\%$ at speeds of up to 30 km/h. Some fluctuations are observed, which we attribute to slight variations in the driving path and the fact that the experiments were conducted over several hours.

**Impact of headlights:** Headlights, particularly at dusk or night, create high-brightness conditions. To evaluate the impact of headlights on the robustness of our approach, we conducted outdoor tests under headlight illumination. We found no significant impact on our results within the tested range of 4–9m. As shown in Figure 13, this resilience is mainly due to regulatory constraints designed to minimize glare for other drivers. Specifically, ECE-R112 [3] [Figure B and Section 6.2.4] stipulates that headlight illumination at a height of 2 meters—where traffic signs are typically placed—must not exceed $\sim 22$ lux at a distance of 9 meters.

## VI. EXPERIMENTS ON SINGLE-STAGE ARCHITECTURES

We now shift our focus to single-stage architectures, which are typically used in object detection. Here, we conducted our experiments on the established Mapillary [9] and GTSDB [39] datasets. Mapillary consists of 401 classes with traffic signs from all continents, while GTSDB contains 43 different German road sign classes, similar to the previously used GTSRB. We train a YOLOv8 model [11] on the Mapillary dataset with a reduced number of classes, i.e., European speed limit signs and stop signs, to achieve better performance (cf. Section II), and obtain an mAP-50 of $64.9\%$. Additionally, we train a Faster-RCNN [34] model on GTSDB and obtain an mAP-50 of $90.76\%$.

**Hiding Attacks (Scenario 5) in the Digital Domain:** In the setting of a single-stage architecture, the goal of the adversary is to ensure that the sign is no longer detected by the system. In other words, speed limits and other important signs, e.g., stop signs, are ignored by the traffic sign recognition system.

We use 25 images per class for Mapillary, i.e., a total of 225 images, and the entire test set for the previously selected classes of GTSDB, while ensuring that we only select bounding boxes with more than $32 \times 32$ pixels. As shown in Table III, we obtain high ASRs at $k = 192$ of $100\%$ for an average of $3.23/4.97$ queries for Mapillary and GTSDB, respectively. Even for the single-stage architectures, we measure high success rates and a lower amount of used queries at a higher value of $k$ and a lower ambient light level.

To assess whether a perturbation generated for one architecture is also successful on another, we use the generated images for Mapillary on YOLOv8 and evaluate the success of a hiding attack on the Faster-RCNN model trained on GTSDB (and vice versa). As shown in Table VI, we measure a higher transferability of $\sim 93\%$ compared to two-stage architectures.

**Perturbation Attacks in the Physical World:** Analogously to the two-stage experiments, we place the camera at a distance of 7 meters and generate ten perturbations in Scenario 5. As shown in Table VIII, we measure a success rate of $100\%$. Note that a larger initial distance is necessary for initial detection, as the dataset consists of more images with smaller signs at a distance rather than close-up signs.

In Figure 11, we further vary the distance and angle between the camera and the traffic sign (starting from the initial distance of 7 meters). Our results consistently show an average success of $\sim 95\%$.

## VII. DEFENSES AGAINST INFRARED PERTURBATIONS

Since infrared spectral filters impair camera performance in low-light conditions (cf. Appendix A), we now explore the solution space to defend against infrared perturbations and then present our defense, dubbed *segmentation-based detection*.

### A. Limitations of Current Defenses

**Spatial Smoothing & Adversarial Training:** First, we evaluate the impact of two popular defenses on our approach: the test-time spatial smoothing defense [47] and the popular (but costly) adversarial training [13].

Local smoothing applies a median blur by replacing each pixel with the median of its neighbors, while non-local smoothing

uses a larger region. Both aim to undo adversarial perturbations, following prior work [44], [45]. Adversarial training strengthens test-time robustness by incorporating adversarial examples into the training process.

Our results for the strongest attacker, i.e., an infrared transformation for 10 lux, are included in Table IX. We observe that all three defenses fail to fully mitigate our attack: test-time defenses [47] reduce ASR to 67.72% (non-local) and 61.77% (local), while adversarial training lowers it to only 62.89%.

**Certified patch detection:** PatchCleanser [46] is a certified defense that selectively masks portions of an image—if the mask covers an adversarial patch, the prediction of the classifier changes. In contrast, benign natural images are generally invariant to this mask. This does not apply to our use case of traffic-sign recognition as masking, e.g., a speed sign, creates an ambiguity of the underlying speed limit [35]. An additional requirement of PatchCleanser is that the mask must be larger than the used adversarial patch—we, however, perturb the entire sign with our perturbation.

**Infrared speckle detection:** [35] uses the characteristic speckle pattern of laser reflections for detection. While we also utilize infrared light, our approach uses an incoherent light source, i.e., *not* a laser, and hence our perturbations do not exhibit a strong speckle pattern as required by [35][6].

**Spatio-temporal consistency:** When conducting evasion attacks in the real world, it has been shown that evading individual camera frames is not sufficient to successfully attack a system [38]. Indeed, by monitoring the spatio-temporal properties of objects, one can detect changes in bounding box size and classification over time [14], [27], [48][6]. These approaches typically rely on inconsistencies resulting from adversarial perturbations and can only be defeated when the model predictions are consistent "enough" over time, while also considering a model's natural error rate. In our moving vehicle experiments (cf. Table VIII), we obtain ASRs of up to 99.4%. Specifically, our targeted attacks are successful over most captured frames and therefore cannot be detected using such approaches. These results show a consistent targeted misclassification *over the 163 frames of the video, with only one flickering frame scattered in between* (i.e., with an error rate of 0.6%), which we attribute to the model's natural error rate due to motion blur. Importantly, as the majority of frames while approaching a sign are consistent, we believe that defenses based on spatio-temporal consistency will have a limited effect here.

Some approaches like [51] utilize object texture, behavior, and interactions with one another and focus specifically on detecting pedestrians and cars. This approach is not effective for traffic sign recognition as traffic signs have a similar texture, remain on fixed trajectories, and generally do not interact with other objects (like cars and pedestrians).

[6]Notice that a comparison to [14], [35], [48] is not possible since the source code has not been made available to us or cannot be extended to new attacks.

(a) Perturbed sign. (b) Perturbed sign (segmented). (c) Segmented sign.

Fig. 14. Output of our segmentation patched defense for a stop traffic sign.

TABLE X
EVALUATION OF OUR SEGMENTATION-BASED DETECTION SCHEME IN THE PHYSICAL DOMAIN. ↑ (RESP. ↓) INDICATES THAT VALUES CLOSE TO 100 (RESP. 0) PROVIDE BETTER RESULTS.

| | Static | | | | | Moving |
|---|---|---|---|---|---|---|
| | Indoor | Outdoor | Distance (Longitudinal) | Distance (Lateral) | Σ | Σ |
| CA ↑ | 100 | 100 | 100 | 100 | 100 | 96.63 |
| ASR ↓ | 2.49 | 2.56 | 1.63 | 1.65 | 2.05 | 25.3 |

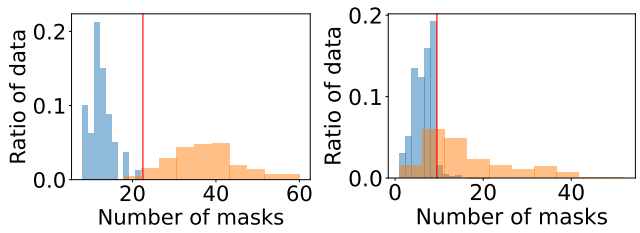
### B. Our Proposal—Segmentation-based Detection

We now propose a novel detection scheme specifically designed to thwart our attack. Our defense builds on the observation that our perturbations introduce a significant amount of additional shapes and edges into the image—considerably beyond the number of edges/shapes that are typically present in common traffic signs (cf. Figure 14). More specifically, our defense measures the number of detected shapes in a given input image and compares it to an empirically derived threshold $\nu$, above which the image is considered adversarial.

To ensure robust and brightness-agnostic detection of shapes within the image, we utilize the Segment Anything [19] segmentation model with the ViT-L architecture to compute segmentation masks of an input image. This model $\mathcal{F}$ outputs a mask $m$ with $t$ pixels, i.e., $m = \{(x_0, y_0), \ldots, (x_t, y_t)\}$, for each of the $u$ detected shapes within an image, constituting the set $\mathcal{R} = \{m_0, \ldots, m_u\}$, i.e., $\mathcal{F}(x) = \mathcal{R}$.

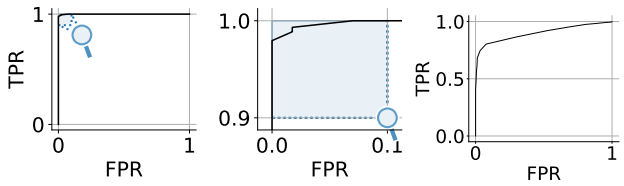
**Dataset.** To evaluate our defense, we relied on a dataset consisting of 54 benign/unperturbed and 400 adversarial/perturbed images of traffic signs taken in static scenarios, as well as 476 and 1158 images taken from a moving vehicle, respectively (both obtained from our real-world experiments in Section V-C).

**Determining $\nu$:** We interpolated the threshold $\nu$ on the number of detected shapes, i.e., $|\mathcal{R}|$, experimentally based on the benign and perturbed traffic signs captured in our static experiments (diverse lighting conditions, distances), and from a moving vehicle. The distribution of segmentation masks is presented in Figure 15, supporting our initial hypothesis that benign images contain fewer detected shapes than adversarial ones, with only limited overlap between the two distributions. To evaluate detection performance across all possible threshold values $\nu$, we employ a receiver operating characteristic (ROC)




(a) Static setting. (b) Moving vehicle.

Fig. 15. Distribution of the segmentation masks for the benign (blue) and adversarial (orange) data for images taken in a static and a moving setting. The red line indicates the EER-optimal $\nu$.




(a) Static setting. (b) Moving vehicle.

Fig. 16. ROC curves for images taken in static and moving settings.

curve, as shown in Figure 16. Performance across different data partitions is summarized in Table X.

**Performance in static scenarios:** For the optimal threshold $\nu = 22$, we obtain a CA $= 100\%$ and an ASR as low as $2.05\%$ at an equal-error rate (EER) of $\sim 2\%$ and an F1-score of $99\%$. Within the static scenarios, we observe no differences in performance between indoor and outdoor settings, nor any variations related to specific distances in latitude or longitude.

**Performance under a moving vehicle:** To evaluate real-world performance in a dynamic setting, we utilize the aforementioned dataset—collected from a moving vehicle. While performance shows a slight decline relative to the static case, the impact on CA remains minimal, and our approach continues to outperform all other defenses by a substantial margin (see Table IX). Specifically, for a threshold of $\nu = 9$, we achieve an ASR of $25.3\%$ at an F1-score of $85\%$. This slight performance degradation is likely due to the inclusion of images captured from the moving vehicle at distances of up to 30 meters, resulting in smaller object sizes that make segmentation more challenging.

## VIII. Conclusion

In this paper, we present a novel and cost-effective attack to generate robust perturbations in the near-infrared domain, which we dub adversarial infrared perturbations. Our approach ensures real-world robustness by accounting for the spectral shift into the infrared domain and is the first practical attack that works in both targeted and untargeted attack scenarios. Extensive experiments in the digital and physical domains show that our approach yields consistently high attack success rates in various situations while requiring up to $65\%$ fewer queries when compared to existing approaches. We showed that existing defenses against perturbations cannot successfully defend against our approach. As a remedy, we proposed a novel segmentation-based detection scheme that is specifically designed to thwart our attack with an F1-score of up to $99\%$.

## Acknowledgment

This work has been co-funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972, by the German Federal Ministry of Research, Technology and Space (BMFTR) through the project TRAIN (01IS23027A).

## Ethics Considerations

Our paper proposes a novel, stealthy, and cost-effective attack to generate both targeted and untargeted robust perturbations. Our measurements have been conducted in a carefully controlled environment to ensure the accuracy and reliability of the data collected. Throughout our measurement process, we took meticulous precautions to guarantee that no harm or disruption occurred to any vehicles or pedestrians on the road. We also ensured at all times full compliance with all applicable safety regulations and standards.

Our main goal in this paper is to raise awareness about this type of attack and to promote the adoption of suitable defenses for autonomous driving. We explored a range of defensive strategies and demonstrated the feasibility of our proposed countermeasure in mitigating this attack. In this paper, we have only focused on open-source datasets and models and *not* considered any vendor-specific machine learning systems.

We responsibly disclosed our findings—along with our proposed segmentation-based detection scheme—to automotive and camera manufacturers that operate under a system model similar to the one described in Section III-A, namely Mercedes, Mobileye, Tesla, Sony, and OnSemi. We track vendor responses and ongoing interactions at: https://rub-infsec.github.io/infrared_perturbations.

## References

[1] Icons by Icons8. https://icons8.de/.

[2] Tesla sales. https://cleantechnica.com/tesla-sales/.

[3] Regulation No 112 of the Economic Commission for Europe of the United Nations (UN/ECE) — Uniform provisions concerning the approval of motor vehicle headlamps emitting an asymmetrical passing-beam or a driving-beam or both and equipped with filament lamps and/or light-emitting diode (LED) modules, August 2014.

[4] Colorimetry – part 4: CIE 1976 L*a*b* colour space - ISO/CIE 11664-4:2019(E). 2019.

[5] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293. PMLR, 2018-07-10/2018-07-15.

[6] Robert L. Bingle, Joseph Camilleri, Peter J. Whitehead, and Kenneth Schofield. Imaging system for vehicle, June 2011.

[7] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society, 2017.

[8] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A. K. Qin, and Yun Yang. Adversarial Camouflage: Hiding Physical-World Attacks With Natural Styles. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 997–1005, Seattle, WA, USA, June 2020. IEEE.

[9] Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, Gerhard Neuhold, and Yubin Kuang. The mapillary traffic sign dataset for detection and classification on a global scale. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 68–84, Cham, 2020. Springer International Publishing.

[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust Physical-World Attacks on Deep Learning Visual Classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, Salt Lake City, UT, USA, June 2018. IEEE.

[11] Glenn Jocher and Ayush Chaurasia and Jing Qiu. Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics, 2023.

[12] Ian J. Goodfellow, Nicolas Papernot, and Patrick D. McDaniel. Cleverhans v0.1: An adversarial machine learning library. *CoRR*, abs/1610.00768, 2016.

[13] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[14] Xingshuo Han, Haozhao Wang, Kangqiao Zhao, Gelei Deng, Yuan Xu, Hangcheng Liu, Han Qiu, and Tianwei Zhang. VisionGuard: Secure and Robust Visual Perception of Autonomous Vehicles in Practice. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1864–1878, Salt Lake City UT USA, December 2024. ACM.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[16] Chengyin Hu, Yilong Wang, Kalibinuer Tiliwalidi, and Wen Li. Adversarial Laser Spot: Robust and Covert Physical-World Attack to DNNs. In *Proceedings of The 14th Asian Conference on Machine Learning*, pages 483–498. PMLR, April 2023.

[17] 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 *2021 IEEE Symposium on Security and Privacy (SP)*, pages 160–175, San Francisco, CA, USA, May 2021. IEEE.

[18] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, Paris, France, October 2023. IEEE.

[20] Sebastian Köhler, Richard Baker, and Ivan Martinovic. Signal Injection Attacks against CCD Image Sensors. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 294–308, Nagasaki Japan, May 2022. ACM.

[21] You Li, Julien Moreau, and Javier Ibañez-Guzmán. Emergent visual sensors for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24:4716–4737, 2023.

[22] Yufeng Li, Fengyu Yang, Qi Liu, Jiangtao Li, and Chenhong Cao. Light can be dangerous: Stealthy and effective physical-world adversarial attack by spot light. *Computers & Security*, 132:103345, 2023.

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[24] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A Convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[25] Giulio Lovisotto, Henry Turner, Ivo Sluganovic, Martin Strohmeier, and Ivan Martinovic. SLAP: Improving physical adversarial examples with Short-Lived adversarial perturbations. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1865–1882. USENIX Association, August 2021.

[26] Yao Ma, Leting Shan, Yiran Ying, Liang Shen, Yufeng Fu, Linfeng Fei, Yusheng Lei, Nailin Yue, Wei Zhang, Hong Zhang, Haitao Huang, Kai Yao, and Junhao Chu. Day-Night imaging without Infrared Cutfilter removal based on metal-gradient perovskite single crystal photodetector. *Nature Communications*, 15(1):7516, August 2024.

[27] Yanmao Man, Raymond Muller, Ming Li, Z. Berkay Celik, and Ryan Gerdes. That person moves like a car: Misclassification attack detection for autonomous systems using spatiotemporal consistency. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 6929–6946, Anaheim, CA, August 2023. USENIX Association.

[28] Shuijiang Mao and Xianqing Guo. Image method of image sensor, imaging apparatus and electronic device. https://www.itu.int/rec/T-REC-H.273/en, December 2018.

[29] Andreas Møgelmose, Mohan M. Trivedi, and Thomas B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13:1484–1497, 2012.

[30] Ben Nassi, Yisroel Mirsky, Dudi Nassi, Raz Ben-Netanel, Oleg Drokin, and Yuval Elovici. 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*, pages 293–308, Virtual Event USA, October 2020. ACM.

[31] Dinh-Luan Nguyen, Sunpreet S. Arora, Yuhang Wu, and Hao Yang. Adversarial Light Projection Attacks on Face Recognition Systems: A Feasibility Study. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3548–3556, Seattle, WA, USA, June 2020. IEEE.

[32] Buu Phan, Fahim Mannan, and Felix Heide. Adversarial Imaging Pipelines. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16046–16056, Nashville, TN, USA, June 2021. IEEE.

[33] LA Rastrigin. The convergence of the random search method in the extremal control of a many parameter system. *Automaton & Remote Control*, 24:1337–1342, 1963.

[34] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[35] Takami Sato, Sri Hrushikesh Varma Bhupathiraju, Michael Clifford, Takeshi Sugawara, Qi Alfred Chen, and Sara Rampazzi. Invisible Reflections: Leveraging Infrared Laser Reflections to Target Traffic Sign Perception. In *Proceedings 2024 Network and Distributed System Security Symposium*, San Diego, CA, USA, 2024. Internet Society.

[36] Athena Sayles, Ashish Hooda, Mohit Gupta, Rahul Chatterjee, and Earlence Fernandes. Invisible Perturbations: Physical Adversarial Examples Exploiting the Rolling Shutter Effect. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14661–14670, Nashville, TN, USA, June 2021. IEEE.

[37] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540, Vienna Austria, October 2016. ACM.

[38] Junjie Shen, Ningfei Wang, Ziwen Wan, Yunpeng Luo, Takami Sato, Zhisheng Hu, Xinyang Zhang, Shengjian Guo, Zhenyu Zhong, Kang Li, Ziming Zhao, Chunming Qiao, and Qi Alfred Chen. SoK: On the semantic AI security in autonomous driving. *CoRR*, abs/2203.05314, 2022.

[39] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012.

[40] Jia Tan, Nan Ji, Haidong Xie, and Xueshuang Xiang. Legitimate Adversarial Patches: Evading Human Eyes and Detection Models in the Physical World. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5307–5315, Virtual Event China, October 2021. ACM.

[41] Simen Thys, Wiebe Van Ranst, and Toon Goedeme. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 49–55, Long Beach, CA, USA, June 2019. IEEE.

[42] Wei Wang, Yao Yao, Xin Liu, Xiang Li, Pei Hao, and Ting Zhu. I Can See the Light: Attacks on Autonomous Vehicles Using Invisible Lights.

In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1930–1944, Virtual Event Republic of Korea, November 2021. ACM.

[43] Hui Wei, Zhixiang Wang, Xuemei Jia, Yinqiang Zheng, Hao Tang, Shin'ichi Satoh, and Zheng Wang. HOTCOLD block: Fooling thermal infrared detectors with a novel wearable design. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 15233–15241. AAAI Press, 2023.

[44] Xingxing Wei, Yao Huang, Yitong Sun, and Jie Yu. Unified adversarial patch for cross-modal attacks in the physical world. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4422–4431. IEEE, 2023.

[45] Xingxing Wei, Jie Yu, and Yao Huang. Physically Adversarial Infrared Patches with Learnable Shapes and Locations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12334–12342, Vancouver, BC, Canada, June 2023. IEEE.

[46] Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. PatchCleanser: Certifiably robust defense against adversarial patches for any image classifier. In Kevin R. B. Butler and Kurt Thomas, editors, *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 2065–2082. USENIX Association, 2022.

[47] Weilin Xu, David Evans, and Yanjun Qi. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *Proceedings 2018 Network and Distributed System Security Symposium*, 2018.

[48] Yuan Xu, Gelei Deng, Xingshuo Han, Guanlin Li, Han Qiu, and Tianwei Zhang. PhyScout: Detecting Sensor Spoofing Attacks via Spatio-temporal Consistency. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1879–1893, Salt Lake City UT USA, December 2024. ACM.

[49] Vivek Yadav. GTSRB-CNN. https://github.com/vxy10/p2-TrafficSigns.

[50] Chen Yan, Zhijian Xu, Zhanyuan Yin, Xiaoyu Ji, and Wenyuan Xu. Rolling colors: Adversarial laser exploits against traffic light recognition. In Kevin R. B. Butler and Kurt Thomas, editors, *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 1957–1974. USENIX Association, 2022.

[51] Zhiyuan Yu, Ao Li, Ruoyao Wen, Yijia Chen, and Ning Zhang. PhySense: Defending Physically Realizable Attacks for Autonomous Systems via Consistency Reasoning. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 3853–3867, Salt Lake City UT USA, December 2024. ACM.

[52] Shibo Zhang, Yushi Cheng, Wenjun Zhu, Xiaoyu Ji, and Wenyuan Xu. CAPatch: Physical adversarial patch against image captioning systems. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 679–696, Anaheim, CA, August 2023. USENIX Association.

[53] Yiqi Zhong, Xianming Liu, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Shadows can be Dangerous: Stealthy and Effective Physical-world Adversarial Attack by Natural Phenomenon. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15324–15333, New Orleans, LA, USA, June 2022. IEEE.

[54] Husheng Zhou, Wei Li, Zelun Kong, Junfeng Guo, Yuqun Zhang, Bei Yu, Lingming Zhang, and Cong Liu. DeepBillboard: Systematic physical-world testing of autonomous driving systems. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 347–358, Seoul South Korea, June 2020. ACM.

[55] Zhe Zhou, Di Tang, Xiaofeng Wang, Weili Han, Xiangyu Liu, and Kehuan Zhang. Invisible mask: Practical attacks on face recognition with infrared. *CoRR*, abs/1803.04683, 2018.

[56] Xiaopei Zhu, Zhanhao Hu, Siyuan Huang, Jianmin Li, and Xiaolin Hu. Infrared Invisible Clothing: Hiding from Infrared Detectors at Multiple Angles in Real World. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13307–13316, New Orleans, LA, USA, June 2022. IEEE.

[57] Xiaopei Zhu, Xiao Li, Jianmin Li, Zheyao Wang, and Xiaolin Hu. Fooling thermal infrared pedestrian detectors in real world using small bulbs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 3616–3624. AAAI Press, 2021.

(a) Ambient light. (b) Infrared light. (c) Ambient/infrared light.

Fig. 17. Examples of different infrared absorbing films on a speed limit 20 sign with various light sources.

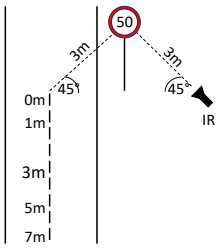



Fig. 18. Experimental setup used for capturing the `GTSRB-IR-100` dataset.

## Appendix A
## Infrared Filters and Films

Another possible defense is to embed an infrared spectral filter before the CMOS sensor to protect against attacks that utilize infrared light. While this completely masks the visibility of our infrared perturbations on inputs to the classifier, such an approach is not feasible for several reasons. First, adding spectral filters would significantly impair the ability of cameras to operate in adverse environmental conditions, such as at night or in low-light conditions. In fact, infrared camera vision is instrumental in detecting lane markings and in providing rich contextual information [21]. Second, with over 600,000 electric vehicles sold by Tesla alone in the US [2] in 2023, each containing numerous camera sensors, fitting them all with infrared filters would incur non-negligible extra costs [42]. Another approach would be to integrate infrared films within traffic signs. In a separate experiment (cf. Figure 17), we observed that this solution hampers the recognition of the traffic sign in the presence of natural ambient light.

## Appendix B
## IR Dataset: GTSRB-IR-100

We publish the dataset `GTSRB-IR-100`, which comprises 100 images of traffic signs under varying lighting conditions, with half of the images additionally illuminated by an infrared light source. Each image in our dataset is annotated with a lux value measured on the surface of the street sign. To our knowledge, this is the first and only publicly available dataset featuring infrared light sources.

More concretely, our dataset has been captured according to Figure 18 at distances of $\{0, 1, 3, 5, 7\}$ meters. For this, we took an image with ambient light and with an additional infrared light source in the following environments:

- Controlled outdoor environment (traffic signs are placed on a stand): Yield, Stop, Speed 20, Speed 50, Speed 100
- Realistic-outdoor environment (taking existing traffic signs): Yield, Stop, Turn right, No entry, Speed 30

## APPENDIX C
## ARTIFACT APPENDIX

### A. Description & Requirements

*1) How to access:* The artifact's main components can be accessed in the following repository[7].

*2) Hardware dependencies:* CUDA/MPS acceleration is recommended for speed, but not required. 100GB of disk space is required.

*3) Software dependencies:* Linux/macOS as operating system, Python, and Conda[8] for creating a virtual environment for the execution of the artifact.

*4) Benchmarks:* This artifact uses the GTSRB/GTSDB, LISA, and Mapillary datasets. Model architectures are simple CNNs for GTSRB[9] and LISA[10] (classification), and Faster-RCNN and YOLOv8 (detection).

### B. Artifact Installation & Configuration

Please create a fresh conda environment with: `conda create --name artifact python=3.9.19`. Then activate it with `conda activate artifact`, navigate to the root of the project (in which the `README.md` is located), and install all dependencies with `pip install -r requirements.txt`.

Ensure that the environment is activated for all experiments and that each file is executed from the project's root directory. Prepend `PYTHONPATH=$(pwd)` before the provided command lines in case module import errors are encountered.

Most models and datasets come bundled with the project. An exception is the large Mapillary dataset (`Mapillary.zip`), which is approximately 45GB in size and can also be found in the repository (cf. Section C-A1). Its unzipped folders `images` and `labels` need to be placed in `dataset/detection/Mapillary`. For the evaluation of our segmentation-based defense the `Segment-Anything` model in the ViT-L architecture needs to be downloaded [11] and placed in `model/segmentation`.

Device configuration (CUDA/MPS/CPU) is handled automatically by the `get_device()` function in `utils/utils.py`.

### C. Experiment Workflow

All experiments are in `experiments` and are numbered in line with Section C-E. They consist of a wrapper that schedules invocations of the underlying framework with different parametrizations, which are the basis for most of the reported results in the paper. Any log output of these runs is logged into the `logs` folder once a process terminates. The results and plots for each experiment are written into its respective folder to keep everything organized. More details on the parameters of the underlying framework can be found in `README.md`.

Each of the experiments consists of one (`eX.py`) or two files (`eX_{classification, detection}.py`) to test classification/two-stage and detection/single-stage pipelines separately. The `X` resembles the number of the experiment defined in Section C-D and Section C-E. The execution for each experiment is structured in the same way. An experiment can be run with `python experiments/eX/eX.py --mode run`, evaluated/plotted with `python experiments/eX/eX.py --mode evaluate`. As the experiments can consume quite some time, the argument `--subset` can be used in combination with `--mode run` to only execute a representative subset of experiments, i.e., one parameter or a reduced number of image samples for each dimension, which does *not* change program behavior to facilitate functionality checks. Especially for a reduced number of samples, the results do not align with those in the paper, as they are based on averages across all samples. The evaluation/plotting always considers only the data that is found (subset/full data/data of a running experiment). Experiments can also be terminated manually at any time to evaluate the data generated by then.

### D. Major Claims

- (C1): To simulate how a traffic sign illuminated by infrared light is captured by a camera at different levels of ambient brightness, we derive transformation coefficients with a series of real-world images. This is proven by experiment (E1) and shown in Figure 3 of the paper.
- (C2): The optimal black-box optimization strategy given our perturbation constraints is local random search (LRS). This is proven by experiment (E2) with results shown in Fig. 6 of the paper for a classification pipeline and in Fig. 7 for a detection pipeline.
- (C3): Based on grid-searches, we can identify the optimal parameters of our proposal in the digital domain as $k = 192$ and $l = 2$. This is proven in experiments (E3) (for a classification and detection pipeline) and (E4), whose results are illustrated in Table III and Figure 9 of the paper, respectively. We also illustrate its efficacy for various types of signs in experiment (E5) in Table V.
- (C4): Perturbations generated on one machine learning model transfer to another. This is proven in experiment (E6) for a classification and detection pipeline and illustrated in Table VI of the paper.
- (C5): Our proposed segmentation-based defense outperforms existing defense schemes against our attack. This is proven in experiments (E7) and (E8) and illustrated in Table IX/X, Figure 15, and Figure 16.
- (C6): We also provide the GTSRB-IR-100 dataset of traffic signs with(out) infrared illumination, which is located in `dataset/gtsrb-ir-100`. Details can be found in Appendix B.

### E. Evaluation

No additional preparation/configuration is needed beyond Section C-B and hence the [How to]/[Preparation] are

[7] https://github.com/RUB-InfSec/infrared_perturbations

[8] https://www.anaconda.com/docs/getting-started/miniconda/install#quickstart-install-instructions

[9] https://github.com/vxy10/p2-TrafficSigns

[10] https://github.com/cleverhans-lab/cleverhans

[11] https://dl.fbaipublicfiles.com/segment_anything/sam_vit_l_0b3195.pth

omitted. [Execution] is in line with Section C-C. The runtime estimates are based on CPU (and subject to further refinement) and benefit from acceleration with MPS/CUDA. For classification/detection experiments, the subset consists of a few samples from the dataset (besides a reduced parameter space).

*1) Experiment (E1):* [Infrared transformation] [1 compute-minute]: based on a set of included images taken at different levels of ambient brightness with and without an infrared light source, color-channel (RGB) specific parameters are computed to fit the brightness-dependent curves.

*[Results]* The saved plot is Figure 3 and shows the values of the images (points) and our fit curves (lines).

*2) Experiment (E2):* [Optimizer grid-search] [8 hours (subset) / 96 hours (full)]: To determine the black-box optimization strategy that optimizes our proposal the best, we conduct a grid search over various optimizers and perturbation sizes. As a representative subset, we propose computing the quickest parameter, i.e., $k = 768$, across all optimizers.

*[Results]* The individual images are saved into `results`, out of which Figures 6 and 7 are generated for classification and detection, respectively. These results illustrate that LRS is the optimization strategy that offers the best performance both in terms of the highest attack success rate and the lowest number of consumed queries.

*3) Experiment (E3):* [Parameter grid-search] [6 hours (subset) / 72 hours (full)]: To understand the performance of our proposal across different scenarios, i.e., combinations of source and target signs, and datasets, we perform a grid search over the two most important parameters that steer the strength of our attack. These parameters are brightness in lux and the number of patches $k$. As a result, we fix lux=10 and ablate $k$, taking the best value of $k = 192$ and then ablate across all levels of lux. As a representative subset, we propose to compute the quickest parameter, i.e., just $k = 192, \mathrm{lux} = 10$ across all scenarios.

*[Results]* The results are saved as heatmaps and have been manually transferred to Table III. The scenarios `"brake", "accelerate", "stop", "untargeted"` of `e3_classification` map to Scenario 1-4, while the result of `e3_detection` maps to Scenario 5. For each scenario, we obtain results for two datasets (columns within a given Scenario). The values for the rows *lux and patches (k)* can be read from the heatmaps. The results show that our proposal has ideal parameter combinations and performs well across an extensive range of lux and patch counts.

*4) Experiment (E4):* [Ablation of patch-width] [3 hours (subset) / 36 hours (full)]: To understand the performance of the parameter $l$, i.e., patch width, on our proposal and to identify the optimal parameter, we perform a grid search for both the strongest attacker (lux=10) across the targeted Scenario 1 on the GTSRB dataset, varying the number of perturbed pixels. As a representative subset, we propose computing the quickest parameter, i.e., $k = 768$, across all scenarios.

*[Results]* The results are saved as heatmaps, which can be found in Figure 9. These results demonstrate that our proposal achieves the best performance for $l = 2$, in terms of both the highest attack success rate and the lowest number of consumed queries.

*5) Experiment (E5):* [Ablation of specific sign pair] [1 hour (subset) / 8 hours (full)]: To understand the impact of dissimilar source and target signs, we test the performance of our proposal for the considered range of lux and number of patches $k$ on various source/target signs.

*[Results]* The results are saved as heatmaps, and the values of the last row and third column are found in Table V. They demonstrate that we can still achieve a high attack success rate even with other source/target signs.

*6) Experiment (E6):* [Ablation of transferability][4 hour (subset) / 48 hours (full)]: To understand how perturbations generated against one model are still effective against other model architectures, we evaluate cross-model transferability. Specifically, we create perturbations against GtsrbCNN, ResNet-50, ConvNeXt, and SwinTransformer for classification, and Faster-RCNN and YOLOv8 for detection, using these models as source models, and evaluate them against the same set as target models.

*[Results]* The results printed on the console contain the individual high transferability values (cf. Table VI).

*7) Experiment (E7):* [Ablation of other defenses] [3 hours]: To understand how our generated digital perturbations can be mitigated by existing defense schemes, we evaluate them against an adversarially trained model and against input smoothing, i.e., feature squeezing.

*[Preparation]* (E3) has to be run beforehand.

*[Results]* The results are printed on the console and contain the clean accuracy (CA) of the (defended) classifier on benign data and the attack success rate (ASR), which is the performance on the adversarial data from the previous experiment. This resembles the first four columns of Table IX.

*8) Experiment (E8):* [Ablation of our defense] [4 hour (subset) / 48 hours (full)]: To understand how our generated perturbations can be mitigated by our segmentation-based approach, we take a self-captured dataset (included) of real-world images and evaluate the performance of our detection scheme against it. The subset considers a few images to which the defense is applied.

*[Results]* The results are a histogram found in Figure 15 and a ROC curve found in Figure 16. All results are presented in Table X, where the values for the moving experiment are listed in the rightmost column, as shown in Table IX. This illustrates that our defense outperforms the other defenses.

### *F. Customization*

At the top of the experiment files that run grid searches, the parameters and ranges are defined (dictionary `params`). In the `__init__` function, the number of processes is adjusted by `tasks_per_worker`. If you encounter out-of-memory errors, consider reducing this number. The function `get_tasks` defines the subset of experiments. Typically, this involves a subset of parameters or a reduced number of samples to speed up the execution of the environment.