ATTA: Adversarial Task-transferable Attacks on Autonomous Driving Systems

Qingjie Zhang*, Maosen Zhang†, Han Qiu*‡, Tianwei Zhang§, Mounira Msahli‖, and Gerard Memmi‖
*Tsinghua University, China †Beijing University of Posts and Telecommunications, China
§Nanyang Technological University, Singapore ‖Telecom Paris, France
‡Corresponding author, email: qiuhan@tsinghua.edu.cn

***Abstract*—Deep learning (DL) based perception models have enabled the possibility of current autonomous driving systems (ADS). However, various studies have pointed out that the DL models inside the ADS perception modules are vulnerable to adversarial attacks which can easily manipulate these DL models' predictions. In this paper, we propose a more practical adversarial attack against the ADS perception module. Particularly, instead of targeting one of the DL models inside the ADS perception module, we propose to use one universal patch to mislead multiple DL models inside the ADS perception module simultaneously which leads to a higher chance of system-wide malfunction. We achieve such a goal by attacking the attention of DL models as a higher level of feature representation rather than traditional gradient-based attacks. We successfully generate a universal patch containing malicious perturbations that can attract multiple victim DL models' attention to further induce their prediction errors. We verify our attack with extensive experiments on a typical ADS perception module structure with five famous datasets and also physical world scenes[1].**

***Index Terms*—Deep learning, adversarial attack, autonomous driving system, computer vision.**

I. Introduction

The recent rapid development of deep learning (DL) significantly accelerates the commercialization of Autonomous Driving Systems (ADS) [1]. Currently, the mainstream of ADS companies like Tesla Autopilot[2] is to integrate multiple powerful vision-based DL models as perception modules [2] to perceive the environments for decision making. There are three typical vision-based DL models deployed in one ADS perception module including line detection (LD) [3], traffic sign detection (TSD) [4], and object detection (OD) [5]. A successful operation of these three models can guide a vehicle to autonomously decide the driving. The motivation of ADS is to revolutionize transportation by improving driving safety and efficiency. However, many recent research works point out that these DL-based perception modules present novel security challenges that threaten the usage of ADS [6], [7].

One of the most important security issues is the adversarial attacks against DNN models. Various recent studies [8], [9] have pointed out that attackers can add human-imperceptible but carefully crafted perturbations on input samples to manipulate the DNN models' predictions. The core idea of generating Adversarial Examples (AEs) is to calculate the gradients of the predictions with respect to the inputs (i.e., the input sample's pixel values). Then, attackers can modify the input sample iteratively guided by the sign of these gradients (e.g. FGSM [9], PGD [10], etc.) or the exact gradient values (e.g. CW) [11] within a pre-defined bound like L_∞ or L_2 [11].

Although these methods have shown that ADS built on DNN models are potentially vulnerable to these adversarial perturbations which may cause catastrophic failures like accidents [12], [13]. Most of the existing adversarial attacks cannot be directly deployed for the physical ADS scenarios. There are two main reasons summarized as follows. First, the adversarial attack methods mainly rely on slightly modifying the pixel values to guarantee its stealthiness goal (i.e. human-imperceptible compared with benign samples). These generated AEs will not work in real-world scenarios since the different lightning environments or distances will significantly perturb the pixel-level perturbations which decrease the effectiveness of the AEs [14], [15]. Thus, gradient-based perturbation generation is precise but fragile considering the dynamic real-world scenarios. Second, modern perception modules are always built by several DL models together. This means successfully attacking one of these DL models will not necessarily cause a system-wide failure since other models can still give correct predictions to help decisions. Thus, misleading more DL models can increase the attack success rate (ASR) to induce the system-wide failure of the ADS perception module. One example is given in Fig. 1 that an accident will occur only when all three DL models fail together. Thus, those works focusing on attacking in real-world scenarios may succeed in misleading one DL model but cannot cause the whole ADS perception module to fail.

To solve the above challenges, we introduce ATTA, a novel Adversarial Task-Transferable Attack against the perception modules of current ADS. The key insight of ATTA is made up of two aspects. First, our goal is to generate a *universal adversarial patch* to cause malfunctions in the perception modules. Particularly, we briefly classify the possible errors of the main DL models inside the perception modules and generate a patch with certain perturbations to induce wrong predictions on multiple DL models. As long as more than one DL model inside the perception modules gives wrong predictions, the vehicles will have a higher possibility of making wrong decisions and accidents may happen. This will require a *task-transferability* that one universal adversarial patch can

[1]We release our code at https://github.com/qingjiesjtu/ATTA
[2]https://www.tesla.com/autopilot

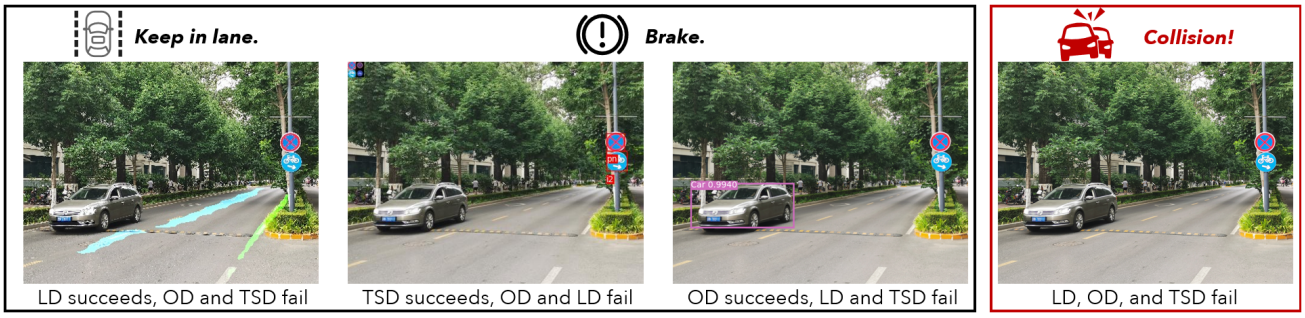

Fig. 1. Scenes where only all failures of tasks will cause a collision.

trigger wrong predictions across different tasks. Moreover, to make the attack more practical, we need this patch can be *scene-transferability* such that the patch can attack models when patched in a new environment (plug and play). The second insight of our work is to target the attention of the DL models instead of the gradients to guide the generation of the adversarial patch. As a higher-level representation of a DL model's prediction, extracting the attention from a given sample can generate more reliable perturbations that outperform the gradient-based methods. Our extensive experiments (see Section V) show that ATTA can achieve higher ASR considering both task- and scene-transferability to cause wrong predictions for multiple DL models. Our contribution can be summarized as follows.

- We present a novel attention-based adversarial attack that aims to induce system-wide malfunction of the ADS perception module by using one patch to cause wrong predictions of multiple DL models.
- Based on extensive experiments, we demonstrate both task- and scene-transferability of ATTA, suggesting that similar vulnerabilities may exist across many different ADS tasks.
- Finally, we experiment ATTA in a real-world setting to highlight the practical implications of our findings.

II. Preliminaries

In this section, we first briefly summarize the existing research on adversarial attacks both in the digital world and the physical world. Then, we specify the actual challenge of deploying adversarial attacks against the perception modules of ADS. We also introduce the attention mechanism of DL models which is the novel vulnerability used in our paper.

A. Adversarial attacks against DNNs

Digital adversarial attacks. The typical definition of AE is that an attacker can add human-unnoticeable perturbations on the benign inputs to fool a DL model. Formally, as in Eq. 1, the target DL model is a mapping function f. Given a clean input sample x, the corresponding AE is denoted as $\widetilde{x} = x + \delta$ where δ is the adversarial perturbation. δ is constrained by certain metric (e.g. L_p norm) to make it imperceptible. Then AE generation can be formulated as the optimization problem in Eq. 1a (targeted attack where $l' \neq f(x)$ is the desired label set by the attacker, e.g. an image is misclassified specifically as the label pre-set by the attacker) or Eq. 1b (untargeted attack, e.g. an image is misclassified as an arbitrary class other than its correct label.).

$$min\|\delta\|, s.t.\ F(\widetilde{x}) = l' \tag{1a}$$
$$min\|\delta\|, s.t.\ F(\widetilde{x}) \neq F(x) \tag{1b}$$

Directly using the above attack method will potentially change all pixels in one image. This will not be practical in most real-world scenarios especially considering the ADS. For instance, the perturbations generated under techniques like PGD [10], [16] require to cover a very large part of the whole scene (e.g. modify a fixed background imagery such as half of the sky).

Physical adversarial attacks. The initial attempt to generate physical AE is to first define a mask and then modify pixels only within this mask [17]. This method can limit the perturbation and is easy to deploy by just printing the adversarial patch and putting it in a certain position in the real-world environment [17], [18]. Later, other physical adversarial attacks are proposed to achieve the human-imperceptible goal. For instance, attackers can turn a commonly seen object (e.g. dirt on the roads [19] or advertisement board [15]) into an adversarial one with physical perturbations. These approaches mainly focus on misleading one of the DL models (e.g. only LD model [19]) inside the perception module which may not succeed in inducing a system-level malfunction of ADS.

B. ADS perception module

Perception modules in ADS. Modern perception module typically contains three DL models: obstacle detection (OD), lane detection (LD), and traffic sign detection (TSD) [20]. OD is a crucial function that identifies obstacles that may obstruct the vehicle's movement. Failing to recognize obstacles can pose significant risks to both vehicles and pedestrians, as demonstrated in the Tesla accident [21]. In this paper, we focus specifically on in-road obstacle detection, such as front cars and pedestrians. LD is another important model that utilizes camera image data to detect lane boundaries on the road. The accuracy of lane detection directly impacts obstacle localization and driving decisions. In a separate incident [22], a Tesla vehicle collided with a guardrail on a highway due to a failure in recognizing the right-turn lane. ADS companies predominantly employ end-to-end DL-based solutions for the LD task. For example, segmentation-based approaches [23]

have shown remarkable performance in various lane detection challenges and have been successfully implemented in commercial ADSs. Polynomial-based approaches, such as PolylaneNet [24], offer real-time and lightweight features and are deployed in production-grade ADSs like Openpilot [25]. TSD is the function that helps AVs recognize the status of traffic lights at intersections. Most ADSs employ end-to-end DL models with high precision to locate and identify traffic signs[3]. Yolo is utilized to locate the traffic lights, and a typical convolutional neural network (CNN) model, specifically RetinaNet, is employed for sign classification [26].

Challenges to attack perception modules. Autonomous driving is an integrated decision process based on the output of multiple DL models in the perception module [27]. Each DL model has its unique task to give predictions of the environments. These DL models are interconnected, forming a complex system in which a failure in one module does not necessarily result in a system-wide failure. For instance, if the TSD model fails to detect a stop sign, it would not necessarily cause a collision if the obstacle detection module is still functioning correctly. In such a case, the OD model would detect an approaching vehicle and signal the vehicle to stop. Likewise, if the OD model fails, it would not necessarily lead to a collision if the TSD model can identify a stop sign. The interconnected nature of these models allows for redundancy and enhances the overall safety and reliability of the ADS. An example is given in Fig. 1 which indicates all three models fail can trigger a wrong driving decision. Consequently, it is important to explore a universal adversarial patch that is capable of affecting multiple DL models of a perception module simultaneously to exploit these interconnected vulnerabilities and potentially cause system-wide malfunction.

C. Attention of DL models

In the context of computer vision, attention refers to a mechanism that allows the DL model to focus on specific regions or features of a sample during the inference. It is inspired by the selective attention mechanism in human visual perception [28]. Attention helps the model to prioritize and allocate its computational resources to the most relevant parts of an image, rather than treating the pixels of the entire image uniformly. DL models, although differing in architecture and functionality for different tasks, are found to share common attention mechanisms [29], [30]. Considering attention information as a high-level representation [31], our intuition is to introduce disruptive attention information to potentially alter the entire perception process across different tasks and models. Existing gradient-based adversarial attacks usually focus on one task which makes it hard to transfer to another task (e.g. an attack exploiting lane features works for lane detection but will fail on traffic sign detection). Thus, in this paper, instead of focusing on gradients with respect to specific samples and model weights, our insight is to ignore task-specific features which may hinder transferability between different models.

[3]https://www.autoware.org/

III. Problem Description and Threat Models

A. Problem description

As indicated in Section I, generating human-imperceptible perturbations in the digital domain or physical adversarial objects can only cause wrong predictions for a single model. Considering a typical ADS perception module contains three DL models (i.e. LD, OD, and TSD), our first problem to solve is to generate a patch that can induce the wrong prediction of at least two DL models simultaneously. This means our attack can be transferred between models for different tasks. Thus, our attack can have a much higher ratio for a potential system-wide malfunction of the ADS. The second problem we aim to solve is to achieve scene-transferability such that our adversarial patch can be used to attack ADS perception modules in a new scene in a plug-and-play manner.

B. Threat model

Our threat model encompasses the adversary's goal, his knowledge, and his capabilities in the context of a typical ADS perception module.

Adversary's goal. The general goal of the adversary is to cause system-wide malfunctions of the ADS perception modules by misleading at least two DL models simultaneously with one patch. Considering the attack on each DL model, we briefly classify three specific goals as *disappear*, *generate*, and *false-detect*. In order to evaluate the attack of task-transferability, we also set a *total* goal defined as any of the above three goals achieved.

- Disappear: any target object is not detected.
- Generate: any non-existent object is detected.
- False-detect: any object is false-detected.
- Total: either of the above goals is achieved.

Adversary's knowledge. Adversary's knowledge consists of two aspects: the environment to deploy the attack and the access to the victim DL models of the perception modules. For the former, we follow the settings of the previous works [19] and assume full knowledge because public roads are open to anyone and pasting patches is possible. For the latter, we consider two scenarios:

- White-box: the adversary has full knowledge of the perception module, including the architecture and weights of its backbone model.
- Black-box: the adversary is unaware of the internal architecture of the perception module but can obtain feedback through querying the model.

Adversary's capability. The adversary is capable of crafting and sticking adversarial patches either physically or digitally to the environment that the ADS perceives. We assume the training datasets and the corresponding DL models of the perception modules have integrity. Besides pasting patches in the environments, the adversary does not have the capability to interfere directly with the ADS's hardware or software.

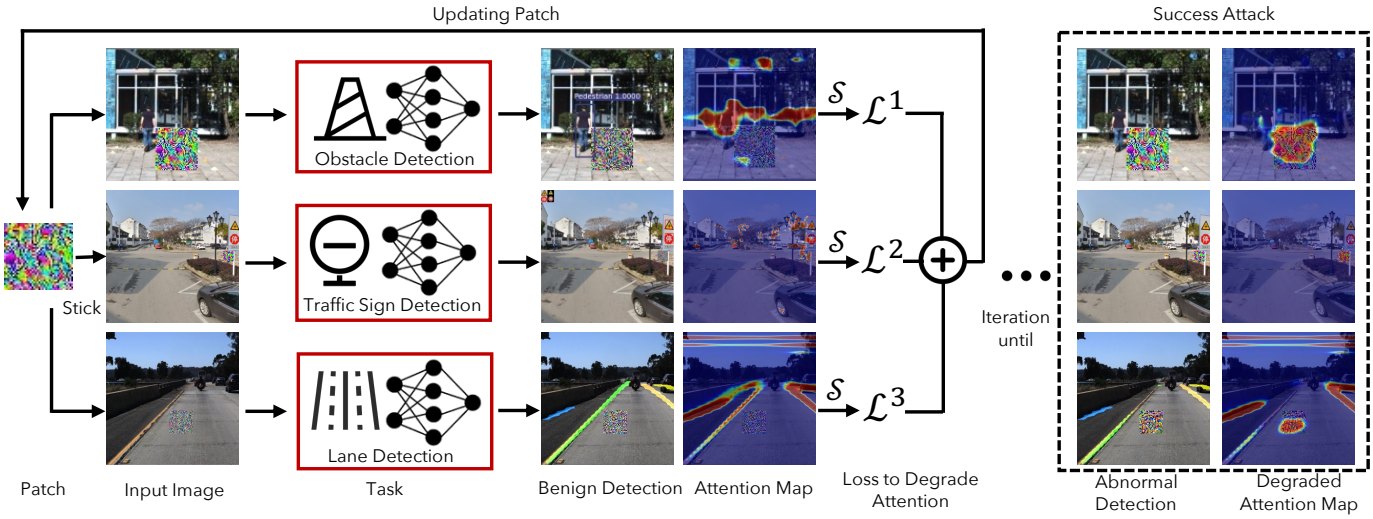

Fig. 2. Overview of ATTA. A universal adversarial patch is crafted through an iteration procedure with multiple branches, corresponding to multiple tasks. The iteration continues until all tasks give an abnormal prediction.

IV. Methodology

The methodology is organized as follows. We first describe the overview of ATTA. We then detail the loss function proposed to degrade the attention information of DNNs. After that, we detail a soft thresholding function to better redistribute the attention map. Finally, we present the algorithm of the iteration procedure to generate the adversarial patches.

A. Overview

Overview of ATTA is in Fig. 2. ATTA crafts a universal adversarial patch specifically designed to simultaneously disrupt the attention maps of multiple perception modules in ADS. When stuck to the input images for different DNNs, such a universal adversarial patch will induce the wrong prediction of all perception modules, leading to a system-wide malfunction of ADS. To achieve this, we first propose attractive loss $\mathcal{L}_a$ to degrade the attention maps in order to hinder the comprehension of DL models. This loss function aims to destroy the high-level feature extraction ability across perception modules, ignoring the task-specific features which may hinder task-transferability. Then, to intensify the deterioration of attention caused by our proposed loss function, we employ a soft thresholding function $\mathcal{S}$ which filters out low values and magnifies gradients of central values. Such a design adjusts the attention map closer to a semantic representation, enhancing the effectiveness of ATTA. Last, we summarize the iteration procedure as an algorithm to facilitate dynamic adjustments and continual fine-tuning of the universal adversarial patch. Such an algorithm is compatible with uni/bi/tri-task, considering the number of white-box perception modules.

B. Attractive loss

Given an input image x and the target model f, $\mathcal{A}(x; f)$ is the attention map obtained by Grad-CAM [32]. Our initial idea is to degrade the attention information learned by DL models. In consideration of ensuring high transferability, the way to degrade attention should not consider task-specific features. This can be realized by designing a loss function to disturb the victim DL model's attention. Particularly, DL models will attach more magnitude to the pixels favorable for feature extraction. The attention distribution on pixels is therefore the target we aim to degrade. An intuitive idea is to distract the victim DL model's attention in a uniform distribution. Indeed, a uniform distribution of attention maps signifies that the victim DL models recognize nothing from the input image.

This is the most straightforward solution since such a design will work theoretically if the attention map can be modified very close to a uniform distribution. However, it is hard to achieve such an attention map in practice. There are two reasons analyzed as follows. First, this goal is naturally very hard to achieve since the DL models usually have sophisticated architectures and the attention for any input image has a huge gap from a uniform distribution. This will introduce practical obstacles for adversarial patch generation. Second, during the crafting procedure of an adversarial patch, even a randomly initialized patch will naturally attract more attention from DNNs, weakening the effectiveness of this idea. We experimented this straightforward solution and the results are given in Section V-F.

Considering the above difficulty, a more appropriate and practical idea is to trick the victim DL models into only focusing on the adversarial patch and ignoring all environmental information in the meantime. This can introduce a distraction for the victim DL models compared with their correct prediction procedure. We propose an *attractive loss* $\mathcal{L}_a$ to achieve the above goal:

$$\mathcal{L}_a = \|\mathcal{A}(x; f) - \mathcal{M}\|_2 \tag{2}$$

where $\mathcal{M}$ is the mask of patch. $\mathcal{M}$ has the same dimension of x and for the pixels in (resp. out of) the patch region, the value equals 1 (resp. 0). Intuitively, $\mathcal{L}_a$ aims to attract the

DL model's attention to a patch irrelevant to environmental perception. To use an analogy, if a driver is attracted by the billboard on the roadside during driving, his attention on the road condition will decrease which may cause danger.

C. Soft threshold function $\mathcal{S}$

In the process of crafting adversarial patches, we observe that the attention map, which lies in the range of [0,1], does not linearly represent the semantic information. The semantic difference between low values (e.g. 0.2 and 0) is relatively small compared to the difference between more centrally located values (e.g., 0.6 and 0.4). Intuitively, degrading a pixel's attention value from 0.2 to 0 is not as harmful as from 0.6 to 0.4. When deriving the loss function to update the adversarial patches, we also derive the attention map. Such a nonlinear relationship between the attention map and the semantics may lead the update to focus on small semantic changes at low values while overlooking substantial semantic changes in the central region.

To address this issue, an integral part of ATTA is a *soft thresholding function* $\mathcal{S}$ on attention map [33], [34] before calculating the loss value:

$$\mathcal{L}_a = \|\mathcal{S}(\mathcal{A}(x; f)) - \mathcal{M}\|_2 \tag{3}$$

$$\mathcal{S}(\mathcal{A}(x; f)) = \frac{1}{1 + \exp\left(-\omega(\mathcal{A}(x; f) - \sigma)\right)} \tag{4}$$

where σ is the threshold ensuring that $\mathcal{S}(\mathcal{A}(x; f))$ is approximately equal to 0 when $\mathcal{A}(x; f)$ is much less than σ. And ω is the scale parameter to magnify the gradient in the central region of [0, 1].

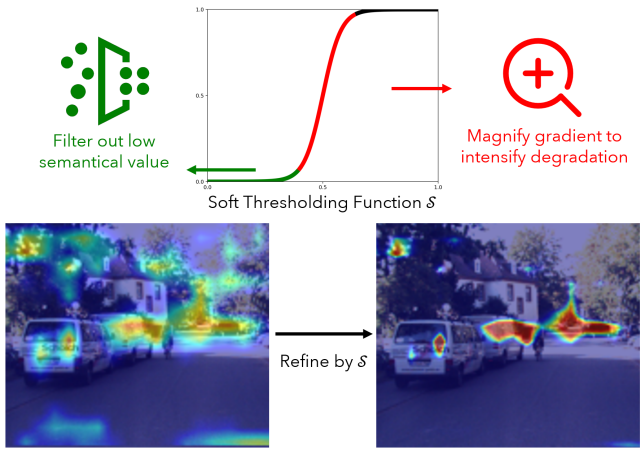

Fig. 3. Mechanism of soft thresholding function $\mathcal{S}$. It filters out low semantical values in the attention map, making it concentrate more on high semantic information. $\mathcal{S}$ also intensifies the degradation of attention information by magnifying the gradient to update the adversarial patch.

Figure 3 shows the mechanism of $\mathcal{S}$. On the one hand, it filters out low values which contribute little to the deterioration of semantical information. On the other hand, it intensifies the degradation of critical semantical values by magnifying their gradients. As a result, the attractive loss can be more effective in modifying the DL model's semantic capturing capability, enhancing the effectiveness of ATTA.

D. Algorithm of iteration procedure

Our universal adversarial patch is trained through an iterative process that begins with an initial random patch. Considering the feasibility of applying the attack in real-world scenarios, this initial patch is stuck to the road scenes in the input images [35]. During each iteration, we use $\mathcal{L}_a$ to degrade the attention information perceived by the victim DL model. The patch is gradually refined until we reach the adversary's goals. We expect three attack goals as indicated in Section III-B. The adversarial patch for each goal can be different due to the conflict of these goals (e.g. if an object is not detected, it cannot be false-detected). We therefore craft three adversarial patches during an iterative procedure: $\mathcal{P}_d$ for disappearing attack, $\mathcal{P}_g$ for generation attack, and $\mathcal{P}_f$ for false-detection attack. Algorithm 1 details our iterative procedure. It can be adapted to uni/bi/tri task, depending on the cardinal of task index set $\mathcal{K}$.

Algorithm 1 Crafting universal adversarial patch.

input: task index set $\mathcal{K}$, image set $\mathcal{X} = \{x_k, \forall k \in \mathcal{K}\}$, victim model set $\mathcal{F} = \{f_k, \forall k \in \mathcal{K}\}$, coefficient set $\mathcal{C} = \{c_k, \forall k \in \mathcal{K}\}$, iterations N, patch size s
output: adversarial patches for three attack goals $\mathcal{P}_d$, $\mathcal{P}_g$, $\mathcal{P}_f$

Initialize three success indicators $\mathcal{I}_d$, $\mathcal{I}_g$, $\mathcal{I}_f$ to 0
Randomly initialize a patch $\mathcal{P}$ of size s
$\forall k \in \mathcal{K}$, stick $\mathcal{P}$ on the road scene of x_k to get $x_k \oplus \mathcal{P}$
while $i < N$ **and** $\mathcal{I}_d \cdot \mathcal{I}_g \cdot \mathcal{I}_f = 0$ **do**
  $\forall k \in \mathcal{K}$, do inference on $x_k \oplus \mathcal{P}$ with f_k
  if $\mathcal{I}_d = 0$ **and** $\forall k \in \mathcal{K}$, disappear happens in f_k **then**
    $\mathcal{I}_d \leftarrow 1$; $\mathcal{P}_d \leftarrow \mathcal{P}$
  end if
  if $\mathcal{I}_g = 0$ **and** $\forall k \in \mathcal{K}$, generate happens in f_k **then**
    $\mathcal{I}_g \leftarrow 1$; $\mathcal{P}_g \leftarrow \mathcal{P}$
  end if
  if $\mathcal{I}_f = 0$ **and** $\forall k \in \mathcal{K}$, false-detect happens in f_k **then**
    $\mathcal{I}_f \leftarrow 1$; $\mathcal{P}_f \leftarrow \mathcal{P}$
  end if
  $\forall k \in \mathcal{K}$, compute the attention map on $x_k \oplus \mathcal{P}$ to get $\mathcal{A}(x_k \oplus \mathcal{P}; f_k)$
  Update $\mathcal{P}$ with loss $\mathcal{L} = \Sigma_{k \in \mathcal{K}} c_k \mathcal{L}^k$ where $\mathcal{L}^k$ is computed on $\mathcal{S}(\mathcal{A}(x_k \oplus \mathcal{P}; f_k))$
end while
return $\mathcal{P}_d$, $\mathcal{P}_g$, $\mathcal{P}_f$

V. Evaluation

A. Experimental setup

Victim tasks and datasets. Table I shows the tasks and datasets. Our experiments are conducted using a variety of perception modules within ADS. They are selected to represent a wide range of tasks and architectures, testing the universal adversarial patch's ability to affect different models. Note that LD is a segmentation task rather than an object detection task. This expands the research scope of our work because theoretically there is no natural transferability between different target

models. We also use several datasets to evaluate our patch, ensuring its generalization ability across different data sources. These datasets are selected to cover a range of scenarios that an ADS might encounter in the real world, providing a comprehensive evaluation of the patch's effectiveness. Noting that the resolution of images differs with the datasets. The relative size of the universal adversarial patch in the image is therefore different because its pixel size is fixed to 20×20.

TABLE I
TASKS, MODELS, AND DATASETS.

Task	Model	Dataset
Obstacle Detection	Yolov3 [36]	KITTI [37]
Traffic Sign Detection	Yolov3 [36] + ResNet18 [38]	CCTSDB_changsha [39] GTSDB [40] tsinghua-tencent 100k [41]
Lane Detection	SCNN [42]	TuSimple [43]

Baseline attacks. We conduct experiments of two classic methods for generating adversarial examples (CW [11] and PGD [10]). Noting that both two baselines are designed for targeted attacks. To adapt to our untargeted attack scenario, we reformulate their loss functions for magnifying the difference between predictions of images with/without patches. Besides, CW and PGD allow updating on the whole image. We restrain their updating in the patch region.

Metrics. To prove the effectiveness of ATTA, we use the Attack Success Rate (ASR) as our primary metric. The ASR measures the proportion of cases where the patch successfully causes the model to produce the wrong output. We calculate the ASR in several ways to match our attack goals:

- Disappear ASR: the proportion where any object is not detected for OD and TSD (resp. a great portion of lane is not segmented for LD).
- Generate ASR: the proportion where any non-existent object is detected for OD and TSD (resp. an ignorable portion of lane is incorrectly segmented for LD).
- False-detect ASR: the proportion where any object is false-detected for OD and TSD (resp. the lane segmented is in the wrong direction).
- Total ASR: the proportion where any above goal happens.

Besides, the ASR can be computed on a single task and on the overall system. The latter is more critical in multi-task scenarios for evaluating system-wide failure.

Experiments settings. In Figure 2, it seems ATTA is specifically designed for attacking tri-task. To provide a comprehensive evaluation of our methodology, we actually conduct experiments following three settings, corresponding to the number of victim tasks and input images:

- Uni-task: stick a patch on an input image (from public datasets) and feed it into one model.
- Multi-task: stick a patch on multiple input images (from public datasets) and feed them into multiple models.
- Physical world: stick a patch on an input image (from real-world scenarios) and feed it into multiple models.

In the following subsections, we first conduct experiments on uni-task to prove ATTA's effectiveness and reveal each task's characteristics under attack. Then, we evaluate ATTA's performance in our proposed multi-task scenario to achieve a system-wide failure. Then, we do a complete study on scene-transferability and task-transferability. Then, we evaluate ATTA in the physical world. At last, we do an ablation study to reveal the effectiveness of $\mathcal{L}_a$ and $\mathcal{S}$.

B. Uni-task

Although ATTA is designed to simultaneously affect multiple tasks within ADS, Table II shows that its performance is comparable, even better than baselines when adapted to the uni-task scenario. This is due to the vulnerability of attention information. Indeed, baseline attacks only focus on the final output rather than exploiting the internal architecture of DNNs. ATTA's strength benefits from the focus on high-level attention information. In addition, we also observe that ASR varies in tasks and attack goals. For OD and TSD, generate attack is the easiest to achieve because the adversarial patch could successfully grab the attention of the model and let it explore some objects in the area filled by the patch. For LD, disappear goal is the easiest to achieve because the LD task work by classifying every pixel so that it is much more sensitive to pixels change near the traffic lane. Within all three tasks, LD is the hardest to attack because the update of the attention map tends to form a lane-like pattern (shown in Figure 4).

TABLE II
UNI-TASK ASR OF ADVERSARIAL PATCH FOR ONE TASK.

Task	Attack goal	Method CW	PGD	ATTA
OD	Disappear	36.6	37.1	**51.7**
	Generate	82.7	93.1	92.2
	False-detect	34.5	32.7	28.4
	Total	86.2	94.8	94.0
TSD	Disappear	6.5	6.5	**20.4**
	Generate	12.9	47.3	**82.8**
	False-detect	0.0	0.0	**8.6**
	Total	12.9	47.3	**91.4**
LD	Disappear	2.0	6.9	**69.3**
	Generate	1.0	12.9	8.9
	False-detect	0.0	20.9	1.2
	Total	3.0	29.7	**69.3**

C. Multi-task

The motivation of ATTA is to craft a universal adversarial patch that can simultaneously affect multiple perception modules. We conduct complete experiments, including bi-task and tri-task, to prove the strength of ATTA in our proposed scenario. For each combination of multi-task, we compute ASR on a single task and on the overall system. Table III and Table IV show that ATTA achieves the best performance. We constate that ATTA is feasible to cause a system-wide failure where more than one task fails. This benefits from the fact that attention is the common vulnerability of DNNs.

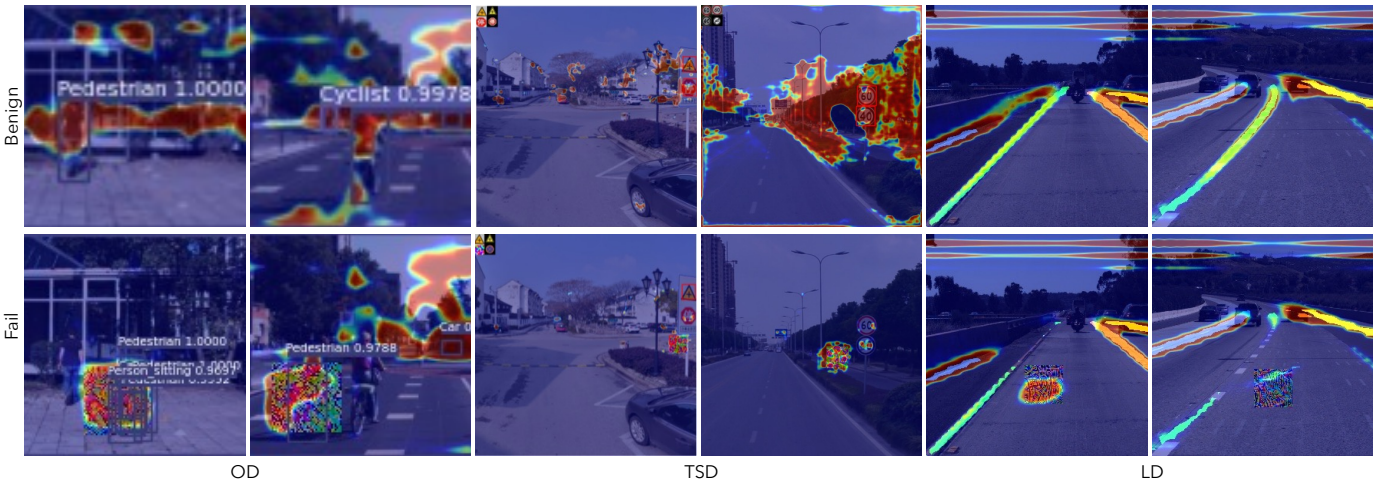

Fig. 4. Visualization results of ATTA. The first row is the benign results. The second row is the successful attack results. The attention maps are attracted by the universal adversarial patches, leading to malfunction of tasks: for OD, the pedestrian and the cyclist disappear and several non-existent objects are generated on the patch; for TSD, the red stop sign disappears, and another blue stop sign is generated on the patch; for LD, half of the green lane disappears.

TABLE III
BI-TASK ASR OF UNIVERSAL ADVERSARIAL PATCH FOR TWO TASKS.

Combination	Method	OD				TSD				OD+TSD
		Disappear	Generate	False-detect	Total	Disappear	Generate	False-detect	Total	Total
OD+TSD	CW	59.0	74.0	29.0	81.0	10.0	17.2	1.0	23.4	22.8
	PGD	65.0	97.0	35.0	97.0	18.0	27.0	4.0	40.6	35.9
	ATTA	**70.0**	**97.0**	34.0	**97.0**	**21.0**	**85.0**	**7.0**	**86.0**	**86.0**
Combination	Method	OD				LD				OD+LD
		Disappear	Generate	False-detect	Total	Disappear	Generate	False-detect	Total	Total
OD+LD	CW	45.5	56.4	27.7	67.3	97.0	2.9	1.0	97.0	19.1
	PGD	41.6	85.1	31.7	93.0	97.0	25.7	14.8	97.0	20.1
	ATTA	**51.5**	**89.1**	**31.7**	**93.1**	**97.0**	4.0	1.0	**97.0**	**56.0**
Combination	Method	TSD				LD				LD+TSD
		Disappear	Generate	False-detect	Total	Disappear	Generate	False-detect	Total	Total
LD+TSD	CW	12.0	16.0	3.0	20.0	100	3.0	1.0	100	7.2
	PGD	4.0	15. 0	1.0	16.0	100	30.0	14.0	100	4.0
	ATTA	11.8	**31.2**	**6.5**	**33.3**	**100**	5.4	1.1	**100**	**8.2**

TABLE IV
TRI-TASK ASR OF UNIVERSAL ADVERSARIAL PATCH FOR THREE TASKS.

Task	Attack goal	Method		
		CW	PGD	ATTA
OD	Disappear	36.0	40.0	**60.0**
	Generate	46.0	74.0	**77.0**
	False-detect	29.0	30.0	**33.0**
	Total	61.0	80.0	**84.0**
TSD	Disappear	8.0	5.0	**10.0**
	Generate	9.0	11.0	**26.0**
	False-detect	3.0	3.0	0.0
	Total	11.0	10.0	**26.0**
LD	Disappear	94.0	96.0	**97.0**
	Generate	0.0	0.0	**5.0**
	False-detect	1.0	2.0	1.0
	Total	95.0	96.0	**97.0**

Besides, an interesting finding is that for LD, the ASR for multi-task is higher than for uni-task. We can see that the attention on the patch in LD task is a diagonal shape, which illustrates that only these pixels contribute to the attention attraction while others are not. The patch in multi-task is updated by both OD, LD, and TSD task loss, which means the pixels in a patch are utilized more efficiently in LD task.

D. Transferability

In general, transferability means for the same task, an adversarial example of model A works for model B. This is designed to attack black-box model using a surrogate model of the same task. However, such transferability makes no sense if there is no surrogate model. Besides, even if the victim models are all white-box, we do not always have full permission to environmental information for crafting patches. In this paper, we propose two novel definitions of transferability, which are of more significance in real-world scenarios.

Scene-transferability. The adversary may not have time or specific environmental information to generate the adversarial patch (e.g. the environment varies with traffic and weather conditions; it is suspicious to take photos at an extremely important crossway). This yields the need for a **plug and play** universal adversarial patch. Table V and Table VI, when

TABLE V
TASK- AND SCENE-TRANSFERABILITY OF ADVERSARIAL PATCH CRAFTED IN UNI-TASK SETTING.

		From								
		OD			TSD			LD		
To	Attack goal	CW	PGD	ATTA	CW	PGD	ATTA	CW	PGD	ATTA
OD	Disappear	23.4	31.0	**33.5**	24.5	23.0	**28.7**	8.6	16.1	**37.5**
	Generate	32.9	58.6	47.2	33.2	43.8	**54.1**	11.0	34	**49.6**
	False-detect	19.1	24.5	**24.9**	26.2	21.0	25.8	28.0	25.4	24.6
	Total	55.6	59.8	**60.5**	52.5	57.5	**67.0**	35.7	69.8	63.8
TSD	Disappear	1.0	2.0	**5.4**	3.5	4.3	**15.1**	1.0	2.1	**6.5**
	Generate	1.0	1.0	**4.3**	2.2	6.5	**97.8**	0.0	2.1	**6.5**
	False-detect	1.0	0.0	**2.1**	1.1	2.1	**5.3**	1.0	1.0	**2.2**
	Total	2.1	2.1	**7.5**	3.2	7.5	**97.8**	1.0	3.2	**8.6**
LD	Disappear	2.8	2.9	**4.0**	2.9	3.9	**5.0**	2.9	5.9	**70.3**
	Generate	0.0	0.0	**2.0**	0.0	1.9	**3.0**	0.9	8.9	8.9
	False-detect	1.9	1.9	1.0	1.9	0.9	0.0	0.9	7.9	**19.8**
	Total	4.9	4.9	**5.9**	4.9	6.9	**6.9**	4.9	19.8	**78.2**

TABLE VI
TASK- AND SCENE-TRANSFERABILITY OF UNIVERSAL ADVERSARIAL PATCH CRAFTED IN MULTI-TASK SETTING.

		From											
		OD+TSD			OD+LD			LD+TSD			OD+TSD+LD		
To	Attack goal	CW	PGD	ATTA	CW	PGD	ATTA	CW	PGD	ATTA	CW	PGD	ATTA
OD	Disappear	25.0	24.5	**30.6**	31.4	21.1	**41.4**	28.6	26.7	**41.8**	24.7	30.5	**40.5**
	Generate	35.6	48.7	44.8	48.2	31.0	**50.0**	51.4	53.2	**55.6**	22.8	57.1	**65.1**
	False-detect	26.2	26.7	24.6	26.2	25.6	23.7	22.8	21.9	**23.3**	35.2	23.8	24.1
	Total	53.8	63.3	60.3	63.3	51.7	**63.4**	66.7	63.5	**67.2**	52.3	72.4	**75.9**
TSD	Disappear	3.2	4.3	**10.8**	4.3	2.2	**6.4**	5.4	3.2	4.3	4.3	1.1	3.2
	Generate	9.6	1.1	**32.3**	3.2	1.1	**4.3**	6.5	2.2	**7.5**	4.3	1.1	**7.5**
	False-detect	0.0	2.2	**4.3**	4.3	2.2	**4.3**	2.2	1.1	1.1	0.0	0	0
	Total	10.8	4.3	**36.6**	6.5	4.3	8.6	8.6	4.3	**8.6**	5.4	1.1	**9.7**
LD	Disappear	4.9	1.1	**5.0**	5.9	5.9	5.9	4.9	5.0	4.9	3.9	3.9	**100**
	Generate	0	0.9	3.0	0.9	0.9	1.0	0.9	1.0	1.0	1.0	2.9	2.0
	False-detect	1.9	5.9	0.0	0.9	1.9	**3.0**	1.9	1.9	**3.0**	1.9	1.0	1.0
	Total	5.9	6.8	**6.9**	6.9	6.9	**7.9**	6.9	6.9	**7.9**	5.9	7.9	**100**

TABLE VII
TRI-TASK AND SCENE-TRANSFERABILITY ASR IN PHYSICAL WORLD.

		Method					
		CW		PGD		ATTA	
Task	Attack goal	Tri	Trans	Tri	Trans	Tri	Trans
OD	Disappear	57.1	55.2	62.3	61.0	**87.2**	**71.6**
	Generate	62.3	52.2	79.2	59.7	**91.6**	58.2
	False-detect	50.6	52.2	55.8	49.3	**81.2**	49.3
	Total	83.1	78.1	90.9	80.6	**100**	80.1
TSD	Disappear	0.0	3.0	0.0	3.2	**20.2**	**35.82**
	Generate	0.0	0.5	0.0	3.4	**100**	**38.8**
	False-detect	0.0	0.0	0.0	0.0	1.5	0.0
	Total	0.0	3.0	0.0	3.4	**100**	**38.8**
LD	Disappear	57.5	22.2	77.9	0.0	**88.7**	**19.4**
	Generate	0.0	10.8	0.0	0.0	**19.6**	3.9
	False-detect	0.0	7.4	12.9	0.0	**21.8**	0.0
	Total	57.5	37.3	77.9	0.0	**97.0**	23.3
OD + TSD + LD	Disappear	0.0	0.0	0.0	0.0	**11.9**	**6.0**
	Generate	0.0	0.0	0.0	0.0	**10.6**	**1.5**
	False-detect	0.0	0.0	0.0	0.0	0.0	0.0
	Total	0.0	0.0	0.0	0.0	**21.0**	**7.5**

the transferred task is the subset of the source task, show the scene-transferability. We constate the existence of scene-transferability, ensuring the feasibility of the *plug and play* patch. And ATTA is better compared to baselines, especially for TSD and LD. However, scene-transferability is less powerful in multi-task settings than uni-task.

Task-transferability. When all tasks are not white-box, the adversary cannot craft a universal adversarial patch via Algorithm 1. He can only craft patch on white-box tasks and verify the transferability on black-box tasks. Table V and Table VI, when the transferred task is not the subset of the source task, show the task-transferability. We constate the feasibility of attacking black-box task without surrogate model. ATTA is better than baselines because all DNNs, whether aimed for the same task or not, exploit attention to understand the data. Among the tasks within ADS, we find that TSD is harder to transfer because the victim model we choose is combined with two sub-models (Yolov3 to detect the existence of traffic sign and ResNet18 to classify), compensating each other's vulnerability.

E. Physical world attack

We expect to deploy an attack in physical world, where the applicability of ATTA can be verified. Nevertheless, it

is generally more difficult to attack in physical world rather than in digital world, because the environmental information perceived varies with the light or traffic conditions. For the sake of fairness, we collect in total 76 images on road with default setting of camera. Each image contains obstacle, traffic sign, and road lane. Different from the experiment setting in digital world where we cannot find an image containing all three types of objects, the input images of all tasks are the same. Table VII shows the ASR and scene-transferability in physical world. Figure 5 shows the visual results.

Feasibility. We constate that it is feasible to craft a universal adversarial patch in physical world, leading to system-wide failure. ATTA exhibits a great strength compared to baselines, especially for the overall failure where baselines do not work anymore. The common vulnerability on attention level is therefore proved in physical world. And due to the same input image, it is more effective than dataset level.

Scene-transferability. ATTA is feasible to craft a *plug and play* patch for scene-transferability. This can cause destructive results if the adversary sticks the patch in any scene he encounters. The cost of attack is almost reduced to a minimum.

Fig. 5. Visual results of ATTA in physical world. We observe that a variety of obstacles disappear when we successfully attract attention to the patch.

F. Ablation Study

In Section IV, we theoretically state our design of attractive loss $\mathcal{L}_a$ and soft thresholding function $\mathcal{S}$. Here we give an ablation study and list their effectiveness in ATTA in Table VIII. The analysis are given as follows.

Effectiveness of $\mathcal{L}_a$. We compare $\mathcal{L}_a$ to the intuitive idea, uniform loss $\mathcal{L}_u$:

$$\mathcal{L}_u = \mathcal{V}(\mathcal{A}(x; f)) \tag{5}$$

We constate that $\mathcal{L}_a$ is better than $\mathcal{L}_u$. Especially for the generate attack in TSD, $\mathcal{L}_a$ even doubles or triples the ASR than $\mathcal{L}_u$. This is because the relative size of patch in TSD is the smallest.

Attracting attention to the patch will generate more non-existent traffic signs on the patch than distracting attention in a uniform distribution.

Effectiveness of $\mathcal{S}$. $\mathcal{S}$ increases the ASR, no matter the loss function. The improvement is also more remarkable for TSD. We infer that when the patch is relatively small compared to the size of input image, $\mathcal{S}$ is more critical because it filters out the distractive low values distributed on the whole image, and intensifies the disruption caused by the adversarial patch.

TABLE VIII
ABLATION STUDY.

Task	Attack goal	Method			
		$\mathcal{L}_u$	$\mathcal{L}_u + \mathcal{S}$	$\mathcal{L}_a$	$\mathcal{L}_a + \mathcal{S}$
OD	Disappear	56.0	64.6	56.4	59.1
	Generate	93.1	98.7	97.4	97.4
	False-detect	28.9	30.5	30.6	31.5
	Total	94.0	98.7	97.4	97.4
TSD	Disappear	9.7	10.7	16.1	20.4
	Generate	24.7	38.7	64.5	82.8
	False-detect	4.3	4.3	4.3	8.6
	Total	26.9	41.9	67.7	91.4
LD	Disappear	100	99.0	100	100
	Generate	3.9	4.0	4.0	15.8
	False-detect	18.8	14.9	8.9	36.6
	Total	100	100	100	100

VI. Discussion and Future Work

Transferability between tasks. The results of ATTA provide compelling evidence of a potential universal adversarial patch on ADS perception modules. However, we do note that the transferability for TSD is much more difficult. The reason can be briefly analyzed as there are two DL models used together for TSD which is different compared with the OD and LD. Attacking one model's attention can be easily transferred to another task with one model but is hard to be transferred to tasks with two models. Thus, our first future work is to investigate further enhancing the transferability between single-model tasks and multi-model tasks.

Potential defense. It is worth noting that even a low ASR on attacking all three DL models inside the ADS perception module can potentially lead to serious car accidents. We hope our research can inspire a novel defense design for these modules containing multiple DL models inside. We believe one possible defense strategy in the future can be exploring the detection scheme to filter out potential input samples with abnormal attention maps.

VII. Conclusion

In this paper, we presented a novel universal adversarial patch capable of affecting multi-task DL models simultaneously inside a typical ADS perception module. We studied two novel transferabilities and exploited a common vulnerability of attention information across different tasks from different DL models. The extensive experiments proved the effectiveness of our attack in uni/bi/tri-task scenarios and outperformed the baseline adversarial attacks.

Acknowledgement

This work is supported by the National Natural Science Foundation of China No. 62106127, Singapore Ministry of Education (MOE) AcRF Tier 1 RS02/19 and Tier 1 RG108/19 (S). The authors would like to thank Pierre Jouvelot from PSL, Mines Paris, France for his help during this work.

REFERENCES

[1] Z. Xiong, W. Li, Q. Han, and Z. Cai, "Privacy-preserving auto-driving: a GAN-based approach to protect vehicular camera data," in *2019 IEEE International Conference on Data Mining (ICDM)*, 2019, pp. 668–677.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[3] X. Han, G. Xu, Y. Zhou, X. Yang, J. Li, and T. Zhang, "Physical backdoor attacks to lane detection systems in autonomous driving," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2957–2968.

[4] F. Nuding and R. Mayer, "Poisoning attacks in federated learning: An evaluation on traffic sign classification," in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 2020, pp. 168–170.

[5] S. Chen, Z. Li, F. Huang, C. Zhang, and H. Ma, "Improving object detection with relation mining network," in *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020, pp. 52–61.

[6] A. Boloor, K. Garimella, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, "Attacking vision-based perception in end-to-end autonomous driving models," *Journal of Systems Architecture*, vol. 110, p. 101766, 2020.

[7] X. Han, Y. Zhou, K. Chen, H. Qiu, M. Qiu, Y. Liu, and T. Zhang, "ADS-lead: Lifelong anomaly detection in autonomous driving systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1039–1051, 2022.

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[11] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 39–57.

[12] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. 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*, 2020, pp. 293–308.

[13] H. Xu, Y. Li, W. Jin, and J. Tang, "Adversarial attacks and defenses: Frontiers, advances and practice," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3541–3542.

[14] F. Nesti, G. Rossolini, S. Nair, A. Biondi, and G. Buttazzo, "Evaluating the robustness of semantic segmentation for autonomous driving against real-world adversarial patch attacks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2280–2289.

[15] Z. Kong, J. Guo, A. Li, and C. Liu, "Physgan: Generating physical-world-resilient adversarial examples for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 254–14 263.

[16] L. Pengcheng, J. Yi, and L. Zhang, "Query-efficient black-box attack by active learning," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 1200–1205.

[17] D. Song, K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, and T. Kohno, "Physical adversarial examples for object detectors," in *12th USENIX workshop on offensive technologies (WOOT 18)*, 2018.

[18] S.-T. Chen, C. Cornelius, J. Martin, and D. H. Chau, "Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*. Springer, 2019, pp. 52–68.

[19] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen, "Dirty road can attack: Security of deep learning based automated lane centering under physical-world attack," in *Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[20] "Baidu apollo project," https://github.com/ApolloAuto/apollo/, 2020.

[21] "Tesla crash: Man who died in autopilot collision filmed previous near-miss, praised car's technology," https://www.abc.net.au/news/2016-07-01/tesla-driver-killed-while-car-was-in-on-//autopilot/7560126, 2016.

[22] "Tesla model x crashes on pennsylvania turnpike, owner says it was on autopilot," https://www.motortrend.com/news/tesla-model-x-autopilot-crashes-//pennsylvania-turnpike/, 2016.

[23] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *AAAI Conference on Artificial Intelligence*, 2018.

[24] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Polylanenet: Lane estimation via deep polynomial regression," in *International Conference on Pattern Recognition*, 2021.

[25] "Commaai openpilot," https://github.com/commaai/openpilot, 2020.

[26] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "Automatic ship detection based on retinanet using multi-resolution gaofen-3 imagery," *Remote Sensing*, vol. 11, no. 5, p. 531, 2019.

[27] M. H. Nazeri and M. Bohlouli, "Exploring reflective limitation of behavior cloning in autonomous vehicles," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1252–1257.

[28] S. Dodge and L. Karam, "Can the early human visual system compete with deep neural networks?" in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2798–2804.

[29] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *arXiv preprint arXiv:1612.03928*, 2016.

[30] X. Li, H. Xiong, H. Wang, Y. Rao, L. Liu, Z. Chen, and J. Huan, "Delta: Deep learning transfer using feature map with attention for convolutional networks," *arXiv preprint arXiv:1901.09229*, 2019.

[31] Y. Yuan, G. Xun, F. Ma, Y. Wang, N. Du, K. Jia, L. Su, and A. Zhang, "Muvan: A multi-view attention network for multivariate temporal data," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 717–726.

[32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[33] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9215–9223.

[34] X. Ouyang, J. Huo, L. Xia, F. Shan, J. Liu, Z. Mo, F. Yan, Z. Ding, Q. Yang, B. Song *et al.*, "Dual-sampling attention network for diagnosis of covid-19 from community acquired pneumonia," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2595–2605, 2020.

[35] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen, "Dirty road can attack: Security of deep learning based automated lane centering under {physical-world} attack," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3309–3326.

[36] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[37] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[39] J. Zhang, X. Zou, L.-D. Kuang, J. Wang, R. S. Sherratt, and X. Yu, "Cctsdb 2021: a more comprehensive traffic sign detection benchmark," *Human-centric Computing and Information Sciences*, vol. 12, 2022.

[40] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark," in *International Joint Conference on Neural Networks*, no. 1288, 2013.

[41] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[42] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, and W. J. Dally, "Scnn: An accelerator for compressed-sparse convolutional neural networks," *ACM SIGARCH computer architecture news*, vol. 45, no. 2, pp. 27–40, 2017.

[43] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," 2017.