Evaluating Adversarial Attacks on Driving Safety in Vision-Based Autonomous Vehicles

Jindi Zhang, Yang Lou, Jianping Wang, *Senior Member, IEEE,* Kui Wu, *Senior Member, IEEE,*
Kejie Lu, *Senior Member, IEEE,* and Xiaohua Jia, *Fellow, IEEE*

Abstract—In recent years, many deep learning models have been adopted in autonomous driving. At the same time, these models introduce new vulnerabilities that may compromise the safety of autonomous vehicles. Specifically, recent studies have demonstrated that adversarial attacks can cause a significant decline in detection precision of deep learning-based 3D object detection models. Although driving safety is the ultimate concern for autonomous driving, there is no comprehensive study on the linkage between the performance of deep learning models and the driving safety of autonomous vehicles under adversarial attacks. In this paper, we investigate the impact of two primary types of adversarial attacks, perturbation attacks and patch attacks, on the driving safety of vision-based autonomous vehicles rather than the detection precision of deep learning models. In particular, we consider two state-of-the-art models in vision-based 3D object detection, Stereo R-CNN and DSGN. To evaluate driving safety, we propose an end-to-end evaluation framework with a set of driving safety performance metrics. By analyzing the results of our extensive evaluation experiments, we find that (1) the attack's impact on the driving safety of autonomous vehicles and the attack's impact on the precision of 3D object detectors are decoupled, and (2) the DSGN model demonstrates stronger robustness to adversarial attacks than the Stereo R-CNN model. In addition, we further investigate the causes behind the two findings with an ablation study. The findings of this paper provide a new perspective to evaluate adversarial attacks and guide the selection of deep learning models in autonomous driving.

Index Terms—Adversarial attacks, driving safety, 3D object detection, autonomous driving.

I. Introduction

OVER the past decade, autonomous driving has gained significant developments and demonstrated its great commercial potentials [1], [2]. The commercial potentials have attracted enormous investment as well as various malicious

attacks [3], [4], for example, close-proximity sensor attacks, remote cyberattacks, perturbation attacks, and patch attacks.

Environment perception and other tasks of autonomous driving systems heavily rely on deep learning models. Researchers have demonstrated that adversarial examples, which are originally designed to affect general-purpose deep learning models, can also be used to cause malfunctions in autonomous driving tasks [5]–[14]. In these studies, researchers usually use the decline of accuracy, or the erroneous rate increase of the deep learning models, to measure the effectiveness of attacks. Amplified by media reports, these attacks are casting cloud and posing psychological barriers to the broader adoption of autonomous driving [15].

From the perspective of autonomous driving, however, the ultimate concern is driving safety. Without a doubt, the inaccurate detection results of a deep learning model in the presence of attacks may impact driving safety, and in some situations, misdetection of traffic signs [7] might have disastrous consequence. Nevertheless, *driving safety is a combined effort of many factors in a dynamic environment*, and the deteriorated model performance does not necessarily lead to safety hazards. The linkage between the performance of a deep learning model under adversarial attacks and driving safety is not studied in the literature. In particular, there are no clear answers to the following questions: 1) Does the precision decline or the erroneous rate increase of the deep learning models under attacks represent their robustness in regard to driving safety of autonomous vehicles? In other words, does a larger decline in accuracy of an attacked deep learning model indicate a higher risk of driving safety? Similarly, does a slight decrease in accuracy of a deep learning model under attacks indicate mild risk of driving safety? 2) If the answers to the previous questions are all no, what are the reasons behind?

In this paper, we aim to answer the aforementioned questions by evaluating the impact of two types of representative adversarial attacks, *perturbation attacks* and *patch attacks*, on driving safety of vision-based autonomous driving systems, rather than the accuracy of deep learning models. We also investigate the reasons causing the decoupling between the detection precision of adversarial attacks and driving safety.

This study considers vision-based autonomous driving which mainly relies on stereo cameras for the task of environmental sensing. The vision-based object detectors that we consider in this paper are Stereo R-CNN [16] and DSGN [17], two state-of-the-art methods in this area.

To facilitate this study, we propose an *end-to-end* driving safety evaluation framework with a set of designed driving

safety performance metrics, where the evaluation framework can directly take the results of the 3D object detector as input and outputs the scores of the driving safety performance metrics as the final assessment.

To implement such an evaluation framework, we are faced with two nontrivial technical challenges. First, the results of the 3D object detector only contain static information, such as position and dimension. Thus, we cannot determine which objects are moving and which are static. Second, to realistically generate a planned trajectory for the self-driving ego-vehicle, real driving constraints, such as speed limits for different road types and dynamics models for different vehicles, must be provided to the motion planning module. Considering that the driving scenarios change dynamically, we need to select appropriate real driving constraints accordingly for driving safety assessment.

To tackle the first challenge, we train a CNN-based classifier with manually labeled ground truth to categorize whether an object is moving or static. For the second challenge, we train another classifier with road type labels to classify the road type of each scenario, so as to select appropriate driving constraints.

To obtain comprehensive experiment results, we apply the aforementioned two types of adversarial attacks with different attack intensities in our evaluation framework and measure the rate that the motion planner successfully finds a trajectory, the rate of collision occurrence, and the rate that the ego-vehicle drives safely from the initial position to the goal region. In the meantime, we also measure the precision changes of the vision-based 3D object detectors when they are under attacks. By linking the impact of adversarial attacks on driving safety and on 3D object detection together, we manage to find the answers to our motivation questions. The main contributions of this paper can be summarized as follows.

- We propose an *end-to-end* driving safety evaluation framework that directly takes the produced results of the 3D object detector as input and outputs driving safety performance scores as the evaluation outcome. With modular design, each individual module can be easily replaced so that the framework can be adapted to evaluate the driving safety of different self-driving systems threatened by various attacks.
- We conduct extensive experiments and observe that the changes in object detection precision and the changes in driving safety performance metrics caused by adversarial attacks are decoupled. Therefore, the answers to our motivation questions are all no. And we also observe that DSGN is more robust than Stereo R-CNN in terms of driving safety.
- We investigate the reasons behind our answers to those questions. We find that the reason for the decoupling is that it is easier for perturbation attacks to mislead object detectors to detect ghost objects at roadside which cause little influence on driving safety but huge impact on detection precision. We also find out that the reason why DSGN is more robust than Stereo R-CNN is that the latter is purely based on deep neural network, while DSGN adopts the Spatial Pyramid Pooling (SPP) in its architecture which can alleviate the attack effects.

The rest of this paper is organized as follows. In Section II, we first introduce the studies related to our work. In Section III, we briefly introduce the attack model of the two adversarial attacks studied in this paper. In Section IV, we elaborate on our proposed end-to-end evaluation framework and the driving safety performance metrics. In Section V, we present the experiment design and results. In Section VI, we investigate the causes of our observations with an ablation study. Finally, we conclude the paper in Section VII.

II. Related Work

In this section, we review the related work from the perspectives of attacks on autonomous driving, vision-based 3D object detection, and motion planning.

Attacks on Autonomous Driving. A survey of general vulnerabilities in autonomous driving can be found in [3], [4]. Among all the vulnerabilities, the perturbation attack and the patch attack are the most dangerous threats of vision-based autonomous driving systems, since they can directly cause damages to input images.

Both the perturbation attack and the patch attack fall into the domain of adversarial attacks. The main idea of adversarial attacks is to leverage small changes in the input to trigger significant errors in the output of deep learning models. According to [18], adversarial attacks can be either universal (effective to all valid inputs) or input-specific (only effective to a specific input). There are mainly two categories of methods to achieve adversarial attacks, namely, optimization-based methods and fast gradient step method (FGSM)-based approach. Optimization-based methods can be used for either universal attacks or input-specific attacks. An example is L-BFGS proposed by Szegedy *et al.* [19]. FGSM-based methods include FGSM [20] and its extensions, such as I-FGSM [21], MI-FGSM [22], and PGD [23]. These methods are usually used only for input-specific attacks.

The perturbation attack and the patch attack work in different ways. The perturbation attack usually affects all pixels in an input image but the changes in pixel values are very small, while the patch attack only affects a small number of pixels but the changes in pixel value are larger. Both the attacks were studied concerning different functional modules needed in vision-based autonomous driving. For example, the perturbation attack was studied regarding sign classification in [7], 2D object detection in [8], semantic segmentation in [9], [24], and monocular depth estimation in [12], [13], while the patch attack was studied regarding lane keeping in [5], optical flow estimation in [6], 2D object detection in [10], [11], and monocular depth estimation in [13]. None of these studies, however, focus directly on the attacks' impact on driving behavior and driving safety of autonomous vehicles.

Vision-Based 3D Object Detection. Vision-based 3D object detection provides a more budget-friendly approach to perform object detection in 3D space by mainly leveraging stereo cameras instead of expensive LiDARs. It is the core of vision-based autonomous driving. Traditional approaches, e.g., 3DOP [25] and Pseudo-LiDAR [26], first generate a pseudo point cloud with depth estimation and then perform 3D object detection with similar methods used in LiDAR-based 3D

object detection. As a result, they are usually not comparable to LiDAR-based methods in terms of accuracy and efficiency. Different from traditional approaches, Stereo R-CNN [16] and DSGN [17] are the two leading methods in this area. The network of Stereo R-CNN consists of a Region Proposal Network (RPN) and a regression part. The 2D bounding box candidates generated by the RPN are fed to the regression part where keypoints of 3D bounding boxes are predicted. The network of DSGN includes a single-stage pipeline which exacts pixel-level features for stereo matching and high-level features for object recognition. Both methods can achieve comparable performance to LiDAR-based methods.

Motion Planning. Motion planning is a key task for autonomous vehicles. Given an initial vehicle state, a goal state region, a cost function, and vehicle dynamics, motion planning finds collision-free trajectories. Currently, sampling-based motion planning algorithms are the mainstream methods. They can be viewed as a discrete planner, such as RRT [27], greedy BFS, and A* [28], in combination with a C-space sampling scheme.

III. ATTACK MODELS

We assess the impact of two types of adversarial attacks, perturbation attacks and patch attacks, on driving safety. Here, we briefly introduce the attack models.

Perturbation Attack. The goal of this type of adversarial attacks is to make the deep learning model dysfunctional by adding small changes to each pixel in the image that are imperceptible to human eyes. With prior knowledge of the deep learning model, attackers can launch perturbation attacks by tapping into the self-driving system and perturbing camera images. We consider the method of PGD [23] to achieve input-specific attacks. Consider a perturbation δ^{per} and an image pair (I_l, I_r), where δ^{per} has the same dimension as I_l and I_r. Let the initial perturbed image pair $(\tilde{I}_{l,0}^{\mathrm{per}}, \tilde{I}_{r,0}^{\mathrm{per}}) = (I_l, I_r)$. The attack is carried out by updating the perturbation using the projected loss gradient of the 3D object detector through multiple iterations with

$$\delta_n^{\mathrm{per}} = \mathrm{Clip}_\epsilon\{\alpha \times \mathrm{sign}(\nabla_{(I_l,I_r)} L(O_\theta(\tilde{I}_{l,n}^{\mathrm{per}}, \tilde{I}_{r,n}^{\mathrm{per}}), b^{\mathrm{true}}))\} \quad (1)$$

and

$$(\tilde{I}_{l,n+1}^{\mathrm{per}}, \tilde{I}_{r,n+1}^{\mathrm{per}}) = (\tilde{I}_{l,n}^{\mathrm{per}} + \delta_n^{\mathrm{per}}, \tilde{I}_{r,n}^{\mathrm{per}} + \delta_n^{\mathrm{per}}) \quad (2)$$

where $\mathrm{Clip}_\epsilon\{\cdot\}$ ensures that the value is within $[-\epsilon, \epsilon]$, α is the parameter that controls the attack intensity, $\mathrm{sign}(\cdot)$ denotes the sign function, $O_\theta(\cdot, \cdot)$ represents the vision-based 3D object detector parametrized by θ, $L(\cdot, \cdot)$ is the loss function of $O_\theta(\cdot, \cdot)$, b^{true} is the ground truth label paired with (I_l, I_r), and $0 \leqslant n \leqslant N - 1$. For convenience, we denote the perturbation attack as $(\tilde{I}_l^{\mathrm{per}}, \tilde{I}_r^{\mathrm{per}}) = A^{\mathrm{per}}(I_l, I_r, b^{\mathrm{true}}, \epsilon, \alpha, N)$.

Patch Attack. The patch attack is designed to model the real-world poster-printing attack in [7]. In the context of vision-based 3D object detection, a patch attack is launched to mislead the detector so that it detects ghost objects by including the patch in the view of the image. With prior knowledge of the deep learning model, attackers can train a patch offline, print it out, and put the physical patch inside the view of cameras to launch the attack. For example, the

attacker can place the patch at the roadside where the vision-based self-driving car passes by. Since a real-world 3D point appears at different positions in two stereo images, we consider a patch δ^{pat} that is pasted onto I_l at loc_l and onto I_r at loc_r, where $(loc_l, loc_r) \in \mathcal{L}$, $\mathcal{L}$ represents a set of random position pairs. Let $\lambda_{loc_l, loc_r} \in \Lambda$ be the disparity between loc_l and loc_r, where Λ denotes a set of valid disparities in the physical world. Let $\tau \in \mathcal{T}$ be a transformation that can be applied to δ^{pat}, where $\mathcal{T}$ includes rotations. Then, the patched image pair can be represented as $(\tilde{I}_l^{\mathrm{pat}}, \tilde{I}_r^{\mathrm{pat}}) = A^{\mathrm{pat}}(I_l, I_r, \delta^{\mathrm{pat}}, loc_l, loc_r, \tau)$. To implement this attack, the patch is optimized as

$$\underset{\delta^{\mathrm{pat}}}{\arg\min} \, \mathbb{E}_{(I_l,I_r)\sim\mathcal{I}, (loc_l,loc_r)\sim\mathcal{L}, \tau\sim\mathcal{T}} L(O_\theta(\tilde{I}_l^{\mathrm{pat}}, \tilde{I}_r^{\mathrm{pat}}), b^*), \quad (3)$$

where b^* denotes the predefined 3D bounding boxes used for misleading the object detector and serves as the optimization target here.

IV. END-TO-END DRIVING SAFETY EVALUATION FRAMEWORK

As discussed in Section II, many previous studies only showed that deep learning models of autonomous driving can be compromised by adversarial attacks, but they did not systematically assess the attack impact on driving safety. Our goal is to answer the questions raised in Section I by investigating the impact of perturbation attacks and patch attacks on driving safety of vision-based autonomous vehicles. This investigation considers not only the performance of the attacked deep learning models but also their impact on the overall safety, which is a combined effect of different functional modules involved in autonomous driving.

To this end, we design an end-to-end driving safety evaluation framework. *End-to-end* means that our system directly takes 3D object detection results as input and outputs the driving safety scores. Moreover, our evaluation framework adopts a modular design, so that each module can be easily replaced with other methods to assess the driving safety of different autonomous driving systems. Note that the existing simulators, such as Baidu Apollo [29] and CARLA [30], either have a low level of customization or are not compatible with real autonomous driving datasets. Therefore, we implement our own evaluation framework with real autonomous driving dataset to evaluate the impact of adversarial attacks on driving safety. In this section, we first introduce our evaluation framework model for vision-based autonomous driving and the driving safety metrics, then elaborate on the framework implementation details.

A. Framework Model

Our evaluation framework works along with the data flow of vision-based autonomous driving systems. In Figure 1, the black lines represent the data flow of our evaluation framework, while the red lines are for the data flow of the autonomous driving system. Usually inside the vision-based autonomous vehicle, a pair of stereo images $(I_l, I_r \in \mathbb{R}^{h \times w \times 3})$ is first fed as the input to the 3D object detection module $O_\theta(\cdot, \cdot)$, which is parameterized by θ and generates detected objects in 3D bounding boxes b (denoted as $b = O_\theta(I_l, I_r)$) as the output.
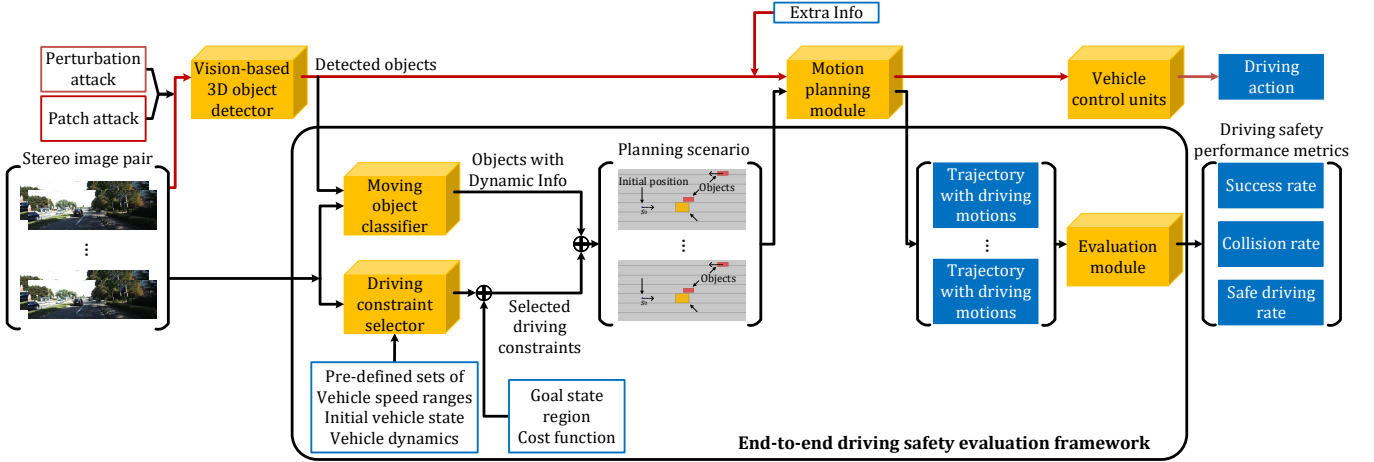
Fig. 1. The end-to-end driving safety evaluation framework.

Next, the bounding boxes b along with some extra information are passed to the the motion planning module $M(\cdot)$. At last, the vehicle control units execute the driving motion orders from the motion planning module. As depicted in Figure 1, our proposed end-to-end driving safety evaluation framework directly takes the detected objects of $O_\theta(\cdot, \cdot)$ as input, uses the same motion planning module of the autonomous vehicle, and outputs scores for driving safety metrics. This modular design makes it possible for our evaluation framework to be adopted by other self-driving systems which have different implementation of the aforementioned modules.

As described in Section I, two main technical challenges need to be addressed after the object detection results are fed into our evaluation framework. First, the 3D bounding boxes b as the object detection results only contain static information, i.e., object category, box dimensions, box center position in 3D space, and the confidence score. Base on the static information of one frame of data, we cannot distinguish between moving objects and static objects. However, the subsequent motion planning module requires dynamic information of objects as part of its input. Second, to realistically produce planned trajectory for the self-driving vehicle, driving constraints, including speed limits for different road types and dynamics constraints for different moving vehicles (acceleration, jerk, energy, etc.), must be considered to comply with the real driving scenarios. In addition, as the real driving scenarios can change dynamically, we must choose appropriate real driving constraints accordingly for driving safety evaluation.

To tackle the first challenge, we train a CNN-based moving object classifier $C(\cdot, \cdot, \cdot)$ to distinguish between dynamic and static objects by leveraging continuous frames of image data. We manually label each object with the ground truth indicating whether this object is moving or not. By doing this, we associate the object detection results with dynamic information. We denote this process as $\vec{b} = C(b, I_l, I_r)$.

To address the second challenge, we train another CNN-based model $S(\cdot, \cdot)$ with road type labels as the driving constraint selector, so that it can classify the road type of driving scenarios and select proper real driving constraints for

the evaluation. We denote this part as $(s_0, r, d) = S(I_l, I_r)$, where s_0 is the initial vehicle state, r is the allowed speed range, and d represents the vehicle dynamics. In this paper, we define a vehicle state $s := (p, v, \varphi, \omega)$ as a combination of position p, velocity v, orientation φ, and steering angle ω at a specific moment. Note that both the two aforementioned models, $C(\cdot, \cdot, \cdot)$ and $S(\cdot, \cdot)$, are trained on KITTI raw dataset [31].

Then, together with goal region g and cost function c, we combine the processed results of both the moving object classifier and the driving constraint selector to form a planning scenario. After that, the scenario is fed to the motion planning module $M(\cdot)$ that outputs a temporal sequence of planned vehicle states $\{s_t\}$ (a trajectory with planned driving motions), which is denoted as $\{s_t\} = M(\vec{b}, s_0, r, g, c, d)$, where $1 \leqslant t \leqslant T$.

The final assessment of driving safety is conducted by the evaluation module based on processing a large number of driving scenarios. Specifically, the evaluation module incorporates the planned trajectory into the planning scenario and detects collision for each driving scenario in the dataset. Then, it generates driving safety performance scores based on all detected collisions. Note that we refer to a collision as the physical contact of objects. In this paper, we evaluate driving safety on the KITTI object detection dataset [32].

Next, we introduce the driving safety performance metrics and present the details of the framework implementation.

B. Driving Safety Performance Metrics

To evaluate the driving safety of the vision-based autonomous driving system in a quantitative manner, we define a set of driving safety performance metrics as follows.

- *Successful planning rate.* In some scenarios, the motion planning module may not be able to generate a trajectory solution, which imposes a risk in driving safety. Thus, we define the successful planning rate as $m_{suc} = \frac{k_{trj}}{k_{dts}}$, where k_{dts} is the total number of scenarios in a dataset, and k_{trj} is the number of scenarios in that dataset where a trajectory can be successfully generated, no

(a) Clean image input.

(b) Ground truth of object detection.

(c) Detection results of DSGN without attack.

(d) Detection results of DSGN under attack.

(e) Detection results of Stereo R-CNN without attack.

(f) Detection results of Stereo R-CNN under attack.

(g) Clean image input.

(h) Ground truth of object detection.

(i) Detection results of DSGN without attack.

(j) Detection results of DSGN under attack.

(k) Detection results of Stereo R-CNN without attack.

(l) Detection results of Stereo R-CNN under attack.

Fig. 2. When there is no attack, both Stereo R-CNN and DSGN can detect objects accurately as shown in (c), (e), (i), and (k). When the perturbation attack is launched, the two models produce erroneous object detection results including inaccurate detection of real objects in (d), (j), and false detection of ghost objects in (f), (l).

matter whether it is collision-free or not. For the sake of simplicity, this metric is referred to as *the success rate*.

- *Collision rate*. We define the collision rate, m_{cls}, as the percentage of scenarios in all successfully planned trajectories where a collision occurs. Let $m_{cls} = \frac{k_{cls}}{k_{trj}}$, where k_{cls} is the number of scenarios with collision occurrence. Collision rate approximately reflects the collision probability under different levels of threats.
- *Safe driving rate*. The safe driving rate, m_{saf}, is defined as the percentage of scenarios in a dataset where a collision-free trajectory can be produced by the motion planning module. We denote it as $m_{saf} = \frac{k_{trj}-k_{cls}}{k_{dts}} = m_{suc} - \frac{k_{cls}}{k_{trj}}\frac{k_{trj}}{k_{dts}} = m_{suc} - m_{cls}m_{suc} = (1-m_{cls})m_{suc}$.

In this paper, we only focus on fatal driving risks when

referring to the driving safety. By measuring successful planning rate and collision rate, we capture the two most risky driving scenarios in autonomous driving, i.e., the failure of path planning and collision.

Note that successful planning rate and collision rate are also common performance metrics measuring the quality of motion planning. Furthermore, safe driving rate is jointly determined by both successful planning rate and collision rate, which is a more direct measure of driving safety.

C. Implementation

To implement this end-to-end driving safety evaluation framework for vision-based autonomous driving, we adopt two pre-trained models for the object detection module, namely,
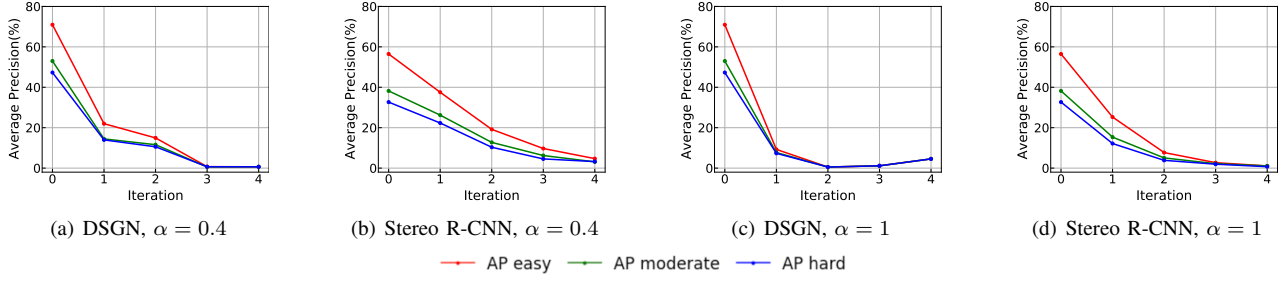
Fig. 3. Average precision for 3D object detection under the perturbation attack.

Changing to left lane

Keeping lane

Changing to right lane

(a) DSGN, $\alpha = 0.4$ (b) Stereo R-CNN, $\alpha = 0.4$ (c) DSGN, $\alpha = 1$ (d) Stereo R-CNN, $\alpha = 1$
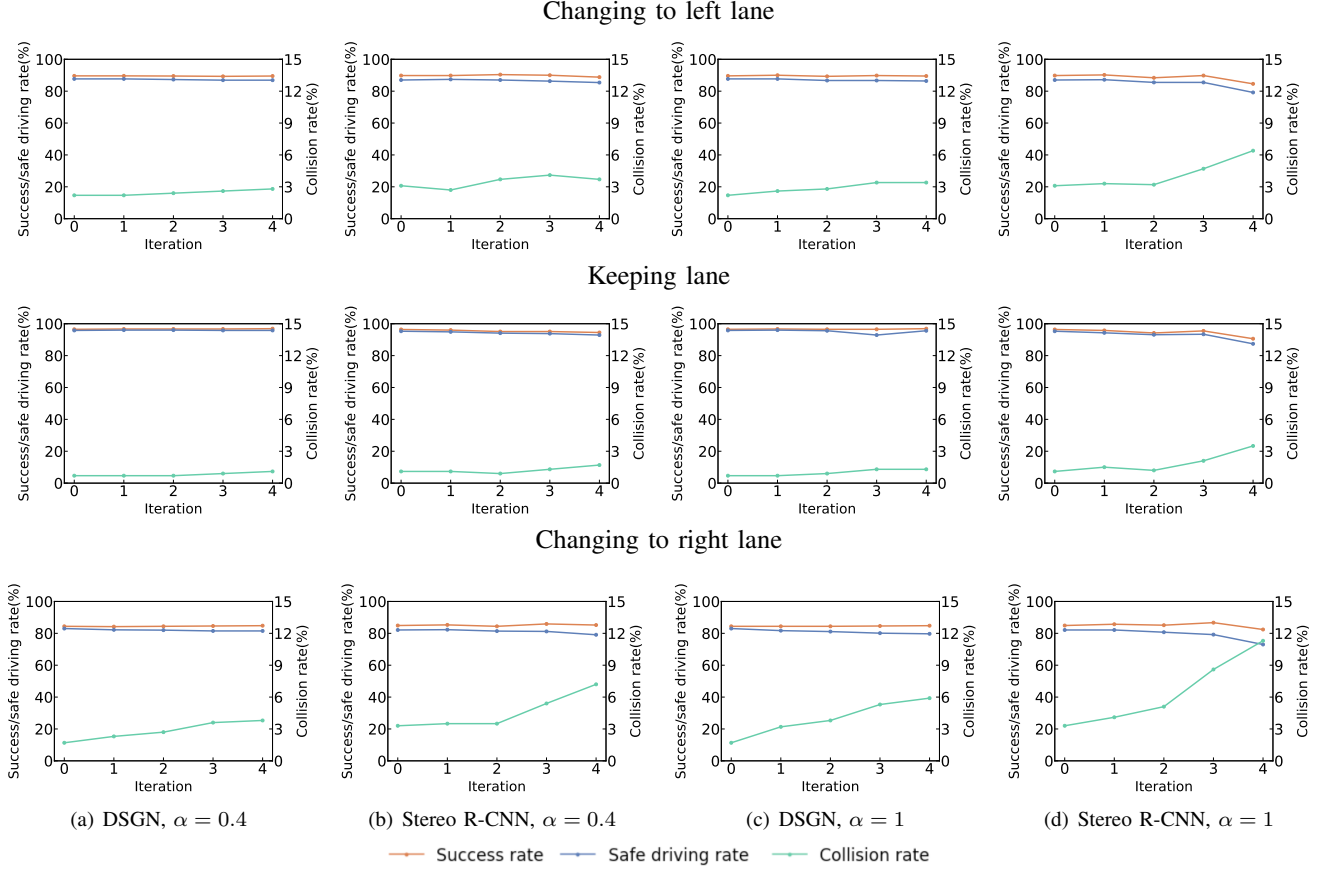
Fig. 4. Driving safety performance metrics under the perturbation attack.

Stereo R-CNN [16] and DSGN [17], which are currently two state-of-the-art methods in the area of vision-based 3D object detection. As for the motion planning module and the evaluation module, we use CommonRoad [33] as the framework and leverage the built-in A* [28] with sampled motion primitives as the motion planning method.

To implement the moving object classifier, we extract in total 600 real driving scenarios from the KITTI raw dataset [31] and manually label each object in each driving scenario with a moving/static property. We use 500 scenarios for training and 100 scenarios for validation. To determine whether an object is moving or not in a driving scenario, we refer to the previous and the subsequent image frames of that scenario. Though it is easy for human to judge the moving object from sequential frames, two people are assigned to manually label the scenarios independently in order to eliminate personal bias

or errors in manual labelling. The independently produced labels are checked together for consistency, and no inconsistent labelling is found. We adopt the 16-layer VGG net [34] as the core network of the moving object classifier and replace its fully connected layers, i.e., $fc6$, $fc7$, and $fc8$, with a flatten layer, a new fully connected layer with a dropout layer and ReLU activation function, and another new fully connected layer with a dropout layer and a sigmoid activation function, respectively, to make sure that there is only one output score to indicate the probability of a moving object. The validation results suggest that the accuracy of the trained moving object classifier is 98.31%.

To implement the driving constraint selector, we also leverage the KITTI raw dataset [31] to train the model so that it can classify the road type of a scenario. Specifically, we divide the dataset into two subsets, i.e., *street* and *highway*.

TABLE I

DRIVING SAFETY PERFORMANCE METRICS UNDER THE PERTURBATION ATTACK ($\alpha = 0.4$)

	Model	DSGN					Stereo R-CNN				
	Iteration	Unattacked	1	2	3	4	Unattacked	1	2	3	4
Success rate (%)	Left	89.6	89.6	89.5	89.3	89.5	89.8	89.8	90.4	90.0	88.8
	Straight	96.5	96.7	96.7	96.7	96.9	96.4	96.0	95.1	95.1	94.5
	Right	84.4	84.2	84.4	84.6	84.8	84.9	85.3	84.4	85.9	85.2
Collision rate (%)	Left	2.2	2.2	2.4	2.6	2.8	3.1	2.7	3.7	4.1	3.7
	Straight	0.7	0.7	0.7	0.9	1.1	1.1	1.1	0.9	1.3	1.7
	Right	1.7	2.3	2.7	3.6	3.8	3.3	3.5	3.5	5.4	7.2
Safe driving rate (%)	Left	87.7	87.7	87.3	86.9	86.9	87.0	87.4	87.0	86.3	85.4
	Straight	95.8	96.0	96.0	95.8	95.8	95.3	94.9	94.1	93.8	92.9
	Right	83.0	82.2	82.0	81.5	81.5	82.1	82.3	81.4	81.2	79.1

TABLE II

DRIVING SAFETY PERFORMANCE METRICS UNDER THE PERTURBATION ATTACK ($\alpha = 1$)

	Model	DSGN					Stereo R-CNN				
	Iteration	Unattacked	1	2	3	4	Unattacked	1	2	3	4
Success rate (%)	Left	89.6	90.0	89.3	89.8	89.5	89.8	90.2	88.4	89.8	84.6
	Straight	96.5	96.7	96.5	96.5	96.9	96.4	95.8	94.2	95.5	90.6
	Right	84.4	84.4	84.4	84.6	84.8	84.9	85.7	85.1	86.7	82.4
Collision rate (%)	Left	2.2	2.6	2.8	3.4	3.4	3.1	3.3	3.2	4.7	6.4
	Straight	0.7	0.7	0.9	1.3	1.3	1.1	1.5	1.2	2.1	3.5
	Right	1.7	3.2	3.8	5.3	5.9	3.3	4.1	5.1	8.6	11.3
Safe driving rate (%)	Left	87.7	87.7	86.7	86.7	86.4	87.0	87.2	85.5	85.5	79.2
	Straight	95.8	96.0	95.6	92.9	95.6	95.3	94.3	93.1	93.4	87.4
	Right	83.0	81.7	81.1	80.1	79.7	82.1	82.1	80.7	79.2	73.0

The street subset consists of city and residential scenarios where the traffic speed is relatively low, while the highway subset contains highway scenarios in which vehicles move much faster. Accordingly, we pre-define two sets of motion primitives for two road types so that the selector can pick the motion primitive with appropriate speed ranges and steering angle ranges after classifying the road type. The selector also chooses the dynamics constraints for moving vehicles predicted by the moving object classifier. The network architecture of the driving constraint selector consists of 5 convolution layers connected by max-pooling layers and 1 fully connected layer with dropout. Both convolution layers and the fully connected layer use ReLU as the activation function. After excluding the scenarios without cars, we select 444 scenarios as the training dataset and 112 scenarios as the validation dataset. The validation result indicates that the accuracy of the driving constraint selector achieves 94.64%.

V. EXPERIMENTS

we conduct extensive experiments to investigate the impact of perturbation attacks and patch attacks on driving safety of vision-based autonomous vehicles. We first introduce the common setup for all experiments, then elaborate on the specific settings for each attack experiment and present corresponding evaluation results. Finally, we summarize our findings at the end of this section.

A. Common Setup

In this paper, we conduct all experiments by applying the two types of adversarial attacks with different settings in our driving safety evaluation framework. Specifically, the evaluation framework includes two object detection modules, Stereo R-CNN [16] and DSGN [17]. In order to assess the impact

comprehensively, we gradually increase the attack intensity by changing the attack settings in fine-grained steps. For the same purpose, we also consider three driving intentions of the ego-vehicle for each scenario when planning the trajectory, namely, *changing to left lane*, *changing to right lane*, and *keeping lane*, which are abbreviated as *left*, *right*, and *straight*, respectively. For these three cases, the initial position of the ego-vehicle is the same and the goal region is located 15 meters away from the initial position but within three different adjacent lanes. Moreover, we randomly assign an initial speed within the selected speed range to each moving vehicle, including the ego-car. Specifically, the initial speed for moving vehicles in street scenarios is randomly assigned within the range of $[22, 29]$ km/h, considering the 30 km/h speed limit in German cities, campus and residential areas. The initial speed in highway scenarios is randomly assigned within the range of $[40, 47]$ km/h, concerning the 50 km/h speed limit of built-up roads in Germany. For each attack, after the framework processes all the scenarios and generates the motion planning results, it assesses the attack impact on the performance metrics of driving safety as well as on the accuracy of 3D object detector. By linking these two attack impacts together, we manage to obtain evaluation results that help answer the questions raised in Section I. The models of Stereo R-CNN [16] and DSGN [17] are pretrained with 3712 data points from the KITTI object detection dataset [32]. For each experiment setting, we test 600 real driving scenarios. The platform that we use is an Ubuntu 18.04 server equipped with an Nvidia Tesla V100 GPU.

In our experiments, the evaluation of driving safety is based on the trajectory produced by the motion planning module and measured by the driving safety performance metrics. In terms of evaluating the accuracy of the vision-based 3D object detector, we adopt the KITTI object detection benchmark that

(a) Clean image input.

(b) Ground truth of object detection.

(c) Detection results of DSGN without attack.

(d) Detection results of DSGN under attack.

(e) Detection results of Stereo R-CNN without attack.

(f) Detection results of Stereo R-CNN under attack.

(g) Clean image input.

(h) Ground truth of object detection.

(i) Detection results of DSGN without attack.

(j) Detection results of DSGN under attack.

(k) Detection results of Stereo R-CNN without attack.
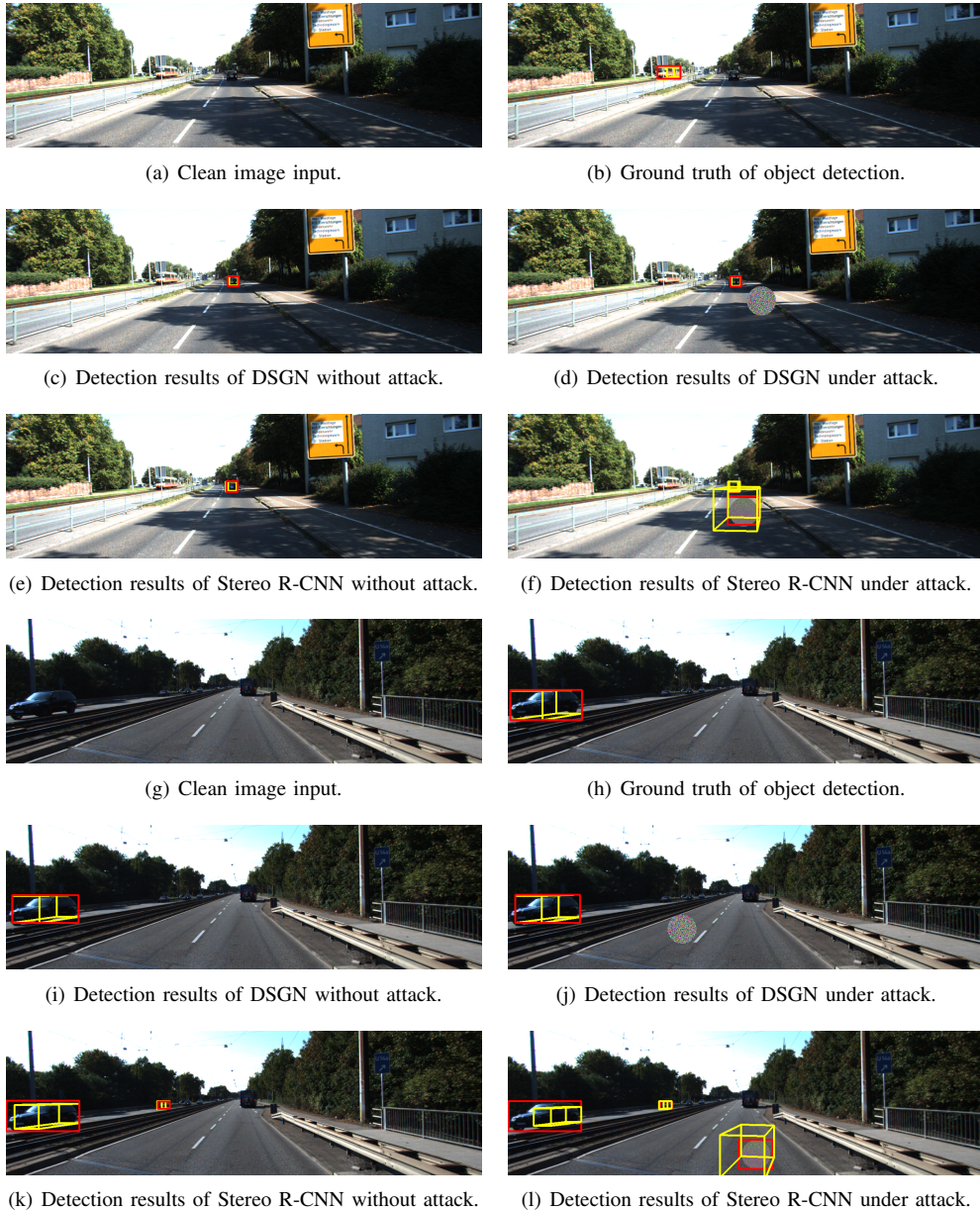
(l) Detection results of Stereo R-CNN under attack.

Fig. 5. The patch attack triggers 3D object detectors to generate ghost bounding boxes at in the area of the patch as shown in (f) and (l). It has little influence on the detection of the objects away from the patch.

tests the detector with a three-level standard, namely, *easy*, *moderate*, and *hard* [32]. We follow the standard to measure the *average precision* (AP) of the detector with *Intersection over Union* (IoU) larger than 70%.

B. Perturbation Attack

In order to perform the perturbation attack against autonomous driving systems at various intensities, we adjust two parameters α and n in Eqn. (1). To ensure that the perturbation is imperceivable to human eyes, their values usually should be set as small as possible. Specifically, we set the value of α as 0.4 and 1, to represent medium to high attack intensities, respectively. The number of iterations n changes from 1 to 4 accordingly, so that the modification on image pixel values

is constrained within the range of $[0.4, 4]$. We note that even the attack with the lowest attack intensity, i.e., $\alpha = 0.4$ and $n = 1$, can cause significant decline in the accuracy of 3D object detectors. Moreover, the produced perturbation and the input stereo images have the same dimension.

Evaluation. The effect of the perturbation attack in some driving scenarios is shown in Figure 2 where we can see that the attack causes inaccurate detection of real objects and false detection of ghost objects. We present the impact of the perturbation attack with different settings on average precision of 3D object detection and on driving safety metrics in Figure 3 and Figure 4, respectively. The numerical results of driving safety scores can be found in Table I and Table II. When the number of iterations n is 0, it indicates that there is no attack applied. From Figure 3, we can observe that

TABLE III
AVERAGE PRECISION FOR 3D OBJECT DETECTION UNDER PATCH ATTACK

	Model	DSGN					Stereo R-CNN				
	Scenario	Unattacked	Random Attack	Specific Attack			Unattacked	Random Attack	Specific Attack		
				Left	Straight	Right			Left	Straight	Right
3D	easy	70.94	63.85	65.72	64.96	64.87	56.47	53.17	48.14	47.82	50.07
Detection	moderate	52.98	48.20	51.47	50.77	51.63	38.20	37.07	36.27	35.23	38.02
AP (%)	hard	47.29	44.30	46.80	46.15	46.48	32.66	31.88	31.21	30.60	32.45

TABLE IV
DRIVING SAFETY PERFORMANCE METRICS UNDER THE PATCH ATTACK

	Model	DSGN			Stereo R-CNN		
	Scenario	Unattacked	Random Attack	Specific Attack	Unattacked	Random Attack	Specific Attack
	Left	89.6	90.0	90.0	89.8	70.3	83.7
Success rate(%)	Straight	96.5	96.5	96.7	96.4	81.2	57.5
	Right	84.4	84.6	84.8	84.9	68.7	74.4
	Left	2.2	2.6	2.8	3.1	1.9	2.1
Collision rate(%)	Straight	0.7	0.7	0.9	1.1	1.4	4.7
	Right	1.7	2.3	2.1	3.3	3.9	4.2
	Left	87.7	87.7	87.5	87.0	68.9	81.9
Safe driving rate(%)	Straight	95.8	95.8	95.8	95.3	80.1	54.7
	Right	83.0	82.6	83.0	82.1	65.9	71.2

TABLE V
DRIVING SAFETY PERFORMANCE METRICS OF STEREO R-CNN UNDER THE PATCH ATTACK WITH VARIOUS INTENTIONS

	Specific attack	Random attack		
	—	Same intentions[1]	Different intentions[2]	Unattacked
Success rate (%)	71.9	76.2	91.6	90.4
Collision rate (%)	3.5	2.3	1.5	2.4
Safe driving rate (%)	69.3	74.4	90.1	88.1

[1] "Same intentions" refers to cases where the attack intention and the driving intention are the same.
[2] "Different intentions" refers to cases where the attack intention differs from the driving intention.

TABLE VI
SAFE DRIVING RATE USING DIFFERENT PLANNING ALGORITHMS

	Planning algorithm	GBFS	A*
	Scenario	Ground Truth	
	Left	87.9	89.7
Safe driving rate (%)	Straight	98.0	98.0
	Right	82.3	85.2

with the enhanced attack intensity by increasing α and n, the average precision of both object detection models drops significantly, while the driving safety metrics only show very small changes. Take DSGN as an example. When α is 0.4 and n is increased from 0 (no attack) to 1, the AP declines by more than half for all three levels of the benchmark standard, i.e., from 70.94% to 21.99% for the category of AP easy, from 52.98% to 14.45% for the category of AP moderate, and from 47.29% to 13.96% for the category of AP hard. When n is 3, the AP of DSGN almost reaches 0. However, in the meantime, the driving safety performance metrics in Figure 4 barely change for all three intention cases (e.g., collision rate is only changed from 1.7% to 3.6% for the case of changing to right lane). When $\alpha = 1$ and n is increased from 0 (no attack) to 4, the AP of DSGN drops even more significantly, but the driving safety metrics only demonstrate slightly larger changes than that when $\alpha = 0.4$ (e.g., the safe driving rate drops by 0.8% when $\alpha = 0.4$ and by 1.3% when $\alpha = 1$ for the case of changing to left lane). The experiment results clearly indicate that the perturbation attack can dramatically

affect the performance of 3D object detection methods, but does not have much influence on the driving safety. In other words, a larger precision decline of the vision-based 3D object detectors under the perturbation attack does not indicate higher risk of driving safety.

Moreover, by comparing DSGN and Stereo R-CNN in terms of driving safety under perturbation attacks, we can observe that the changes in driving safety metrics for Stereo R-CNN tend to be larger than the changes for DSGN when both of them are tested in the same driving intention scenarios and at the same intensity. Therefore, Stereo R-CNN is more prone to perturbation attacks than DSGN in regard to driving safety.

C. Patch Attack

Different from perturbation attacks, the size of a patch in a patch attack is much smaller than the size of an input image. In our patch attack experiments, the radius of the patch is limited to 38 pixels. Here, the patch attack is launched as a white-box attack, which means that the patch is trained for Stereo R-CNN and DSGN, respectively. Specifically, we train the patch according to Eqn. 2 by placing the patch at a random position in stereo image pair and setting b^* in Eqn. 3 accordingly for each training scenario. To ensure that the patch for Stereo R-CNN and the patch for DSGN are equally optimized, we use the same learning step size and the same number of epochs when training patches.

We design two attack approaches, namely, *random attack* and *specific attack*. Random attacks are to place the trained

TABLE VII
SAFE DRIVING RATE WITH DIFFERENT INPUTS

	Model	—	DSGN			Stereo R-CNN		
	Scenario	Ground Truth	Unattacked	Perturbation Attack	Patch Attack	Unattacked	Perturbation Attack	Patch Attack
Safe driving rate (%)	Left	89.7	87.7	86.4	87.7	87.0	79.2	68.9
	Straight	98.0	95.8	95.6	95.8	95.3	87.4	80.1
	Right	85.2	83.0	79.7	82.6	82.1	73.0	65.9

patch at a random position within the entire image no matter which driving intention case it is. In other words, the attack intention may or may not be consistent with the driving intention in random attacks. Specific attacks are also to place the patch randomly but within a certain region of the image, depending on the driving intention case, e.g., if the driving intention is changing to right lane, then the patch is placed in the right part of the image. In other words, the attack intention is always consistent with the driving intention in specific attacks. By designing two attack approaches, we can create different attack intensities for patch attacks. Specifically, the attack intensity of random attacks is lower than that of specific attacks.

Evaluation. The performance of patch attacks in some driving scenarios is shown in Figure 5 in which we can observe that patch attacks cause false detection of ghost objects. The impact of patch attacks on object detection and driving safety are shown in Table III and Table IV respectively. From the tables, we can observe that, when different attack approaches are applied, the average precision of both object detection models declines slightly, while some of the driving safety metrics degrade significantly. For example, when random patch attacks are applied to the Stereo R-CNN model, AP declines slightly for all three levels of the benchmark standard, i.e., from 56.47% to 53.17% for the case of AP easy, from 38.20% to 37.07% for the case of AP moderate, and 32.66% from to 31.88% for the case of AP hard. However, the driving safety performance metrics of Stereo R-CNN have a relatively larger drop under random patch attacks (e.g., safe driving rate drops from 95.3% to 80.1% for the case of keeping lane). At the same time, for specific patch attacks, Stereo R-CNN shows the similar average precision decline which is only within the range of $[0.21\%, 8.65\%]$, while significant driving safety performance degradation can be observed (e.g., the safe driving rate decreases to half for the case of keeping lane). The experiment results suggest that a slight precision decline of the 3D object detectors under patch attacks does not indicate mild risk of driving safety.

Since the driving safety performance of Stereo R-CNN can be significantly affected by patch attacks, we further investigate the performance under the attacks where the attack intention is the same as the driving intention, and the attacks where the attack intention is different from the driving intention. The results are listed in Table V. From Table V, we can see that the driving safety performance under the attacks where the driving intention and the attack intention are different is very similar to that in unattacked scenarios, and the performance under the attacks where the attack intention is the same as the driving intention is very close to that in

specific attack scenarios.

Furthermore, the DSGN model again shows its much better robustness in object detection and driving safety under patch attacks. We can observe that, even under the well-designed specific patch attacks, DSGN's average precision decline is only less than 6%, and the driving safety performance metrics almost remain unchanged, while Stereo R-CNN performs worse under both random patch attacks and specific patch attacks.

D. Attack Impact Demonstration

To demonstrate that the performance of different models under different attacks is mainly caused by adversarial attacks, not by the motion planning algorithms, we conduct two experiments.

We first evaluate the performance of the motion planning module using different inputs and then calculating the safe driving rates in different scenarios. Specifically, we first use the ground truth data of 3D object detection as the inputs to evaluate two popular motion planning algorithms, A* and Greedy-BFS, so as to show the impact of the motion planning module on the driving safety. The experimental results are summarized in Table VI. From Table VI, we can first observe that, when ground truth data are used, the A* planning algorithm can achieve the safe driving rates $89.7\%, 98.0\%$, and 85.2% for the three driving intention scenarios, respectively, while the Greedy-BFS algorithm can achieve the safe driving rates $87.9\%, 98.0\%$, and 82.3% for the three driving intentions scenarios, respectively. The performance of A* and the performance of the Greedy-BFS are very close. In other words, the performance variance demonstrated by DSGN and Stereo R-CNN under adversarial attacks is irrelevant to the selection of the motion planning algorithm. Since the performance of A* is slightly better than that of Greedy-BFS, we select A* as the motion planning algorithm for our driving safety evaluation framework.

We then use detection data without attacks (i.e., unattacked) and detection data under two types of attacks to demonstrate the impact of detection module and adversarial attacks on the driving safety. The results are shown in Table VII. From Table VII, we can observe that the safe driving rates produced by the detection data without attacks are slightly smaller than the safe driving rates when the ground truth data are used as inputs. This slight difference is caused by the accuracy of the two models. Finally, compared with the safe driving rates when the inputs are unattacked detection data, the safe driving rates under adversarial attacks are significantly declined in all driving intention scenarios. Since all experiments are conducted using the same motion planning algorithm, we can
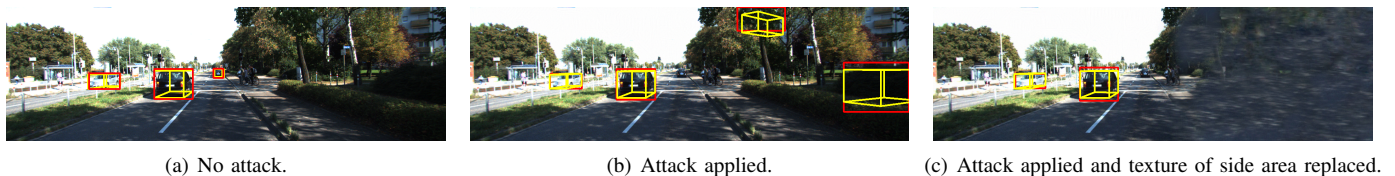
(a) No attack.	(b) Attack applied.	(c) Attack applied and texture of side area replaced.

Fig. 6. Results of the texture replacement experiment for Stereo R-CNN.

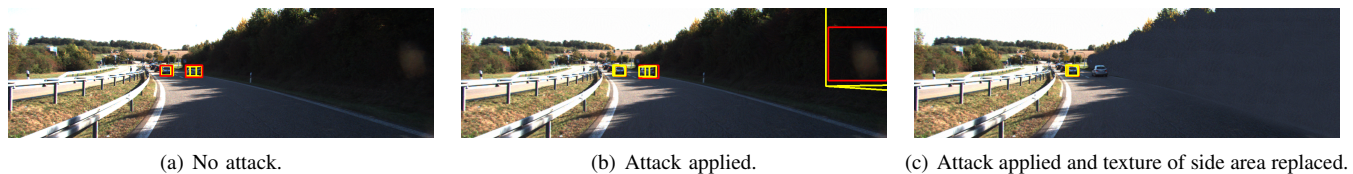|		
(a) No attack.	(b) Attack applied.	(c) Attack applied and texture of side area replaced.

Fig. 7. Results of the texture replacement experiment for DSGN.

conclude that the declines in the driving safety performance metrics are primarily caused by adversarial attacks.

E. Findings

To briefly summarize, the findings from the experiments of perturbation attacks and patch attacks are listed as follows:

- A larger precision decline of the attacked vision-based object detectors does not necessarily indicate a higher risk of driving safety. Similarly, a slight precision decline of the vision-based object detectors under attacks does not necessarily indicate a small risk of driving safety, either. Hence, the precision decline or the erroneous rate increase of the vision-based object detectors under attacks cannot represent their robustness with respect to driving safety of autonomous vehicles.
- Stereo R-CNN is less robust than DSGN in terms of driving safety and detection accuracy when the attacks launched on them are at the same intensity level. Hence, DSGN is a better selection of the vision-based 3D object detection for its stronger robustness and higher detection precision.

In terms of these two findings, we further design more experiments to explain the real causes behind them in the next section.

VI. ABLATION STUDY

In this section, we first investigate the reason of the decoupling between the precision of 3D object detectors and the driving safety performance metrics under adversarial attacks. Second, we investigate the reason why the DSGN model is more robust than the Stereo R-CNN model.

A. The Cause of the Decoupling between the Precision of 3D Object Detectors and Driving Safety under Adversarial Attacks

From the experiments in Section V, we observe that perturbation attacks cause a significant precision decline in 3D object detection, but only a slight change in driving safety performance. Patch attacks cause a slight decline in 3D object detection precision, but a relatively significant performance degradation in driving safety. To figure out the reasons beneath these results, we take a closer look at the detection results of the attacked and unattacked 3D object detectors and compare them accordingly.

The reasons for the decoupling caused by patch attacks are relatively straightforward. First, we notice that the affected area of patch attacks is limited to the patch itself where the patch is usually quite small in order to make it difficult to be detected. Thus, patch attacks can only trigger the object detectors to produce a very small number of ghost 3D bounding boxes inside the patch. This is the reason why the object detection precision does not show a significant decline when detectors are under patch attacks. Second, since the adversarial patch is randomly placed in driving scenarios, the resulted ghost 3D bounding boxes have a fair chance to appear on the road surface and block the way of the ego-vehicle, which directly leads to noticeably driving safety performance degradation. These two reasons together explain the decoupling between the precision of 3D object detectors and driving safety under patch attacks. In the rest of this section, we mainly focus on investigating the reasons for the decoupling caused by perturbation attacks.

Apart from the fact that the perturbation attacks cause slight drifts of 3D bounding boxes of real objects which are originally produced accurately when no attack is launched, the most significant consequence of a perturbation attack is that it triggers the object detectors to produce a lot of ghost 3D bounding boxes which do not circle any specific or meaningful object inside. In particular, almost all ghost boxes appear in the side areas of a road instead of on the surface of a road. Since the optimal trajectory generated by the motion planner most likely will not traverse the side areas of a road, the ghost objects will not affect the trajectory generated by the motion planner. In other words, the trajectories generated before and after perturbation attacks are essentially the same. Thus, driving safety is not affected dramatically by permutation attacks.

We further investigate why the ghost 3D bounding boxes caused by perturbation attacks tend to appear in the side
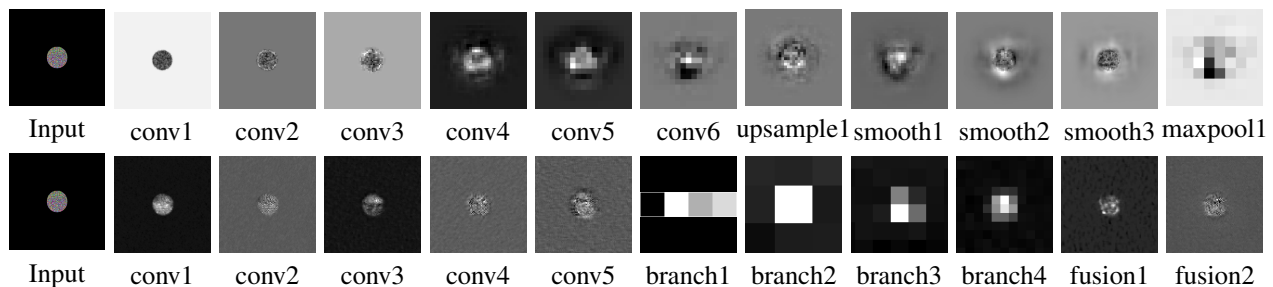
Fig. 8. Visualization of the feature maps of Stereo R-CNN (the 1st row) and DSGN (the 2nd row) from shallow to deeper layers.

areas of a road. After inspecting the positions where ghost bounding boxes appear in a large number of driving scenarios, we hypothesize that the difference in the texture complexity between the side areas of a road and the road surface may be the cause of this. The reason is that the texture of the road surface is more regular than the texture of the side areas of a road. Thus, it takes more "efforts" for perturbation attacks to change the pixel values to generate ghost boxes on the road surface than that in the side areas of a road.

In order to validate our hypothesis, we design a texture replacement experiment. Specifically, for a driving scenario in which vision-based 3D object detectors under a perturbation attack produce ghost 3D bounding boxes in the side areas of a road, we replace the texture of the side areas with the texture of the road surface, then feed this modified driving scenario to the attacked object detectors and check the detection results. If our hypothesis is correct, we shall expect that the attacked object detectors do not produce any ghost boxes in the side areas of a road for the modified driving scenario.

The results of the texture replacement experiments are shown in Figure 6 for the Stereo R-CNN model and in Figure 7 for the DSGN model. The attack setting of the perturbation attack applied here is set to be $\alpha = 1$ and $n = 4$ in Eqn. (1). From the figures, we can observe that both 3D object detectors can detect the object accurately when there is no perturbation attack applied (Figure 6(a), 7(a)), and ghost 3D bounding boxes appear in the side areas of a road when the perturbation attack is launched on the same driving scenario (Figure 6(b), 7(b)). More importantly, after we replace the texture of side area of road with the texture of road surface (Figure 6(c), 7(c)), no more ghost boxes are produced in the side area of the road by the 3D object detectors, which matches our expectation. Hence, the texture replacement experiment results validate our hypothesis that the difference in the texture complexity between the side areas of a road and the road surface leads to the decoupling between the precision of 3D object detectors and the driving safety performance metrics when the 3D object detectors are attacked.

B. The Cause of Difference in Robustness

The experiment results in Section V indicate that DSGN is more robust than Stereo R-CNN in terms of driving safety and object detection when they are under adversarial attacks. Especially, when patch attacks are launched, DSGN is more robust than Stereo R-CNN.

To better understand the cause of such a difference in robustness, we conduct a contrast experiment by implementing the black-box patch attack where instead of training a patch for each model separately, we learn a patch p that is jointly optimized for both the DSGN model and the Stereo R-CNN model using Eqn. (2). Thus, the patch is capable of attacking both models. To conduct this experiment, we also generate an image I with uniformly distributed random noise and paste the patch p on I to form the attacked input image $\tilde{I}$. We then feed the two input images into the models to observe the intermediate results produced by their network architectures. In Figure 8, we visualize the difference between corresponding intermediate feature maps generated from I and $\tilde{I}$ for both models respectively. In other words, the feature map F_k in Figure 8 refers to the average norm of the difference between the k-th layer output with an attack applied and the k-th layer output without any attack. For each model, we inspect the intermediate feature map of layers from shallow to deep in the feature extraction part of its network architecture. Each feature map is cropped so that only the central part is used for the propose of demonstration.

It is the spatial propagation of the patch activation area in feature maps of a model that implies the robustness of the model to patch attacks. Specifically, if the patch activation area in feature maps propagates along the data flow direction of the network architecture, then the network architecture amplifies the impact of patch attacks on the model, suggesting weak robustness of the model to patch attacks. In contrast, if the patch activation area in feature maps does not propagate or even contracts along the data flow, then the network architecture of the model is more resilient to patch attacks, indicating stronger robustness of the model to adversarial patches.

In the first row of Figure 8, we show how the patch activation area propagates layer by layer in the Stereo R-CNN model. In the first few convolution layers ($conv<1$, 2, $3>$), the patch activation area is bounded by the original region. However, starting from the last two layers of the feature extractor ($conv<4$, $5>$), we observe that the activation area gradually propagates as we move on to deeper layers. After the $maxpool1$ layer, the patch activation area propagates to almost the entire cropped image. Since the patch activation area keeps propagating through the network architecture, Stereo R-CNN shows poor robustness under patch attacks.

In the second row of Figure 8, the DSGN model shows less propagation of the patch activation in the first three convolution layers ($conv<1$, 2, $3>$), but the activation area

at the last two layers of the feature extractor ($conv$<4, 5>) are expanded slightly. According to the network structure of DSGN, the feature extractor is connected to the Spatial Pyramid Pooling (SPP) module, and the outputs of SPP branches ($branch$<1, 2, 3, 4>) are fused with features from the former layers for future prediction. Interestingly, we observe that the patch activation area shrinks to its original size after the SPP module ($fusion$<1, 2>). Hence, different from Stereo R-CNN, the SPP module in the DSGN model restrains the propagation of the patch impact. This demonstrates that DSGN has strong robustness to adversarial patches due to the SPP module in its network architecture. A similar observation of the Spatial Pyramid structure can be found in another study [6]. We can conclude that when the adversarial patch is used to attack the model of Stereo R-CNN whose network architecture is not equipped with the SPP module, the patch exploits the weakness of the network architecture and amplifies its impact on 3D object detection. For the DSGN model, the SPP module in its network architecture restrains the impact of the adversarial patch on 3D object detection. Therefore, DSGN and Stereo R-CNN have different robustness to patch attacks and demonstrate different performance on the average precision of 3D object detection and the driving safety metrics.

VII. Conclusion

In this paper, we have systematically investigated the impact of adversarial attacks not only on the object detection precision, but also on the driving safety of vision-based autonomous vehicles. Specifically, we proposed an end-to-end driving safety evaluation framework with designed performance metrics for the assessment of driving safety. Through extensive evaluation experiments, we found that a significant precision decline of 3D object detectors under the perturbation attack only leads to a slight decline in the driving safety performance metrics, but a mild precision decline of 3D object detectors under the patch attack can result in a significant performance degradation in driving safety. This finding suggests that it is desirable to evaluate the robustness of deep learning models in terms of driving safety rather than model precision. The proposed work can help guide the selection of deep learning models. The code of our evaluation framework is available upon request.

In the future, based on our experiments and discovered causes, we plan to expand our study to the autonomous vehicles that fuse the information from both LiDARs and cameras, and consider other types of attacks, such as attacks against LiDARs. Furthermore, we plan to design countermeasures for deep learning models against adversarial attacks by leveraging the spatial pyramid structure. Our future studies will also be conducted in an end-to-end fashion.

References

[1] A. Davies, "Waymo's So-Called Robo-Taxi Launch Reveals a Brutal Truth," 2018. [Online]. Available: https://www.wired.com/story/waymo-self-driving-taxi-service-launch-chandler-arizona/

[2] A. Hawkins, "Tesla's 'Full Self-Driving' feature may get early-access release by the end of 2019," 2019. [Online]. Available: https://www.theverge.com/2019/10/23/20929529/tesla-full-self-driving-release-2019-beta

[3] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, "The Security of Autonomous Driving: Threats, Defenses, and Future Directions," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 357–372, Feb. 2019.

[4] A. M. Wyglinski, X. Huang, T. Padir, L. Lai, T. R. Eisenbarth, and K. Venkatasubramanian, "Security of Autonomous Systems Employing Embedded Computing and Sensors," *IEEE Micro*, vol. 33, no. 1, pp. 80–86, Jan. 2013.

[5] H. Zhou, W. Li, Y. Zhu, Y. Zhang, B. Yu, L. Zhang, and C. Liu, "Deepbillboard: Systematic Physical-World Testing of Autonomous Driving Systems," *arXiv preprint arXiv:1812.10812*, 2018.

[6] A. Ranjan, J. Janai, A. Geiger, and M. J. Black, "Attacking Optical Flow," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 2404–2413.

[7] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 1625–1634.

[8] J. Lu, H. Sibai, and E. Fabry, "Adversarial Examples that Fool Detectors," *arXiv preprint arXiv:1712.02494*, 2017.

[9] J. Hendrik Metzen, M. Chaithanya Kumar, T. Brox, and V. Fischer, "Universal Adversarial Perturbations Against Semantic Image Segmentation," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2774–2783.

[10] 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)*. USENIX Association, Aug. 2018.

[11] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, "ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, 2019, pp. 52–68.

[12] Z. Zhang, X. Zhu, Y. Li, X. Chen, and Y. Guo, "Adversarial Attacks on Monocular Depth Estimation," *arXiv preprint arXiv:2003.10315*, 2020.

[13] A. Mathew, A. P. Patra, and J. Mathew, "Monocular Depth Estimators: Vulnerabilities and Attacks," *arXiv preprint arXiv:2005.14302*, 2020.

[14] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial Sensor Attack on Lidar-Based Perception in Autonomous Driving," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. ACM, Nov. 2019, pp. 2267–2281.

[15] M. Slovick, "Security Issues Could Still Crimp the Self-Driving Car," *Electronic Design, June*, vol. 28, 2017.

[16] P. Li, X. Chen, and S. Shen, "Stereo R-CNN Based 3D Object Detection for Autonomous Driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 7636–7644.

[17] Y. Chen, S. Liu, X. Shen, and J. Jia, "DSGN: Deep Stereo Geometry Network for 3D Object Detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 12 533–12 542.

[18] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing Properties of Neural Networks," in *2nd International Conference on Learning Representations, ICLR*, Apr. 2014.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *3rd International Conference on Learning Representations, ICLR*, May 2015.

[21] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Examples in the Physical World," in *5th International Conference on Learning Representations, ICLR*, Apr. 2017.

[22] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting Adversarial Attacks with Momentum," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 9185–9193.

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *6th International Conference on Learning Representations, ICLR*, Apr. 2018.

[24] C. Xiao, R. Deng, B. Li, F. Yu, M. Liu, and D. Song, "Characterizing Adversarial Examples Based on Spatial Consistency Information for Semantic Segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 217–234.

[25] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3D Object Proposals Using Stereo Imagery for Accurate Object Class Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1259–1272, 2018.

[26] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 8437–8445.

[27] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Iowa State University, Tech. Rep., 1998.

[28] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[29] Apollo Open Platform, "Apollo 5.0 Perception Module," 2019. [Online]. Available: http://apollo.auto/platform/perception.html

[30] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Conference on Robot Learning*. PMLR, 2017, pp. 1–16.

[31] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision Meets Robotics: The KITTI Dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[32] 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, Jun. 2012, pp. 3354–3361.

[33] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable Benchmarks for Motion Planning on Roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 719–726.

[34] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.