2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Adv3D: Generating 3D Adversarial Examples for 3D Object Detection in Driving Scenarios with NeRF

Leheng Li[1], Qing Lian[2], Ying-Cong Chen[1,2,*]

***Abstract*— Deep neural networks (DNNs) have been proven extremely susceptible to adversarial examples, which raises special safety-critical concerns for DNN-based autonomous driving stacks (*i.e.*, 3D object detection). Although there are extensive works on image-level attacks, most are restricted to 2D pixel spaces, and such attacks are not always physically realistic in our 3D world. Here we present Adv3D, the first exploration of modeling adversarial examples as Neural Radiance Fields (NeRFs) in driving scenarios. Advances in NeRF provide photorealistic appearances and 3D accurate generation, yielding a more realistic and realizable adversarial example. We train our adversarial NeRF by minimizing the surrounding objects' confidence predicted by 3D detectors on the training set. Then we evaluate Adv3D on the unseen validation set and show that it can cause a large performance reduction when rendering NeRF in any sampled pose. To enhance physical effectiveness, we propose primitive-aware sampling and semantic-guided regularization that enable 3D patch attacks with camouflage adversarial texture. Experimental results demonstrate that our method surpasses the mesh baseline and generalizes well to different poses, scenes, and 3D detectors. Finally, we provide a defense method to our attacks that improves both the robustness and clean performance of 3D detectors.**

I. Introduction

The perception system of self-driving cars heavily rely on DNNs to process input data and comprehend the environment. Although DNNs have exhibited great improvements in performance, they have been found vulnerable to adversarial examples [1]–[4]. These adversarial examples crafted by adding imperceptible perturbations to input data, can lead DNNs to make wrong predictions. Motivated by the safety-critical nature of self-driving cars, we aim to explore the possibility of generating physically effective adversarial examples to disrupt 3D detectors in driving scenarios, and further improve the robustness of 3D detectors through adversarial training.

The 2D pixel perturbations (digital attacks) [1], [2] have been proven effective in attacking DNNs in various computer vision tasks [5]–[7]. However, these 2D pixel attacks are restricted to digital space and are difficult to realize in our 3D world. To address this challenge, several works have proposed physical attacks. For example, [4] propose the framework of Expectation Over Transformation (EOT) to improve the attack robustness over 3D transformation. Other researchers generate adversarial examples beyond image space through differentiable rendering, as seen in [8], [9]. These methods show great promise for advancing the field of 3D adversarial attacks and defense but are still limited in synthetic environments.

Given the safety-critical demand for self-driving cars, several works have proposed physically realizable attacks and defense methods in driving scenarios. For example, [10], [11] propose to learn 3D adversarial attacks capable of generating adversarial mesh to attack 3D detectors. However, their works only consider learning a 3D adversarial example for a few specific frames. Thus, the learned example is not universal and may not transfer to other scenes. To mitigate this problem, [12], [13] propose to learn a transferable adversary that is placed on top of a vehicle. Such an adversary can be used in any scene to hide the attacked object from 3D detectors. However, reproducing their attack in our physical world can be challenging since their adversary must have direct contact with the attacked object. We list detailed comparisons of prior works in Tab. I.

To address the above challenges and generate 3D adversarial examples in driving scenarios, we build Adv3D upon recent advances in NeRF [14] that provide both differentiable rendering and realistic synthesis. In order to generate physically effective attacks, we model Adv3D in a patch-attack [15] manner and use an optimization-based approach that starts with a realistic NeRF object [16] to learn its 3D adversarial texture. We optimize the adversarial texture to minimize the predicted confidence of all objects in the scenes, while keeping shape unchanged. During the evaluation, we render the input agnostic NeRF in randomly sampled poses, then we paste the rendered patch onto the unseen validation set to evaluate the attack performance. Owing to the transferability to poses and scenes, our adversarial examples can be executed without prior knowledge of the scene and do not need direct contact with the attacked objects, thus making for more feasible attacks compared with [12], [13], [17], [18]. Finally, we provide thorough evaluations of Adv3D on camera-based 3D object detection with the nuScenes [19] dataset. Our contributions are summarized as follows:

- We introduce **Adv3D**, the first exploration of formulating adversarial examples as NeRF to attack 3D detectors in autonomous driving. Adv3D provides photorealistic synthesis and demonstrates effective attacks on various detectors.
- Incorporating the proposed primitive-aware sampling and semantic-guided regularization, Adv3D generates adversarial examples with enhanced physical realism and effectiveness.

*Corresponding author.
[1] Artificial Intelligence Thrust, The Hong Kong University of Science and Technology (Guangzhou). lli181@connect.hkust-gz.edu.cn, yingcongchen@hkust-gz.edu.cn
[2] Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. qlianab@connect.ust.hk

Methods	Transferability	Adv. Type	Requirements
Cao *et al.* [10], [11]	Poses	3D Mesh	Model, Annotation
Tu *et al.* [12], [13]	Poses, Scenes	3D Mesh	Model, Annotation
Xie *et al.* [18]	Scenes, Categories	2D Patch	Model, Annotation
Ours	Poses, Scenes, Categories	3D NeRF	Model

TABLE I: Comparison with prior works of adversarial attack in autonomous driving.

- We conduct extensive real-world experiments to validate the transferability of our adversarial examples across unseen environments and detectors. Additionally, the analysis of these experiments provides valuable insights for developing more robust detectors.
- We show that by employing adversarial training with a trained adversarial NeRF, we can enhance the robustness and clean performance of 3D detectors.

II. Related Work

A. Adversarial Attack

DNNs are known to be vulnerable to adversarial attacks. [1] first discovered that adversarial examples, generated by adding visually imperceptible perturbations to the original images, make DNNs predict a wrong category with high confidence. These vulnerabilities were also discovered in object detection and semantic segmentation [5], [20]. Moreover, DPatch [20] proposes transferable patch-based attacks by compositing a small patch to the input image. However, perturbing image pixels alone does not guarantee that adversarial examples can be created in the physical world. To address this issue, several works have performed physical attacks [4], [21]–[26] and exposed real-world threats. For example, AdvPC [27] investigates adversarial perturbations on 3D point clouds. SADA [28] proposes semantic adversarial diagnostic attacks in various autonomous applications. ViewFool [29] and VIAT [30] evaluate the robustness of DNNs to adversarial viewpoints by using NeRF's differentiability. In our work, we mainly aim to generate 3D adversarial examples for camera-based 3D object detection in driving scenarios.

B. Robustness in Autonomous Driving

With the safety-critical nature, it is necessary to pay special attention to robustness in autonomous driving. LiDAR-Adv [10] proposes to learn input-specific adversarial point clouds to fool LiDAR detectors. [12] produces generalizable point clouds that can be placed on a vehicle roof to hide it. Adv3D [31] and SHIFT3D [32] generate adversarial 3D shapes to fool full autonomy stack and Lidar detector, respectively. Furthermore, several work [11], [13], [33] try to attack a multi-sensor fusion system by optimizing 3D mesh through differentiable rendering. We compare our method with prior works in Tab. I. Our method demonstrates stronger transferability and fewer requirements than prior works.

C. Image Synthesis using NeRF

NeRF [14] enables photorealistic synthesis in a 3D-aware manner. Recent advances [34] in NeRF allow for control over materials, illumination, and 6D pose of objects. Additionally, NeRF's rendering comes directly from real-world reconstruction, providing more physically accurate and photorealistic synthesis than previous mesh-based methods that relied on human handicrafts. Moreover, volumetric rendering [35] enables NeRF to perform accurate and efficient gradient computation compared with dedicated renderers in mesh-based differentiable rendering [36], [37].

III. Preliminary

A. Camera-based 3D Object Detection

Camera-based 3D object detection is the fundamental task in autonomous driving. Without loss of generality, we focus on evaluating the robustness of camera-based 3D detectors.

The 3D detectors process image data and aim to predict 3D bounding boxes of all surrounding objects. The parameterization of a 3D bounding box can be written as $\mathbf{b} = \{\mathbf{R}, \mathbf{t}, \mathbf{s}, c\}$, where $\mathbf{R} \in SO(3)$ is the rotation of the box, $\mathbf{t} = (x, y, z)$ indicate translation of the box center, $\mathbf{s} = (l, w, h)$ represent the size (length, width, and height) of the box, and c is the confidence of the predicted box.

The network structure of camera-based 3D object detectors can be roughly categorized into FoV-based (front of view) and BEV-based (bird's eye view). FoV-based methods [38]–[40] can be easily built by adding 3D attribute branches to 2D detectors. BEV-based methods [41], [42] typically convert 2D image feature to BEV feature using camera parameters, then directly detect objects on BEV planes. We refer readers to recent surveys [43] for more detail.

B. Differentiable Rendering using NeRF

Our method leverages the differentiable rendering scheme proposed by NeRF. NeRF parameterizes the volumetric density and color as a function of input coordinates. NeRF uses multi-layer perceptron (MLP) or hybrid neural representations [44], [45] to represent this function. For each pixel on an image, a ray $\mathbf{r}(t) = \mathbf{r}_o + \mathbf{r}_d \cdot t$ is cast from the camera's origin $\mathbf{r}_o$ and passes through the direction of the pixel $\mathbf{r}_d$ at distance t. In a ray, we uniformly sample K points from the near plane t_{near} to the far plane t_{far}, the k^{th} distance is thus calculated as $t_k = t_{near} + (t_{far} - t_{near}) \cdot k/K$. For any queried point $\mathbf{r}(t_k)$ on the ray, the network takes its position $\mathbf{r}(t_k)$ and predicts the per-point color $\mathbf{c}_k$ and density τ_k with:

$$(\mathbf{c}_k, \tau_k) = \text{Network}(\mathbf{r}(t_k)). \tag{1}$$

Note that we omit the direction term as suggested by [46]. The final predicted color of each pixel $\mathbf{C}(\mathbf{r})$ is computed by approximating the volume rendering integral using numerical quadrature [47]:

$$\begin{aligned} \mathbf{C}(\mathbf{r}) &= \sum_{k=0}^{K-1} T_k \left(1 - \exp\left(-\tau_k \left(t_{k+1} - t_k\right)\right)\right) \mathbf{c}_k, \\ \text{with} \quad T_k &= \exp\left(-\sum_{k'<k} \tau_{k'} \left(t_{k'+1} - t_{k'}\right)\right). \end{aligned} \tag{2}$$

We build our NeRF upon Lift3D [16]. Lift3D is a 3D generation framework that generates photorealistic objects by fitting multi-view images synthesized by 2D generative

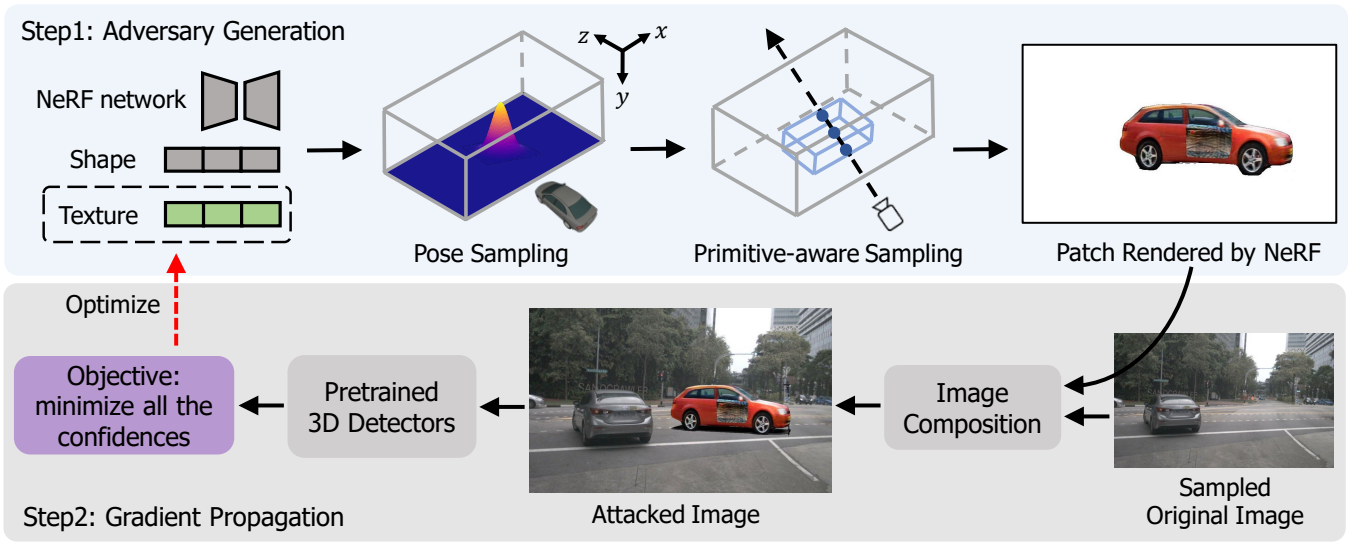


Fig. 1: **Adv3D** aims to generate 3D adversarial examples that consistently perform attacks under different poses during rendering. We initialize adversarial examples from Lift3D [16]. During training, we optimize the texture latent codes of NeRF to minimize the detection confidence of all surrounding objects. During inference, we evaluate the performance reduction of pasting the adversarial patch rendered using randomly sampled poses on the validation set.

modes [48] using NeRF. The network of Lift3D is a conditional NeRF with additional latent code input, which controls the shape and texture of the rendered object. The conditional NeRF in Lift3D is a tri-plane [49] generator. With its realistic generation and 3D controllability, Lift3D has demonstrated that the training data generated by NeRF can help to improve downstream task performance. To further explore and exploit the satisfactory property of NeRF, we present a valuable and important application in this work: we leverage the NeRF-generated data to investigate and improve the robustness of the perception system in self-driving cars.

IV. Method

We illustrate the pipeline of Adv3D in Fig. 1. We aim to learn a transferable adversarial example in 3D detection that, when rendered in any pose (*i.e.*, location and rotation), can effectively hide surrounding objects from 3D detectors in any scenes by lowering their confidence. In Sec. IV-A, to improve the physical realizability of adversarial examples, we propose (1) Primitive-aware sampling to enable 3D patch attacks. (2) Disentangle NeRF that provides feasible geometry, and (3) Semantic-guided regularization that enables camouflage adversarial texture. To enhance the transferability across poses and scenes, we formulate the learning paradigm of Adv3D within the EOT framework [4] that is discussed in Sec. IV-C.

A. 3D Adversarial Example Generation

We use a gradient-based method to train our adversarial examples. The training pipeline involves 4 steps: **(i)** randomly sampling the pose of an adversarial example, **(ii)** rendering the example in the sampled pose, **(iii)** pasting the rendered patch into the original image of the training set, and finally, **(iv)** computing the loss and optimizing the latent codes. During inference, we discard the **(iv)** step.

1) Pose Sampling: To achieve adversarial attack in arbitrary object poses, we apply Expectation of Transformation (EOT) [4] by randomly sampling object poses. The poses of adversarial examples are parameterized as 3D boxes $\mathbf{b}$ that are restricted to a predefined ground plane in front of the camera. We model the ground plane as a uniform distribution $\mathcal{B}$ in a specific range that is detailed in the supplement. During training, we independently sample the rendering poses of adversarial examples, and approximate the expectation by taking the average loss over the whole batch.

2) Primitive-aware Sampling: We model the primitive of adversarial examples as NeRF tightly bound by 3D boxes, in order to enable non-contact and physically realistic attacks. During volume rendering, we compute the intersection of rays $\mathbf{r}(t)$ with the sampled pose $\mathbf{b} = \{\mathbf{R}, \mathbf{t}, \mathbf{s}\} \in \mathcal{B}$, finding the first hit point and the last hit point of box (t_{near}, t_{far}) by the AABB-ray intersection algorithm [50]. We then sample our points inside the range (t_{near}, t_{far}) to reduce large unnecessary samples and avoid contact with the environment.

$$(t_{near}, t_{far}) = Intersect(\mathbf{r}, \mathbf{b}), \tag{3}$$

$$\mathbf{r}'(t_k) = \tilde{\mathbf{r}}(t_{near}) + (\tilde{\mathbf{r}}(t_{far}) - \tilde{\mathbf{r}}(t_{near})) \cdot k/K, \tag{4}$$

$$\tilde{\mathbf{r}}(t) = Transform(\mathbf{r}(t), \mathbf{b}), \tag{5}$$

where $\tilde{\mathbf{r}}(t)$ is the sampled points with additional global to local transformation. Specifically, we use a 3D affine transformation to map original sampled points $\mathbf{r}(t) = \mathbf{r}_o + \mathbf{r}_d \cdot t$ into a canonical space $\tilde{\mathbf{r}} = \{x, y, z\} \in [-1, 1]$. This ensures that all the sampled points regardless of their distance from the origin, are transformed to the range $[-1, 1]$, thus providing a compact input representation for NeRF network. The transformation is given by:

$$Transform(\mathbf{r}, \mathbf{b}) = \mathbf{s}^{-1} \cdot (\mathbf{R}^{-1} \cdot \mathbf{r} - \mathbf{t}), \tag{6}$$

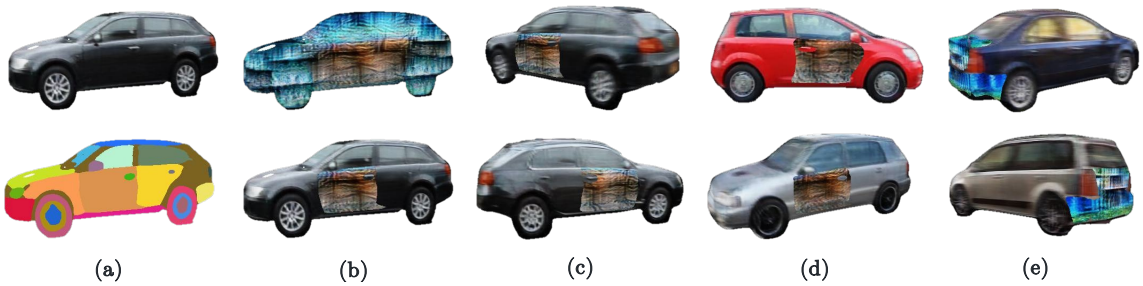


Fig. 2: Rendered results of adversarial examples. **(a)** Image and semantic label of an instance predicted by NeRF. **(b)** Top: our example without semantic-guided regularization. Bottom: our example with semantic-guided regularization. **(c)** Multi-view consistent synthesis of our examples. **(d,e)** The texture transfer results of side and back part adversary to other vehicles.

where $\mathbf{b} = \{\mathbf{R}, \mathbf{t}, \mathbf{s}\}$, $\mathbf{R} \in SO(3)$ is rotation matrix of the box, $\mathbf{t}, \mathbf{s} \in \mathbb{R}^3$ indicate translation and scale vector that move and scale the unit cube to desired location and size. The parameters of $\mathbf{b}$ are sampled from a pre-defined distribution $\mathcal{B}$ detailed in the supplement.

Then, the points lied in $[-1, 1]$ are projected to exactly cover the tri-plane features $\mathbf{z}$ for interpolation. Finally, a small MLP takes the interpolated features as input and predicts RGB and density:

$$(\mathbf{c}_k, \tau_k) = \mathrm{MLP}(Interpolate(\mathbf{z}, \mathbf{r}'(t_k))). \tag{7}$$

The primitive-aware sampling enables patch attacks [15] in a 3D-aware manner by lifting the 2D patch to a 3D box, enhancing the physical realizability by ensuring that the adversarial example only has a small modification to the original 3D environment.

3) Disentangled NeRF Parameterization: The original parameterization of NeRF combines the shape and texture into a single MLP, resulting in an entangled shape and texture generation. Since shape variation is challenging to reproduce in the real world, we disentangle shape and texture generation and only set the texture as adversarial examples. We obtain texture latents $\mathbf{z}_{\mathbf{tex}.}$ and shape latents $\mathbf{z}_{\mathbf{shape}}$ from the Lift3D. During volume rendering, we disentangle shape and texture generation by separately predicting RGB and density:

$$\mathbf{c}_k = \mathrm{Network}(\mathbf{z}_{\mathbf{tex}.}, \mathbf{r}'(t_k)), \tag{8}$$

$$\tau_k = \mathrm{Network}(\mathbf{z}_{\mathbf{shape}}, \mathbf{r}'(t_k)), \tag{9}$$

where $\mathbf{z}_{\mathbf{shape}}$ is fixed and $\mathbf{z}_{\mathbf{texture}}$ is being optimized. Our disentangled parametrization can also be seen as a geometry regularization in [12], [13] but keeps geometry unchanged as a usual vehicle, leading to a more realizable adversarial example.

4) Semantic-guided Regularization: Setting the full part of the vehicle as adversarial textures is straightforward, but not always feasible in the real world. To improve the physical realizability, we propose to optimize individual semantic parts, such as doors and windows of a vehicle. Specifically, as shown in Fig. 2 (d, e)), we only set a specific part of the vehicle as adversarial texture while maintaining others unchanged. This semantic-guided regularization leads to a camouflage adversarial texture that is less likely spotted in the real world and improves physical effectiveness.

To achieve this, we add a semantic branch to Lift3D to predict semantic part labels of the sampled points. We re-train Lift3D by fitting multi-view images and semantic labels generated by EditGAN [51]. Using semantic-guided regularization, we maintain the original texture and adversarial part texture at the same time but only optimize the adversarial part texture while leaving the original texture unchanged. This approach allows us to preserve a large majority of parts as usual, but to alter only the specific parts that are adversarial (see Fig. 2 (b, c)). In our implementation, we query the NeRF network twice, one for the adversarial texture and the other for the original texture. Then, we replace the part of original texture with the adversarial texture indexed by semantic labels in the point space.

Owing to this property, these adversarial textures can be printed and pasted on certain parts of cars to perform attacks. We provide real-world reproduction in supplementary video.

B. Gradient Propagation

After rendering the adversarial examples, we paste the adversarial patch into the original image through image composition. The attacked image can be expressed as $I_1 \times M + I_2 \times (1 - M)$ where I_1 and I_2 are the patch and original image, M is foreground mask predicted by NeRF. Next, the attacked images are fed to pretrained and fixed 3D detectors to compute the objective and back-propagate the gradients. Since both the rendering and detection pipelines are differentiable, Adv3D allows gradients from the objective to flow into the texture latent codes during optimization.

C. Learning Paradigm

We formulate our learning paradigm as EOT [4] that finds adversarial texture codes by minimizing the expectation of a binary cross-entropy loss over sampled poses and scenes:

$$\mathbf{z}_{\mathbf{tex}.} = \arg\min_{\mathbf{z}_{\mathbf{tex}.}} \mathbb{E}_{\mathbf{b}\sim\mathcal{B}} \mathbb{E}_{\mathbf{x}\sim\mathcal{X}} [-\log(1 - P(I(\mathbf{x}, \mathbf{b}, \mathbf{z}_{\mathbf{tex}.}))], \tag{10}$$

where $\mathbf{b}$ is the rendering pose sampled from the predefined distribution of ground plane $\mathcal{B}$, $\mathbf{x}$ is the original image sampled from the training set $\mathcal{X}$, $I(\mathbf{x}, \mathbf{b}, \mathbf{z}_{\mathbf{tex}.})$ is the attacked image that composited by the original image $\mathbf{x}$ and the adversarial patch rendered using pose $\mathbf{b}$ and texture latent code $\mathbf{z}_{\mathbf{tex}.}$, and $P(I(\cdot))$ represents the confidence of all proposals predicted by detectors. We approximate the expectation by averaging the

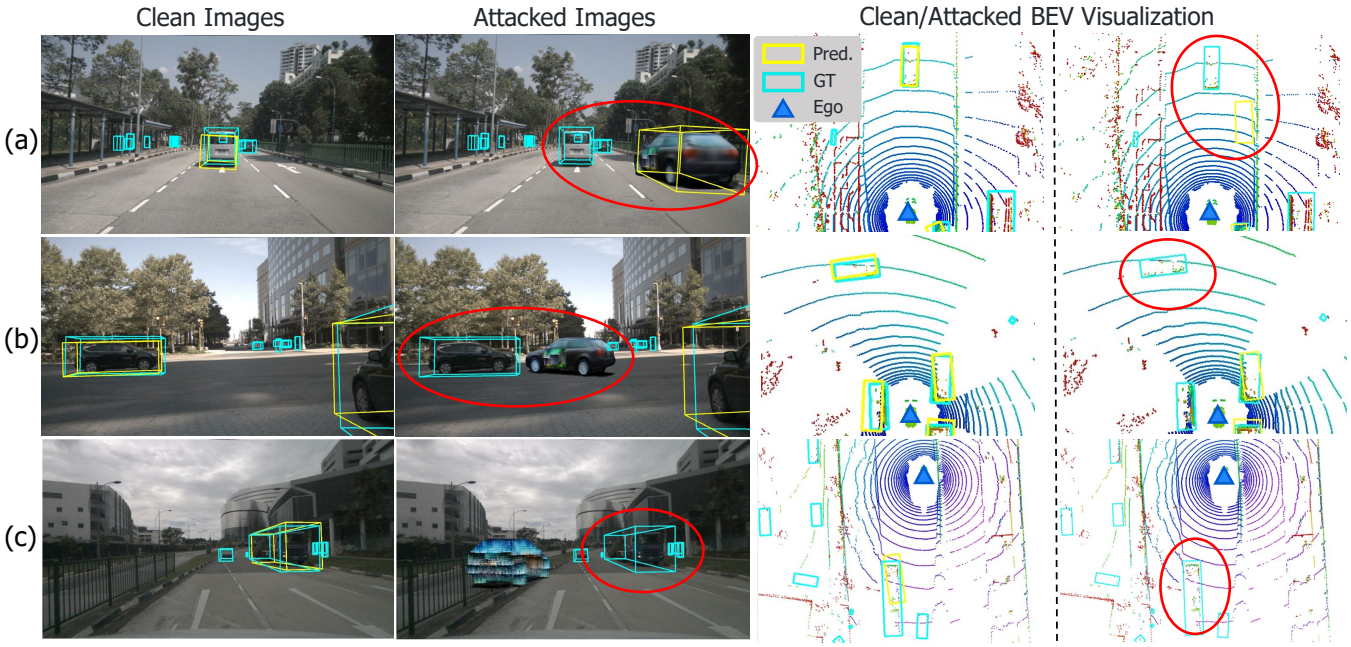


Fig. 3: Visualization of BEVDet prediction on nuScenes validation set under our attacks. The visualization threshold is set at 0.6. The adversarial NeRF can hide surrounding objects by minimizing their predicted confidence in a non-contact manner (making the yellow boxes disappear). Lidar point clouds are only used for visualization.

Models	Backbone	Type	Clean NDS	Adv NDS	Clean mAP	Adv mAP
FCOS3D [38]	ResNet101	FoV	0.3770	0.2674	0.2980	0.1272
PGD-Det [40]	ResNet101	FoV	0.3934	0.2694	0.3174	0.1321
DETR3D [39]	ResNet101	FoV	0.4220	0.2755	0.3470	0.1336
BEVDet [52]	ResNet50	BEV	0.3822	0.2247	0.3076	0.1325
BEVFormer-Tiny [53]	ResNet50	BEV	0.3540	0.2264	0.2524	0.1217
BEVFormer-Base [53]	ResNet101	BEV	0.5176	0.3800	0.4167	0.2376

TABLE II: Comparison of different detectors under our attack. Clean NDS and mAP denote evaluation using original validation data. Adv NDS and mAP denote evaluation using attacked data.

objective of the independently sampled batch. The objective is a binary cross-entropy loss that minimizes the confidence of all predicted bounding boxes, including adversarial objects and normal objects.

Built within the framework of EOT, Adv3D helps to improve the transferability and robustness of adversarial examples over the sampling parameters (poses and scenes here). This means that the attack can be performed without prior knowledge of the scene and are able to disrupt models across different poses and times in a non-contact manner.

D. Adversarial Defense by Data Augmentation

Toward defenses against our adversarial attack, we also study adversarial training to improve the robustness of 3D detectors. Adversarial training is typically performed by adding image perturbations using a few PGD steps [54], [55] during the training of networks. However, our adversarial example is too expensive to generate for the bi-level loop of the min-max optimization objective. Thus, instead of generating adversarial examples from scratch at every iteration, we leverage the transferable adversarial examples to augment the training set. We use the trained adversarial example to locally store a large number of rendered images to avoid repeated computation. During adversarial training, we randomly paste the rendered adversarial patch into the training images with a probability of 30%, while remaining others unchanged. We provide experimental results in Sec. V-D.

V. Experiments

In this section, we first present the experiments of semantic-guided regularization in Sec. V-A, the analysis of 3D attack in Sec. V-B, and our adversarial defense method in Sec. V-D. We provide real world experiments in supplementary video.

Dataset We conduct our experiments on the nuScenes dataset [19]. This dataset is collected using 6 surrounded-view cameras that cover the full 360° field of view around the ego-vehicle. It contains 700 scenes for training and 150 scenes for validation. In our work, we train our adversarial examples on the training set and evaluate performance drop on the validation set.

Target Detectors and Metrics As shown in Tab. II, we evaluate the robustness of six representative detectors. Three

Target \ Source	Clean	FCOS3D	PGD-Det	DETR3D	BEVDet	BEVFormer
FCOS3D [38]	0.298	**0.124**	0.141	0.144	0.176	0.158
PGD-Det [40]	0.317	0.172	**0.131**	0.150	0.186	0.172
DETR3D [39]	0.347	0.188	0.170	**0.133**	0.212	0.198
BEVDet [52]	0.307	0.148	0.145	0.140	**0.132**	0.140
BEVFormer [53]	0.252	0.175	0.155	0.136	0.177	**0.124**

TABLE III: Transferability of our attack to unseen detectors. We evaluate the robustness of **target** detectors using an adversarial example trained on **source** detectors. Reported in mAP.

are FoV-based, and three are BEV-based. Following prior work [18], we evaluate the performance drop on the validation set after the attack. Specifically, we use the Mean Average Precision (mAP) and nuScenes Detection Score (NDS) [19] to evaluate the performance of 3D detectors.

Quantitative Results We provide the experimental results of adversarial attacks in Tab. II. The attacks are conducted in a full-part manner without semantic-guided regularization to investigate the upper limit of attack performance. We found that, in spite of FoV-based or BEV-based, they display similar robustness. Meanwhile, we see a huge improvement of robustness by utilizing a stronger backbone (ResNet101 versus ResNet50) when comparing BEVFormer-Base with BEVFormer-Tiny. We hope these results will inspire researchers to develop 3D detectors with enhanced robustness.

Rendering Results We visualize our attack results with semantic-guided regularization in Fig. 3 (a,b), and without regularization in Fig. 3 (c). The disappearance of detected objects is caused by their lower confidence scores. For example, the confidence predicted by detectors in Fig. 3 (a) have declined from 0.6 to 0.4, and are therefore filtered out by the threshold of 0.6. In Fig. 3 (a), we find that our adversarial NeRF is realistic enough to be detected by a 3D detector if it doesn't display much of the adversarial texture. However, once the vehicle shows a larger area of the adversarial texture as seen in Fig. 3 (b), it will hide all objects including itself due to our untargeted objective.

A. Semantic Parts Analysis

In Tab. IV, we provide experiments on the impact of different semantic parts on attack performance. We use three salient parts of the car: the front, side, and rear. It shows that compared with adversarial attacks using full parts, the semantic-guided regularization leads to a slightly lower performance drop, but remains a realistic appearance and less likely spotted adversarial texture as illustrated in Fig. 2 (b).

Since we do not have access to annotation during training, we additionally conduct "No Part" experiment that no part of the texture is adversarial, to evaluate the impact of occlusion. We acknowledge that part of performance degradation can be attributed to the occlusion to original objects and the false positive prediction of adversarial objects (see Fig. 3 (a)), since we do not update the ground truth of adversarial objects to the validation set.

Part	NDS	mAP	Part	NDS	mAP
Clean	0.382	0.307	Front	0.267	0.148
No Part	0.302	0.234	Side	0.265	0.149
Full Parts	0.224	0.132	Rear	0.268	0.151

TABLE IV: Ablations of different semantic parts.

B. Effectiveness of 3D-aware attack

To validate the effectiveness of our 3D attacks, we ablate the impact of different poses on the attack performance. In Fig. 4 (a), we divide the BEV plane into 10×10 bins ranging from $x \in [-5m, 5m]$ and $z \in [10m, 15m]$. We then evaluate the relative mAP drop (percentage) of BEVDet [52] by sampling one adversarial example inside the bin per image, while keeping rotation randomly sampled. Similarly, we conduct experiments of 30 uniform rotation bins ranging from $[0, 2\pi]$ in Fig. 4 (b). The experimental results demonstrate that all aspects of location and rotation achieve a valid attack (performance drop $> 30\%$), thereby proving the transferability of poses in our 3D-aware attack.

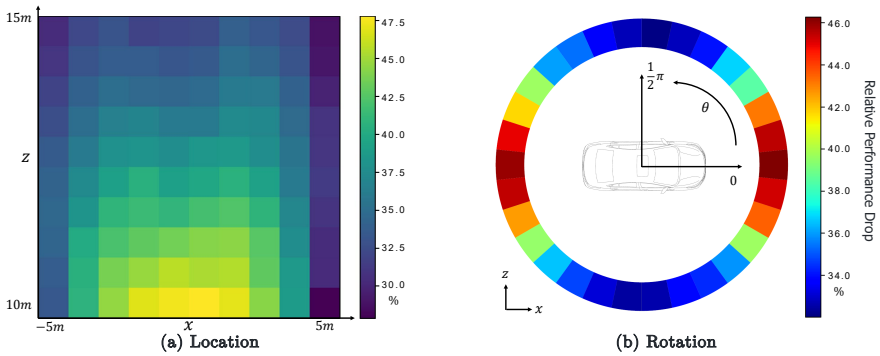
A finding that contrasts with prior work [12] is the impact of near and far locations in z axis. Our adversarial example is more effective in the near region compared with the far region, while [12] displays a roughly uniform distribution in all regions. We hypothesize that the attack performance is proportional to the area of the rendered patch, which is highly related to the location of objects. Similar findings are also displayed in rotation. The vehicle that poses vertically to the ego vehicle results in a larger rendered area, thus better attack performance.

Data	Adv train	NDS	mAP
Clean val	✗	0.304	0.248
Clean val	✓	**0.311**	**0.255**
Adv val †	✗	0.224	0.132
Adv val †	✓	**0.264**	**0.181**
Adv val §	✓	0.228	0.130

TABLE V: Results of adversarial training. The symbol † indicates attacks using the same adversarial example used in adversarial training, while § indicates a different example.

C. Transferability Across Different Detectors

In Tab. III, we evaluate the transferability of adversarial examples across different detectors. To this end, we train a



(a) Location (b) Rotation

Fig. 4: To examine the 3D-aware property of our adversarial examples, we ablate the relative performance drop by sampling adversarial examples within different bins of location and rotation.

single adversarial example of each detector separately, then use the example to evaluate the performance drop of other detectors. We show that there is a high degree of transferability between different models. Among them, we observe that DETR3D [39] appears to be more resilient to adversarial attacks than other detectors. We hypothesize this can be attributed to the sparsity of the query-based method. During the projection of 3D query to the 2D image plane, only a single point of the feature is indexed by interpolation, thus fewer areas of adversarial features will be sampled. This finding may have insightful implications for the development of more robust 3D detectors in the future.

D. Adversarial Defense by Data Augmentation

We present the results of adversarial training in Tab. V. We observe that incorporating adversarial training improves not only the robustness against the seen adversarial examples, but also the clean performance. However, we also note that our adversarial training is not capable of transferring to unseen adversarial examples trained in the same way, mainly due to the fixed adversarial example during adversarial training. Furthermore, we hope that future work can conduct in-depth investigations and consider handling the bi-level loop of adversarial training in order to better defend against adversarial attacks.

VI. Limitation and Future Work

While our method demonstrated promising results in attacks and downstream improvements, it is important to acknowledge our current limitations.

Perceptibility As we optimize the adversarial texture with only semantic regularization, the adversarial part texture still can be easily detected by humans as shown in Fig 3. Future work can explore more advanced techniques [56]–[58] to build inconspicuous 3D adversarial examples.

Lighting Since our optimized adversarial examples have a different data source from the original datasets. There exists an illumination gap between the rendered patch and the original environment. This gap can lead to domain gaps when deploying adversarial examples in the real world. In future work, researchers can explore the use of physical-based rendering techniques [34], [59] to address this issue and train lighting-invariant adversarial examples.

Broader Impact The recent development of NeRF has led to remarkable progress in NeRF-based driving scene simulation. Our adversarial framework is general and can be extended to integrate with the advances in NeRF-based simulators to benefit a wide spectrum of practical systems. For instance, our framework can be combined with UniSim [60] to perform adversarial closed-loop evaluations of self-driving cars in NeRF environments, or with ClimateNeRF [61] to identify adverse weather conditions that may corrupt the autonomous driving system. We believe that our work provides valuable insights and opens up new possibilities for creating authentic adversarial evaluations that verify and improve the robustness of self-driving cars.

VII. Conclusion

In this paper, we propose **Adv3D**, the first attempt to model adversarial examples as NeRF in driving scenarios. Adv3D enhances the physical realizability of attacks through our proposed primitive-aware sampling and semantic-guided regularization. Compared with prior works of adversarial examples in autonomous driving, our examples are more threatening in practice as we carry non-contact attacks, have feasible 3D shapes as usual vehicles, and display camouflage adversarial texture. Extensive experimental results also demonstrate that Adv3D achieves better attack performance and transfers well to different poses, scenes, and detectors. We hope our work provides valuable insights for creating more realistic evaluations to investigate and improve the robustness of autonomous driving systems.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *ICLR*, 2014.
[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *ICLR*, 2015.
[3] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *ICLR Workshop*, 2017.
[4] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," 2018.
[5] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *CVPR*, 2017.
[6] C. Xiang, C. R. Qi, and B. Li, "Generating 3d adversarial point clouds," in *CVPR*, 2019.
[7] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, "Benchmarking adversarial robustness on image classification," in *CVPR*, 2020.

[8] C. Xiao, D. Yang, B. Li, J. Deng, and M. Liu, "Meshadv: Adversarial meshes for visual recognition," in *CVPR*, 2019.

[9] X. Zeng, C. Liu, Y.-S. Wang, W. Qiu, L. Xie, Y.-W. Tai, C.-K. Tang, and A. L. Yuille, "Adversarial attacks beyond the image space," in *CVPR*, 2019.

[10] Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, and B. Li, "Adversarial objects against lidar-based autonomous driving systems," *arXiv preprint arXiv:1907.05418*, 2019.

[11] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li, "Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks," in *IEEE Symposium on Security and Privacy (SP)*, 2021.

[12] J. Tu, M. Ren, S. Manivasagam, M. Liang, B. Yang, R. Du, F. Cheng, and R. Urtasun, "Physically realizable adversarial examples for lidar object detection," in *CVPR*, 2020.

[13] J. Tu, H. Li, X. Yan, M. Ren, Y. Chen, M. Liang, E. Bitar, E. Yumer, and R. Urtasun, "Exploring adversarial robustness of multi-sensor perception systems in self driving," *CORL*, 2021.

[14] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[15] A. Sharma, Y. Bian, P. Munz, and A. Narayan, "Adversarial patch attacks and defences in vision-based tasks: A survey," *arXiv preprint arXiv:2206.08304*, 2022.

[16] L. Li, Q. Lian, L. Wang, N. Ma, and Y.-C. Chen, "Lift3d: Synthesize 3d training data by lifting 2d gan to 3d generative radiance field," in *CVPR*, 2023.

[17] Z. Zhu, Y. Zhang, H. Chen, Y. Dong, S. Zhao, W. Ding, J. Zhong, and S. Zheng, "Understanding the robustness of 3d object detection with bird's-eye-view representations in autonomous driving," *arXiv preprint arXiv:2303.17297*, 2023.

[18] S. Xie, Z. Li, Z. Wang, and C. Xie, "On the adversarial robustness of camera-based 3d object detection," *TMLR*, 2023.

[19] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[20] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen, "Dpatch: An adversarial patch attack on object detectors," *arXiv preprint arXiv:1806.02299*, 2018.

[21] S.-T. Chen, C. Cornelius, J. Martin, and D. H. Chau, "Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector," in *ECML PKDD*, 2019.

[22] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, and X. Lin, "Adversarial t-shirt! evading person detectors in a physical world," in *ECCV*, 2020.

[23] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.

[24] L. Huang, C. Gao, Y. Zhou, C. Xie, A. L. Yuille, C. Zou, and N. Liu, "Universal physical camouflage attacks on object detectors," in *CVPR*, 2020.

[25] T. Wu, X. Ning, W. Li, R. Huang, H. Yang, and Y. Wang, "Physical adversarial attack on vehicle detector in the carla simulator," *arXiv preprint arXiv:2007.16118*, 2020.

[26] Y. Zhang, H. Foroosh, P. David, and B. Gong, "Camou: Learning physical vehicle camouflages to adversarially attack detectors in the wild," in *International Conference on Learning Representations*, 2019.

[27] A. Hamdi, S. Rojas, A. Thabet, and B. Ghanem, "Advpc: Transferable adversarial perturbations on 3d point clouds," in *ECCV*, 2020.

[28] A. Hamdi, M. Müller, and B. Ghanem, "Sada: semantic adversarial diagnostic attacks for autonomous applications," in *AAAI*, 2020.

[29] Y. Dong, S. Ruan, H. Su, C. Kang, X. Wei, and J. Zhu, "Viewfool: Evaluating the robustness of visual recognition to adversarial viewpoints," *NeurIPS*, 2022.

[30] S. Ruan, Y. Dong, H. Su, J. Peng, N. Chen, and X. Wei, "Improving viewpoint robustness for visual recognition via adversarial training," *arXiv preprint arXiv:2307.11528*, 2023.

[31] J. Sarva, J. Wang, J. Tu, Y. Xiong, S. Manivasagam, and R. Urtasun, "Adv3d: Generating safety-critical 3d objects through closed-loop simulation," *arXiv preprint arXiv:2311.01446*, 2023.

[32] H. Chen, Z. Chen, G. P. Meyer, D. Park, C. Vondrick, A. Shrivastava, and Y. Chai, "Shift3d: Synthesizing hard inputs for tricking 3d detectors," in *ICCV*, 2023.

[33] M. Abdelfattah, K. Yuan, Z. J. Wang, and R. Ward, "Adversarial attacks on camera-lidar models for 3d car detection," in *IROS*, 2021.

[34] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ToG*, 2021.

[35] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, 1984.

[36] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *CVPR*, 2018.

[37] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *ICCV*, 2019.

[38] T. Wang, X. Zhu, J. Pang, and D. Lin, "FCOS3D: Fully convolutional one-stage monocular 3d object detection," in *ICCV Workshop*, 2021.

[39] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, , and J. M. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *CoRL*, 2021.

[40] T. Wang, X. Zhu, J. Pang, and D. Lin, "Probabilistic and Geometric Depth: Detecting objects in perspective," in *CoRL*, 2021.

[41] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *ECCV*, 2020.

[42] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distributionnetwork for monocular 3d object detection," *CVPR*, 2021.

[43] Y. Ma, T. Wang, X. Bai, H. Yang, Y. Hou, Y. Wang, Y. Qiao, R. Yang, D. Manocha, and X. Zhu, "Vision-centric bev perception: A survey," 2022.

[44] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *CVPR*, 2022.

[45] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM ToG*, 2022.

[46] J. Gu, L. Liu, P. Wang, and C. Theobalt, "Stylenerf: A style-based 3d aware generator for high-resolution image synthesis," in *ICLR*, 2022.

[47] N. Max, "Optical models for direct volume rendering," *IEEE TVCG*, 1995.

[48] T. Karras, S. Laine, Aittala, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *CVPR*, 2020.

[49] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. D. Mello, O. Gallo, L. Guibas, J. Tremblay, S. Khamis, T. Karras, and G. Wetzstein, "Efficient geometry-aware 3D generative adversarial networks," in *CVPR*, 2022.

[50] A. Majercik, C. Crassin, P. Shirley, and M. McGuire, "A ray-box intersection algorithm and efficient dynamic voxel rendering," *Journal of Computer Graphics Techniques Vol*, vol. 7, no. 3, 2018.

[51] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler, "Editgan: High-precision semantic image editing," in *NeurIPS*, 2021.

[52] J. Huang, G. Huang, Z. Zhu, Y. Yun, and D. Du, "Bevdet: High-performance multi-camera 3d object detection in bird-eye-view," *arXiv preprint arXiv:2112.11790*, 2021.

[53] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," *ECCV*, 2022.

[54] 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.

[55] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *CVPR*, 2020.

[56] T. Bai, J. Luo, and J. Zhao, "Inconspicuous adversarial patches for fooling image-recognition systems on mobile devices," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9515–9524, 2021.

[57] S. Jia, B. Yin, T. Yao, S. Ding, C. Shen, X. Yang, and C. Ma, "Adv-attribute: Inconspicuous and transferable adversarial attack on face recognition," *NeurIPS*, 2022.

[58] H. Mohaghegh Dolatabadi, S. Erfani, and C. Leckie, "Advflow: Inconspicuous black-box adversarial attacks using normalizing flows," *NeurIPS*, 2020.

[59] Z. Wang, W. Chen, D. Acuna, J. Kautz, and S. Fidler, "Neural light field estimation for street scenes with differentiable virtual object insertion," in *ECCV*, 2022.

[60] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, "Unisim: A neural closed-loop sensor simulator," in *CVPR*, 2023.

[61] Y. Li, Z.-H. Lin, D. Forsyth, J.-B. Huang, and S. Wang, "Climatenerf: Extreme weather synthesis in neural radiance field," in *CVPR*, 2023.