RAUCA: A Novel Physical Adversarial Attack on Vehicle Detectors via Robust and Accurate Camouflage Generation

Jiawei Zhou [1] **Linye Lyu** [1] **Daojing He** [1] **Yu Li** [1]

Abstract

Adversarial camouflage is a widely used physical attack against vehicle detectors for its superiority in multi-view attack performance. One promising approach involves using differentiable neural renderers to facilitate adversarial camouflage optimization through gradient back-propagation. However, existing methods often struggle to capture environmental characteristics during the rendering process or produce adversarial textures that can precisely map to the target vehicle, resulting in suboptimal attack performance. Moreover, these approaches neglect diverse weather conditions, reducing the efficacy of generated camouflage across varying weather scenarios. To tackle these challenges, we propose a robust and accurate camouflage generation method, namely RAUCA. The core of RAUCA is a novel neural rendering component, Neural Renderer Plus (NRP), which can accurately project vehicle textures and render images with environmental characteristics such as lighting and weather. In addition, we integrate a multi-weather dataset for camouflage generation, leveraging the NRP to enhance the attack robustness. Experimental results on six popular object detectors show that RAUCA consistently outperforms existing methods in both simulation and real-world settings.

1. Introduction

Deep Neural Networks (DNNs) have achieved remarkable performance in many real-world applications such as face recognition and autonomous vehicles (Krizhevsky et al., 2012; Wang et al., 2023; Chai et al., 2021). However, DNNs suffer from adversarial examples (Szegedy et al., 2014). For instance, in the task of vehicle detection, adversarial inputs can deceive detection models, leading to incorrect detection of the surrounding vehicles, which poses a severe threat to the safety of autonomous vehicles.

(a) Normal (b) DAS (c) FCA (d) ACTIVE (e) RAUCA (ours)

Figure 1. Comparison of different adversarial camouflage under sunny (first row) and foggy (second row) environments, where only our method succeeds in both cases. (a) A car with normal texture; (b) DAS (Wang et al., 2021). (c) and (d) are top-performed methods FCA (Wang et al., 2022) and ACTIVE (Suryanto et al., 2023), respectively. (e) Our method RAUCA.

Adversarial examples can be categorized into digital and physical adversarial types. Digital adversarial examples involve introducing pixel-level perturbations to the model's digital input, while physical ones manipulate real-world objects or surroundings, indirectly influencing the model inputs (Chen et al., 2018; Sharif et al., 2016a; Mahendran & Vedaldi, 2015). Physical adversarial examples are generally deemed more practical, as gaining direct access to the model's input often necessitates system authentication. However, they are inherently more challenging as they must prove effective in complex physical environments, including various viewing perspectives, spatial distances, and lighting/weather conditions.

This paper focuses on physical adversarial examples against vehicle (e.g., car) detection models, given their wide adoption in autonomous driving scenarios where safety is of great importance. To ensure the attack effectiveness across various viewing angles, current methods prefer generating adversarial camouflage capable of covering the entire surface of the vehicle (Wang et al., 2021; 2022; Suryanto et al., 2022; 2023). Top-performing methods achieve this by leveraging a differential neural renderer. This renderer maps the 3D vehicle and its texture to 2D images, establishing a differentiable path between the 3D vehicle and the vehi-

[1]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, P.R. China. Correspondence to: Yu Li <li.yu@hit.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

cle detection models. Thus, the texture can be optimized through gradient back-propagation for effective camouflage generation.

In the literature, there are two ways to generate camouflage with a neural renderer: one way is to optimize a 2D square texture pattern and project it onto the vehicle repeatedly, referred to as world-align-based methods; the other is to optimize the 3D texture of the vehicle in the form of UV-maps, referred to as UV-map-based method. However, both approaches currently suffer from certain issues. The former method cannot guarantee the texture pattern projected onto the car in the same manner in both texture generation and evaluation, leading to differences in adversarial camouflage between generation and evaluation. The latter UV-map-based render methods are not capable of rendering sophisticated environment characteristics such as light and weather, leading to sub-optimal camouflage generation. Besides, all current methods fail to consider the effectiveness of the camouflage under various weather conditions. As shown in Figure 1, the camouflage generated by the top-performed FCA (Wang et al., 2022) and ACTIVE (Suryanto et al., 2023) methods fails to attack the detector in a foggy scene.

To address the above issues, we develop a novel adversarial camouflage generation framework against vehicle detectors. Our key insight is that, for successful physical attacks, the generated camouflage must accurately map onto the vehicle. Additionally, the camouflage needs to be robust under different environmental conditions. Achieving the first goal necessitates an end-to-end optimization of the camouflage. The second goal requires a dataset encompassing ample environmental effects and also requires that these effects can be successfully utilized for camouflage optimization. To fulfill these objectives, we leverage the UV-map-based neural renderer for accurate texture optimization. We adapt this renderer to achieve photo-realistic rendering of the target vehicle. Furthermore, we augment the dataset used for camouflage generation to include diverse weather conditions. This enhanced dataset, combined with the refined renderer, facilitates the generation of effective camouflage.

Our contributions can be summarized as follows:

- We present the Robust and Accurate UV-map-based Camouflage Attack (RAUCA), a framework for generating physical adversarial camouflage against vehicle detectors. It enhances the effectiveness and robustness of the adversarial camouflage through a novel rendering component and a multi-weather dataset.
- We propose the Neural Renderer Plus (NRP), a novel neural rendering component that allows for the optimization of textures that can be accurately mapped to vehicle surfaces and can render images with environmental characteristics such as lighting and weather.
- We incorporate a multi-weather dataset with ample environmental effects into the camouflage generation process. Our experiments show that the use of this dataset substantially enhances the attack robustness when using NRP for rendering.

Our extensive studies demonstrate that our method outperforms the current state-of-the-art method by around 10% in car detection performance (e.g., AP@0.5). Our attacks achieve strong robustness and effectiveness, excelling in multi-weather, multi-view, and multi-distance conditions. We also demonstrate our method is effective in various weather conditions in the physical world. Our code is available at: https://github.com/SeRAlab/Robust-and-Accurate-UV-map-based-Camouflage-Attack.

2. Related work

Physical Adversarial Attack: Physical adversarial attacks need to consider the robustness of the attacks because the objects and the environment are not constant. The Expectation over Transformation (EoT) (Athalye et al., 2018) is a prime method of generating robust adversarial examples under various transformations, such as lighting conditions, viewing distances, angles, and background scenes. As a result, many adversarial camouflage methods (Zhang et al., 2019; Wu et al., 2020; Wang et al., 2021; 2022; Suryanto et al., 2022; 2023) employ EoT-based algorithms to enhance their attack robustness in the real world scenarios.

Adversarial Camouflage: For self-driving cars, precise detection of surrounding vehicles is a critical safety requirement. Consequently, there has been a growing interest in developing adversarial vehicle camouflage to evade vehicle detection systems. Current research works mostly leverage a 3D simulation environment to obtain 2D rendered vehicle images with various transformations such that they can obtain robust adversarial camouflage. The early researches of adversarial camouflage for vehicles are mostly black-box methods because the rendering process in the simulation environment is non-differentiable. Zhang et al. (2019) first conducts experiments in the 3D space to generate adversarial camouflage for cars. They propose CAMOU, a method to train an approximate gradient network to mimic the behavior of both rendering and detection of the camouflage vehicles. Then, they can optimize the adversarial texture using this network. Meanwhile, Wu et al. (2020) propose an adversarial camouflage generation framework based on a genetic algorithm to search an adversarial texture pattern. Then, they repeat the optimized texture pattern to build the 3D texture that covers the full-body vehicle.

Recent methods introduce neural renderers, which enable differentiable rendering. With this technique, the adversar-

ial texture can be optimized via gradient back-propagation. Two primary methods are employed in adversarial camouflage generation with neural renderer. One is to optimize a 3D model texture directly, which we refer to as UV-map-based camouflage methods; for example, Wang et al. (2021) propose a Dual Attention Suppression (DAS) attack, which minimizes the model attention and human attention on the camouflaged vehicle. Besides, Wang et al. (2022) propose the Full-coverage Camouflage Attack (FCA), which optimizes full-body surfaces of the vehicle in multi-view scenarios. The other is to optimize a 2D square texture pattern and then project it repeatedly to the target vehicle surface, which we refer to as world-align-based camouflage methods. Suryanto et al. (2022) present the Differentiable Transformer Attack (DTA), which proposes a differentiable renderer that can express physical and realistic characteristics (e.g., shadow). ACTIVE, proposed by Suryanto et al. (2023), introduces a texture mapping technique that utilizes depth images and improves the naturalness of the camouflage by using larger texture resolution and background colors.

While prior neural-renderer-based methods have achieved impressive attack success rates, they typically struggle to capture the environmental characteristics such as shadow or produce adversarial camouflage that can precisely map to the target vehicle, resulting in sub-optimal camouflage. Moreover, these methods frequently overlook diverse weather conditions during the generation of adversarial camouflage, hindering the effectiveness of the camouflage in varying weather environments.

3. Method

In this section, we present the overview of our framework for generating adversarial camouflage. Subsequently, we provide an in-depth explanation of the essential components comprising our attack framework, RAUCA.

3.1. Overview

Figure 2 shows our entire framework for adversarial camouflage generation. Firstly, we modify the weather parameters in the simulation environment to obtain a multi-weather vehicle dataset (I_{in}, Y, Φ_{cam}, M), where I_{in} is the original input images, Y is the ground truth labels, Φ_{cam} is the camera pose parameters (position and angle) for viewing the car, and M is the binary masks of I_{in}, where the target vehicle areas are set to 0. With I_{in} and M, we can use:

$$X_{ref} = I_{in} \cdot (1 - M) \tag{1}$$

$$B = I_{in} \cdot M \tag{2}$$

to obtain the foreground car images X_{ref} and the ambient background images B. Then, we use the Neural Renderer Plus (NRP) N, our proposal neural rendering component, to obtain the rendered 2D vehicle images X_{ren} through

$$X_{ren} = N\left(Msh, T_{adv}, \Phi_{cam}, X_{ref}\right), \tag{3}$$

where Msh and T_{adv} are the 3D mesh and the UV texture map of the vehicle, respectively. To obtain the realistic vehicle pictures, we apply a simple transformation $I_{out} = X_{ren} + B$ to attach the foreground X_{ren} to the corresponding background. We then input I_{out} into the target detector D and obtain the detection results with $R = D(I_{out})$.

Our framework aims to generate adversarial camouflages for vehicles to evade the detection of the vehicle detector. We can obtain the final adversarial camouflage through the solution of a specific optimization problem denoted as

$$\underset{T_{adv}}{\operatorname{argmax}} \mathcal{L}\left(D\left(N\left(Msh, T_{adv}, \Phi_{cam}, X_{ref}\right) + B\right), Y\right), \tag{4}$$

where $\mathcal{L}$ is our proposal loss function.

3.2. Multi-Weather Dataset

According to Athalye et al.'s EOT study (Athalye et al., 2018), adding various weather conditions to training data notably boosts attack robustness. Nevertheless, real-world multi-weather dataset collection is hindered by high labor expenses, weather's inherent unpredictability, and difficulties in getting the mask M of the vehicle.

To address the above difficulties, we use CARLA (Dosovitskiy et al., 2017), an autonomous driving simulation environment based on Unreal Engine 4 (UE4), to obtain the multi-weather dataset. Modifying the weather and time parameters with CARLA API to simulate different weather and light environment conditions is convenient. Moreover, with its built-in semantic segmentation camera, we can accurately and conveniently segment foreground and background.

In the assembly of the Multi-Weather Dataset, as shown in Figure 2, we strategically vary the sun altitude angle and fog density to simulate different weather conditions. The sun altitude angle is instrumental in modulating the intensity of sunlight within the environment, which in turn influences the light's interaction with the vehicle's camouflage. This variation in lighting can significantly affect the visibility and effectiveness of the camouflage. Concurrently, the density of fog, a critical environmental parameter, determines the extent to which the vehicle's surface is obscured. At specific densities, fog can effectively render parts of the texture partially or entirely invisible, a factor that is crucial in determining the success of adversarial textures.

3.3. Neural Render Plus (NRP)

In the rendering phase, we introduce NRP, a novel rendering component that mitigates the limits of the previous neural-renderer-based camouflage methods. The world-align-based

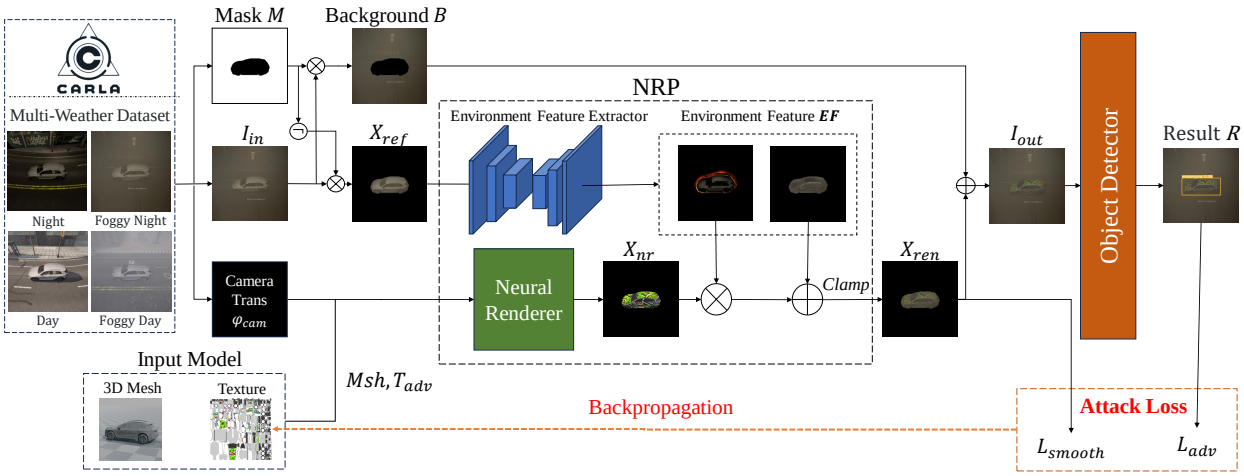

Figure 2. The overview of RAUCA. First, a multi-weather dataset is created using CARLA, which includes car images, corresponding mask images, and camera angles. Then the car images are segmented using the mask images to obtain the foreground car and background images. The foreground car, together with the 3D model and the camera angle is passed through the NRP rendering component for rendering. The rendered image is then seamlessly integrated with the background. Finally, we optimize the adversarial camouflage through back-propagation with our devised loss function computed from the output of the object detector.

Figure 3. Comparison of rendering results of neural renderers used in different methods. The first row shows the results obtained by different neural renderers (already blended with the background) and the second row shows the rendered results in UE4. Our renderer is the only one that does both foreground environment rendering and texture rendering similar to UE4.

methods cannot accurately wrap the adversarial camouflage to the vehicle surface, which might weaken its attack performance in the real world. Hence, we choose a UV-map-based neural renderer to avoid this issue. However, the UV-map-based methods struggle to render the environmental characteristics on the vehicle surface, resulting in unrealistic rendered images. Therefore, we introduce the environment feature extractor that can combine the environmental characteristics and neural renderer output to obtain a realistic and accurate image of the camouflaged vehicle.

Following (Wang et al., 2021; 2022), we use Neural Renderer (NR) (Kato et al., 2018) , whose input are 3D mesh Msh, adversarial texture T_{adv} and camera angle Φ_{cam}, to generate a rendered image of camouflaged vehicle. However, NR uses its own light source during rendering, which makes it impossible to render complex environmental characteristics similar to UE4. To amend this, we adopt the method proposed in DTA and ACTIVE (Suryanto et al., 2022; 2023), introducing an encoder-decoder network to extract the environmental characteristics X_{ref}. Since the NR's output X_{nr} already encompasses the shape, rotation, and texture transformation of the vehicle, our network only needs to learn the transformation of environmental characteristics. We call this network as Environment Feature Extractor (EFE). EFE's outputs are two maps of environment features EF. We can fuse them with X_{nr} through pixel-by-pixel multiplication and addition to get X_{ren}, an image of a textured vehicle with environmental characteristics.

Before applying the framework to generate adversarial textures, NRP needs to be trained in advance to optimize the parameters of EFE. The NRP training process inputs the masked images of the white car X_{ref}, 3D mesh Msh, poses of the camera Φ_{cam}, and multiple preset 3D texture T with different colors. At the same time, we need to obtain images of different vehicle colors based on Φ_{cam} and the colors of T from the photo-realistic renderer. Then, we cut out the vehicle parts as the network output target TG.

For each iteration of the training process, we input $x_{ref} \in X_{ref}$, Msh, T and $\varphi_{cam} \in \Phi_{cam}$ into NRP to get the rendering result x_{ren}. To optimize the parameters of the network, we compute the following loss function

$$L_{EFE}(x_{ref}) = W(x_{ref})\, BCE(x_{ren}, tg), \qquad (5)$$

where BCE is the binary cross-entropy loss, and $W(x_{ref}) = \frac{h*w}{s}$ is a weight function. h and w are the height and width of x_{ref} and s is the number of pixel points in the vehicle part of x_{ref}. We introduce $W(x_{ref})$ to balance NRP's rendering optimization across various camera viewpoints. The original BCE loss, calculated over the entire image, unfairly prioritizes rendering when the car occupies a small

area of the entire image. To address this, we multiply BCE by $W(x_{ref})$ to compute the difference over the vehicle area, improving rendering performance for views where the car occupies a small image area.

Compared to NR, the rendering component used in FCA (Wang et al., 2022) and DAS (Wang et al., 2021), our NRP considers the environmental characteristics to make the final input image to the target detector more realistic. Compared to neural renderers used in DTA and ACTIVE (Suryanto et al., 2022; 2023), our rendering component can render the adversarial camouflage based on UV mapping projection instead of world-aligned projection, which makes our textures more robust to multiple views. Moreover, our renderer has a more accurate projection of the texture compared to DTN and NSR, due to the use of NR for texture projection.

As shown in Figure 3, the renderings obtained by DTA and ACTIVE's renderers have a noticeable difference in the vehicle textures from that in UE4. Additionally, the FCA's renderer is not able to represent the complexity of light and shadow information, showing a clear difference between the foreground and the background. In contrast, the result of our rendering component is relatively accurate both in terms of environmental characteristics and texture mapping.

3.4. Attack Loss

We propose a novel attack loss function that consists of two key components for improving attack effectiveness. The first component is the Intersection Over Union (IOU) between the object detection model's output detection box and the ground-truth box; the second component comprises the class confidence score and objectiveness score of the output. The loss function is denoted as

$$\begin{aligned} H_d(x) &= \mathrm{IoU}\left(H_b\left(x\right), gt\right) * H_c\left(x\right) * H_o\left(x\right) \\ L_{atk}\left(x\right) &= -\log\left(1 - \max\left(H_d(x)\right)\right), \end{aligned} \tag{6}$$

where x is the input image of the target detector, $H_b(x)$ is the detection bounding box, gt is the ground-truth box and $IoU\left(H_b(x), gt\right)$ is the Intersection over Union (IoU) between $H_b(x)$ and gt. We use $IoU\left(H_b(x), gt\right)$ as a weight term, which allows the optimized loss function to focus more on the bounding box with a larger intersecting ratio area with gt. $H_o(x)$ and $H_c(x)$ represent the objectiveness score and the car class confidence score for the bounding box, respectively. $H_d\left(x\right)$ is our detection score, which is the product of the objectiveness confidence, class confidence, and the intersect ratio. Due to $IoU\left(H_b(x), gt\right)$, we only assign a non-zero detection score to the detected boxes which are intersected with the ground-truth box. This makes the texture optimization focus on making the target vehicle detection ineffective. We select the highest $H_d(x)$ and use it to compute $L_{atk}(x)$ through a log loss. By minimizing $L_{atk}(x)$, we can make the camouflaged vehicle misclassified or undetected by the object detector.

3.5. Smooth Loss

To ensure the smoothness of the generated texture for human vision, we follow (Sharif et al., 2016b) to utilize smooth loss L_{sm} to enhance texture consistency. We use the output of NRP x_{ren} to compute for texture smoother in natural light and shadow. The smooth loss function can be defined as:

$$L_{sm} = \sum_{i,j} \left(x_{i,j} - x_{i+1,j}\right)^2 + \left(x_{i,j} - x_{i,j+1}\right)^2, \tag{7}$$

where $x_{i,j}$ is the pixel value of x_{ren} at coordinate (i, j).

Finally, our loss function L_{total} can be summarized as

$$L_{total} = \alpha L_{atk} + \beta L_{sm}, \tag{8}$$

where α, β are the hyperparameters to control the contribution of each loss function.

4. Experiments

In this section, we first describe our experimental settings. Then we evaluate the attack effectiveness of our adversarial camouflage in both simulated and physical environments.

4.1. Experimental Settings

Datasets: We utilize CARLA to generate datasets in our experimentation. To facilitate a comparative analysis with previous studies (Wang et al., 2021; 2022; Suryanto et al., 2022; 2023), we select the Audi E-Tron as the target vehicle model. We create datasets using various simulation settings. For NRP training and testing, we utilized 69,120 and 59,152 photo-realistic images encompassing 16 distinct weather conditions, respectively. These conditions are a combination of four sun altitudes and four fog densities. Additionally, we generate a set of 40,960 images for our texture generation and prepare 7593 images for each adversarial camouflage in robust evaluation. The weather conditions present in these two datasets are the same conditions employed during the training and testing of NRP. For transferability evaluation, we utilize an "unseen-weather dataset" consisting of 7593 images in 16 novel weather conditions that are not seen during the texture generation phase. Moreover, we conduct experiments in the real world by printing five types of adversarial camouflages and sticking them to the 1:12 Audi E-Tron car models. We shoot 432 photos under different light conditions and 120 photos in a foggy environment for each model. Additional information regarding the construction process of the datasets can be found in the appendix.

Compared methods: We compare our framework with the state-of-the-art adversarial camouflage methods: DAS (Wang et al., 2021), FCA (Wang et al., 2022), DTA

(Suryanto et al., 2022), and ACTIVE (Suryanto et al., 2023). Both DAS and FCA are UV-map-based methods. They optimize the 3D texture by minimizing the attention-map-based scores and the detection scores of the detection model, respectively. Meanwhile, DTA and ACTIVE are both world-align-based methods. They optimize a square texture pattern with a neural network that can project the texture onto the target object, whereas ACTIVE uses a projection network that is closer to the world-aligned projection in UE4. We compare our results using the official carefully optimized textures generated by these methods. Given the similarity in experimental setups between our approach and DAS/FCA, as well as the generality of the textures generated by DTA and ACTIVE, we consider that the comparison is fair.

Evaluation metrics: To evaluate the NRP rendering component, Mean Absolute Error (MAE) is computed to quantify the difference between the output of NRP and the established ground truth, particularly within the vehicle region. Furthermore, we evaluate the attack effectiveness of the adversarial camouflage with the AP@0.5 (Everingham et al., 2015), a standard benchmark reflecting both the recall and precision value when the detection IOU threshold is 0.5.

Target detection models: Aligning with previous studies, we adopt YOLOv3 (Redmon & Farhadi, 2018) as the white-box target detection model for adversarial camouflage generation. To evaluate the effectiveness of the optimized camouflage, we utilize a suite of widely used object detection models, treated as black-box models except for YOLOv3. This suite includes YOLOX (Ge et al., 2021), Deformable DETR (DDTR) (Zhu et al., 2021), Dynamic R-CNN (DRCN) (Zhang et al., 2020), Sparse R-CNN (SRCN) (Sun et al., 2021), and Faster R-CNN (FrRCNN) (Ren et al., 2017), all of which are pretrained on the COCO dataset and implemented in MMDetection (Chen et al., 2019).

Training details: We utilize the Adam optimizer with a learning rate of 0.01 for NRP training and texture generation. We train the NRP over a span of 20 epochs and select the model exhibiting the best performance on the testing dataset. Aligning with Wang's study (2022), we configure the α and β (See Eq. 8) values at 1 and 0.0001, respectively. We directly obtain the mask of the vehicle from the semantic segmentation camera feature in CARLA (Dosovitskiy et al., 2017). During the adversarial camouflage generation phase, the camouflage texture is initialized randomly and trains with five epochs. We conduct experiments on a cluster with four NVIDIA RTX 3090 24GB GPUs.

4.2. Evaluation in Physically-Based Simulation Settings

In this section, we conduct a comparative analysis of RAUCA against current advanced adversarial camouflage attack methods, including DAS, FCA, DTA, and ACTIVE.

Table 1. Comparison of the effectiveness of camouflages across various object detection models. Values are AP@0.5 (%) of the car.

Methods	Single-stage			Two-stage		
	YOLOv3	YOLOX	DDTR	DRCN	SRCN	FrRCN
Normal	0.70	0.899	0.780	0.784	0.785	0.761
Random	0.623	0.812	0.581	0.670	0.659	0.652
DAS	0.638	0.875	0.707	0.709	0.710	0.710
FCA	0.555	0.795	0.555	0.655	0.619	0.654
DTA	0.507	0.692	0.341	0.592	0.461	0.534
ACTIVE	0.439	0.625	0.384	0.513	0.464	0.496
RAUCA	**0.304**	**0.611**	**0.285**	**0.449**	**0.384**	**0.406**

Robustness in multi-weather dataset: We run an extensive attack comparison using diverse detection models. In addition to YOLOv3, we also use various black-box detection models to evaluate the camouflage. The results are shown in Table 1, showing that RAUCA has the best performance on all models. DAS generally performs less effectively, primarily due to the limitations of partially painted camouflage. Meanwhile, FCA exhibits sub-optimal performance, only slightly better than random camouflage, because it is not capable of rendering sophisticated environment characteristics during texture generation. DTA and ACTIVE are also less effective than our method because they do not consider multi-weather conditions during texture generation. Besides, their texture projection is not consistent between generation and testing. Our method demonstrates a nearly 13% improvement in attack effectiveness compared to other models on the white-box YOLOv3 model. Additionally, it achieves an improvement of around 8% on most other black-box models. Figure 4 shows an example of different vehicle camouflage at night: our method is still effective, while other methods fail to attack.

Figure 6 shows the summarized performance of each camera pose and weather parameter; values are car AP@0.5 averaged from the detectors used in Table 1. We can see that the texture produced by our method outperforms other approaches in most camera transformations and weather conditions. Our method improves multi-view robustness over previous methods in most viewpoints, thanks to the UV mapping-based projection for our textures and effective rendering of the environmental characteristics in the foreground. In addition, due to the incorporation of a multi-weather dataset during camouflage generation, our method is also the most robust in different weather environments.

Transferability in unseen-weather dataset: We further evaluate the transferability of our camouflage under unseen weather conditions. As shown in Table 2, our method achieves the best attack performance on all types of detectors. Our method outperforms the previous state-of-the-art method, ACTIVE, by 23% on the white-box model YOLOv3, and approximately 11% on other black-box models. Our camouflage shows better attack performance under new weather conditions than under the weather conditions

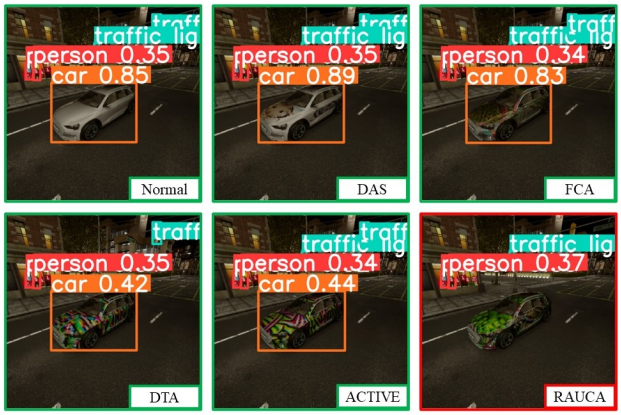

Figure 4. Attack comparison at night, a weather condition that has been included in our training set.

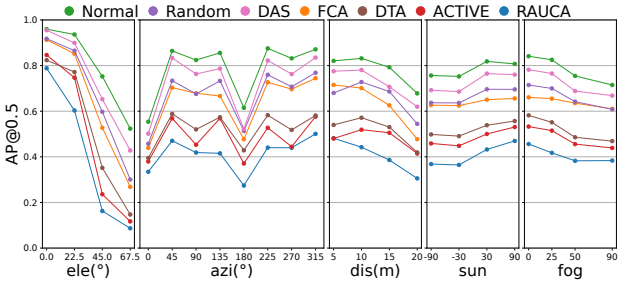

Figure 6. Attack comparison on different camera poses and weather parameters. "ele" denotes elevation, "azi" denotes azimuth, "dis" denotes distance, "fog" denotes fog density, and "sun" denotes sun altitude angle. Values are car AP@0.5 (%) averaged from all models.

Table 2. Attack comparison in unseen-weather dataset. Values are the car AP@0.5 (%).

Methods	Single-stage			Two-stage		
	YOLOv3	YOLOX	DDTR	DRCN	SRCN	FrRCN
Normal	0.814	0.923	0.815	0.816	0.807	0.813
Random	0.752	0.837	0.597	0.698	0.651	0.691
DAS	0.757	0.887	0.739	0.730	0.725	0.744
FCA	0.634	0.808	0.533	0.641	0.576	0.655
DTA	0.651	0.711	0.328	0.629	0.458	0.545
ACTIVE	0.594	0.643	0.372	0.533	0.446	0.513
RAUCA	**0.369**	**0.608**	**0.272**	**0.426**	**0.349**	**0.395**

used for texture generation. This highlights the transferability of our approach across diverse weather conditions. Such good transferability stems from the use of the multi-weather dataset with various fog levels and lighting conditions for texture optimization. The dataset includes 16 different weather conditions, including extreme conditions, such as dark light and dense fog. In the dense fog weather, the vehicle camouflage is heavily obscured but still effective against detectors. The weather in real life is typically not so extreme, so our camouflage can transfer well and achieve effective attacking performance in the unseen conditions.

Figure 5 shows an example of different camouflage on rainy days. Notice that rainy weather is excluded from texture generation. Even though the rain blurs part of the texture, RAUCA uniquely succeeds in attacking the target detector.

Figure 5. Attack comparison in the rainy day, a weather condition that hasn't appeared in our training set.

Table 3. Attack comparison under diverse real-world environmental conditions. Values are the car AP@0.5 (%) with YOLOv3.

Methods	Environmental condition				Total
	Noon	Afternoon	Night	Fog	
Normal	0.810	0.82	0.728	0.721	0.770
DAS	0.778	0.781	0.670	0.793	0.756
FCA	0.488	0.547	0.462	0.473	0.493
DTA	0.382	0.438	0.379	**0.158**	0.339
ACTIVE	0.253	0.264	0.242	0.163	0.231
RAUCA	**0.252**	**0.191**	**0.137**	0.200	**0.195**

4.3. Evaluation in Real-World Settings

In this section, we move our test to the real world. We conduct experiments with five types of camouflage in multi-light and foggy conditions to demonstrate the robustness of our camouflage in different physical environments.

Robustness in multi-light conditions: We change the lighting conditions by taking photos at different times of the day. We report the car AP@0.5 with YOLOv3. The results are shown in Table 3, which shows our method achieves the best attack performance in all light conditions. Although our camouflage has a similar attack effectiveness to ACTIVE at noon, it significantly outperforms all other methods, particularly under low-light conditions at night. We can see that almost all of the adversarial camouflage, except DAS, attack better in the real world than in the simulation world, probably because our camera viewpoint chosen in the real world is more susceptible to attacks. The first three rows of Figure 7 show the sample prediction results at different light conditions. Our method distinguishes itself as the only successful approach to deceive the target detector consistently across diverse lighting conditions in these examples.

Robustness in foggy conditions: We compare the effects of various attack textures in real-world fog scenarios, simulated

Figure 7. Real-world evaluation using five different types of camouflage in different environmental conditions.

Table 4. Comparing the impact $W(x_{ref})$ on NRP training. Values are Mean Absolute Error values between the output of NRP and ground truth within the car area.

METHODS	DISTANCE			
	5	10	15	20
WITHOUT $W(x_{ref})$	7.22	6.91	7.14	6.76
WITH $W(x_{ref})$	6.50	5.51	5.53	5.23

using a smoke machine. The car AP@0.5 for YOLOv3 detection is reported in the fifth column of Table 3. DTA, ACTIVE, and our method demonstrate increased effectiveness in attacking, with DTA and ACTIVE slightly surpassing our method. Given the challenge of maintaining consistent fog density, we acknowledge this discrepancy and consider it reasonable. Figure 7 provides examples of vehicle texture detection in foggy environments. It is evident that both DAS and FCA struggle to effectively attack the target detector, while DTA, ACTIVE, and RAUCA successfully induce misclassification in the target detector's output.

We average the detection results for all environmental cases, and the results are in the last column of Table 3, which shows that our method has the best attack results. It illustrates the robustness of our method in different physical environments.

4.4. Ablation Studies

Effectiveness of the $W(x_{ref})$ for NRP training: During the training of the NRP, we utilize $W(x_{ref})$, a weight function that deflates the loss according to the vehicle area in the image, to balance NRP's rendering optimization across various camera viewpoints. In this section, we evaluate the compact of $W(X_{ref})$ on the accuracy of NRP rendering. We report the MAE values between the output of NRP and ground truth within the car area. As shown in Table 4, the incorporation of $W(x_{ref})$ enhances the rendering capability across all camera distances. Notably, we observe a more pronounced improvement in rendering ability at longer camera distances, aligning with our intended design expectations.

Effectiveness of the multi-weather dataset and NRP for texture generation: We evaluate the effectiveness of incorporating multi-weather conditions into our dataset and our proposed rendering component, NRP, for texture generation. As can be seen in Table 5, we generate the camouflage according to the four combinations ways. The single-weather dataset in the table is identical to the multi-weather dataset, except that the weather parameter is fixed to the default value in Carla. We observe that the multi-weather dataset and the NRP rendering component are crucial to improving attacks' performance. Specifically, the NRP rendering component has the most significant impact on adversarial camouflage effects, with an improvement of 0.34% under the multi-weather dataset.

We also observe that when the framework employs NR as the renderer, introducing multi-weather conditions for texture generation results in a slightly diminished attack performance. This discrepancy may stem from NR's limitations in rendering comprehensive environmental characteristics. The weather information in the multi-weather dataset can't be effectively incorporated into the foreground; instead, it amplifies the contrast between the foreground and background. Consequently, the image obtained from the fusion of the front and rear backgrounds becomes less realistic, causing the texture optimization in the wrong direction.

Table 5. The effectiveness of the multi-weather dataset and NRP for texture generation. Values are car AP@0.5 (%) with YOLOv3.

TEX-GEN DATASET	RENDERER	
	NR	NRP
SINGLE-WEATHER	0.60	0.42
MULTI-WEATHER	0.64	**0.30**

Table 6. The effectiveness of the different adversarial losses for texture generation. Values are car AP@0.5 (%) with YOLOv3.

METHODS	SINGLE-STAGE			TWO-STAGE		
	YOLOv3	YOLOX	DDTR	DRCN	SRCN	FRRCN
FCA LOSS	0.331	0.620	0.404	**0.437**	0.479	0.453
ACTIVE LOSS	0.348	0.625	0.344	0.439	0.462	0.424
RAUCA LOSS	**0.304**	**0.611**	**0.285**	0.449	**0.384**	**0.406**

Effectiveness of our adversarial loss for texture generation: In this section, we evaluate the effectiveness of our designed adversarial loss function. We compare our adversarial loss with the ones used in FCA and ACTIVE. All these losses are calculated using the predicted detection boxes: the object confidence, class confidence, and IOU between the detection box and the ground truth. The difference between our designed loss function and FCA's loss function is that FCA's adversarial loss only considers the output detection box where the ground-truth center is located. In contrast, ours considers all of the detection boxes. It makes the camouflage to attack a broader range of detection boxes, preventing the target from being detected by boxes that do not contain the center of the ground truth. Compared with ACTIVE's, our loss function incorporates the IOU value into the calculation, which makes our camouflage optimization focus more on confusing the detection boxes with a large degree of intersection with the target. Moreover, unlike minimizing the maximum confidence score across all classes in ACTIVE's adversarial loss function, we minimize the maximum confidence score for the car class to achieve a more substantial attack effect. The comparing results are shown in Table 6. Values are the car AP@0.5. It can be seen that our proposed adversarial loss function achieves the strongest attack effect on most of the models.

Effectiveness of the smooth loss for texture generation: We investigate the impact of the smooth loss on the attack effectiveness of the generated camouflage. We vary the hyperparameter of the smooth loss β; the results are shown in Table 7. Values are the car AP@0.5. It can be seen that the overall attack effectiveness is better than the previous state-of-the-art ACTIVE method when β changes, demonstrating the effectiveness of our generated camouflage.

5. Conclusion

We have proposed RAUCA, a UV-map-based physical adversarial camouflage attack framework with a realistic neural renderer and incorporating multi-weather cases. In particular, we utilize our novel neural render component, namely NRP, which offers the advantages of precise texture mapping and the ability to render environment characteristics. Additionally, we incorporate a multi-weather dataset during camouflage generation to further enhance its robustness. Our experiments demonstrate that RAUCA outperforms the existing works under multi-weather situations, making it more robust in both simulation and physical world settings.

Table 7. The effectiveness of the smooth loss for texture generation. Values are car AP@0.5 (%) with YOLOv3.

METHODS	SINGLE-STAGE			TWO-STAGE		
	YOLOv3	YOLOX	DDTR	DRCN	SRCN	FRRCN
ACTIVE	0.439	0.625	0.384	0.513	0.464	0.496
β=0.0	0.343	0.619	0.307	0.454	0.419	0.431
β=0.0001	**0.304**	**0.611**	**0.285**	**0.449**	**0.384**	**0.406**
β=0.0003	0.341	0.626	0.337	0.460	0.404	0.427

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grants: 62306093, 62376074), the National Key R&D Program of China (Grant: 2021YFB2700900), the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (Grant: 2022B1212010005), and Shenzhen Science and Technology Program (Grants: JSGGKQTD20221101115655027, KCXST20221021111404010, JSGG20220831103400002, SGDX20230116091244004, KJZD20230923114405011, KJZD20231023095959002, RKX20231110090859012), the Fundamental Research Funds for the Central Universities (Grants: HIT.DZJJ.2023118, HIT.OCEF.2024047), and the Fok Ying Tung Education Foundation of China (Grant: 171058).

Impact Statement

Attackers might exploit our proposed adversarial camouflage method to attack autonomous driving and transportation systems. To remedy this, we suggest enhancing the current vehicle detection models via adversarial training with our camouflage, which can reinforce models against these potential attacks. Additionally, the generated adversarial camouflage can be used to test the models and ensure that the model is robust before deployment.

References

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In *International conference on machine learning*, pp. 284–293. PMLR, 2018.

Chai, J., Zeng, H., Li, A., and Ngai, E. W. Deep learning

in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6:100134, 2021.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., and Lin, D. MMDetection: Open MMLab Detection Toolbox and Benchmark. *CoRR*, abs/1906.07155, 2019. URL http://arxiv.org/abs/1906.07155.

Chen, S., Cornelius, C., Martin, J., and Chau, D. H. P. ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector. In Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., and Ifrim, G. (eds.), *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part I*, volume 11051 of *Lecture Notes in Computer Science*, pp. 52–68. Springer, 2018. doi: 10.1007/978-3-030-10925-7_4. URL https://doi.org/10.1007/978-3-030-10925-7_4.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. CARLA: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. YOLOX: Exceeding YOLO Series in 2021. *CoRR*, abs/2107.08430, 2021. URL https://arxiv.org/abs/2107.08430.

Kato, H., Ushiku, Y., and Harada, T. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3907–3916, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Mahendran, A. and Vedaldi, A. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 5188–5196. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7299155. URL https://doi.org/10.1109/CVPR.2015.7299155.

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

Ren, S., He, K., Girshick, R. B., and Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. doi: 10.1109/TPAMI.2016.2577031. URL https://doi.org/10.1109/TPAMI.2016.2577031.

Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In Weippl, E. R., Katzenbeisser, S., Kruegel, C., Myers, A. C., and Halevi, S. (eds.), *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 1528–1540. ACM, 2016a. doi: 10.1145/2976749.2978392. URL https://doi.org/10.1145/2976749.2978392.

Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In Weippl, E. R., Katzenbeisser, S., Kruegel, C., Myers, A. C., and Halevi, S. (eds.), *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 1528–1540. ACM, 2016b. doi: 10.1145/2976749.2978392. URL https://doi.org/10.1145/2976749.2978392.

Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14454–14463, 2021.

Suryanto, N., Kim, Y., Kang, H., Larasati, H. T., Yun, Y., Le, T.-T.-H., Yang, H., Oh, S.-Y., and Kim, H. DTA: Physical Camouflage Attacks Using Differentiable Transformation Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15305–15314, June 2022.

Suryanto, N., Kim, Y., Larasati, H. T., Kang, H., Le, T.-T.-H., Hong, Y., Yang, H., Oh, S.-Y., and Kim, H. ACTIVE: Towards Highly Transferable 3D Physical Camouflage for Universal and Robust Vehicle Evasion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4305–4314, October 2023.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for

real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7464–7475, 2023.

Wang, D., Jiang, T., Sun, J., Zhou, W., Gong, Z., Zhang, X., Yao, W., and Chen, X. Fca: Learning a 3d full-coverage vehicle camouflage for multi-view physical adversarial attack. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 2414–2422, 2022.

Wang, J., Liu, A., Yin, Z., Liu, S., Tang, S., and Liu, X. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8565–8574, 2021.

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

Zhang, H., Chang, H., Ma, B., Wang, N., and Chen, X. Dynamic R-CNN: Towards high quality object detection via dynamic training. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pp. 260–275. Springer, 2020.

Zhang, Y., Foroosh, P. H., and Gong, B. Camou: Learning a vehicle camouflage for physical adversarial attack on object detections in the wild. *ICLR*, 2019.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=gZ9hCDWe6ke.

A. Datasets

This section presents the parameters of all of the datasets in our experiments.

For NRP model training and testing: We collect vehicle datasets in the CARLA simulation environment under 16 weather conditions. These weather conditions are generated by combining four sun altitude angles (-90°, -30°, 30°, 90°) with four fog densities (0, 25, 50, 90). Within each weather scenario, we randomly choose 30 locations for model training and 3 for testing. For each car location, imagery is acquired at two azimuth angles (0°, 45°), two altitude angles (22.5°, 67.5°), and four distances (5m, 10m, 15m, 20m). In terms of color paint for the vehicle, nine vehicle colors are utilized during training(red, green, blue, magenta, yellow, cyan, white, and black and grey), and 64 vehicle colors (RGB values are taken from [0, 85, 170, 255] respectively) for testing. In summary, we employ 69,120 photo-realistic images for model training (consisting of 16 weather conditions $\times$ 30 car locations $\times$ 16 camera poses $\times$ 9 colors) and 49152 for testing (consisting of 16 weather conditions $\times$ 3 car locations $\times$ 16 camera poses $\times$ 64 colors)

For texture generation and evaluation in physically-based simulation settings: For the generation of adversarial camouflage, we choose the same 16 weather combinations as NRP training. For evaluation in physically-based simulation, we produce two datasets: one with the same weather combinations as the training set and the other featuring 16 novel weather conditions that are not seen during the texture generation, including diverse scenarios like MidRainyNight and DustStorm. We use the former dataset for robust evaluation and the latter for transferability evaluation. Within each weather scenario, we randomly choose 20 locations for texture generation and 4 distinct locations for each test dataset. For each car transformation, imagery is captured at every 45-degree increment in azimuth angle, spanning four distinct altitude angles (0.0°, 22.5°, 45.0°, 67.5°) and four varying distances (5m, 10m, 15m, 20 m) for each. In summary, we employ 40,960 photo-realistic images for model training (consisting of 16 weather conditions $\times$ 20 car locations $\times$ 128 camera poses $\times$ 1 colors) and 7593 for each test dataset (consisting of 16 weather conditions $\times$ 4 car locations $\times$ 128 camera poses $\times$ 1 colors and obliterating the images of the car entirely obscured by the obstacle).

For evaluation in the real world settings: We print our texture and adversarial texture from DAS (Wang et al., 2021), FCA (Wang et al., 2022), DTA (Suryanto et al., 2022) as well as ACTIVE (Suryanto et al., 2023) and stick them on official 1:12 Audi E-Tron vehicle models. For the adversarial texture generated by the UV-map-based method (DAS, FCA, and RAUCA), we print it and paste it on the vehicle directly according to its corresponding position since it is in the form of a UV map. For the world-align-based method (DTN and ACTIVE), since it generates a 2D square texture pattern, it cannot be directly pasted like the UV-map-based camouflage. Hence, we adopt the implementation from previous methods: repeating and enlarging the pattern, printing it, and applying it to the entire vehicle. In addition to this, we also prepare an unpainted vehicle. We shoot these models in the real world with the Xiaomi 14 for comparison. To demonstrate the robustness of our camouflage in physical multi-light conditions, we capture photos at different times to change the light condition. We take 2592 pictures of six cars with poses of 3 distances (5cm, 10cm, 15cm), 8 azimuths per 45°, and 3 altitude angles (0°, 30°, 60°), as well as at two locations (an asphalt road and an open square) at noon, in the afternoon, and at night . Moreover, we evaluate textures' effectiveness in the presence of fog in the real world. We use a smoke machine to simulate the foggy environment on the imitated road on the desktop. We capture pictures of six cars at every 45-degree increment in azimuth angle, 3 altitude angles (0°, 30°, 60°). We take five consecutive pictures for the same camera angle with each car to reduce the unfairness caused by the randomness of the fog. Finally, we obtained 720 pictures for comparison.