Physical Adversarial Camouflage Through Gradient Calibration and Regularization

Jiawei Liang[1,2] , **Siyuan Liang**[3*] , **Jianjie Huang**[1] , **Chenxi Si**[1] , **Ming Zhang**[4] , **Xiaochun Cao**[1,2*]
[1]School of Cyber Science and Technology, Sun Yat-sen University Shenzhen Campus, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]Nanyang Technological University, Singapore
[4]National Key Laboratory of Science and Technology on Information System Security, Beijing, China
{liangjw57, huangjj67, sichx}@mail2.sysu.edu.cn, siyuan.liang@ntu.edu.sg,
zm_stiss@163.com, caoxiaochun@mail.sysu.edu.cn

Abstract

The advancement of deep object detectors has greatly affected safety-critical fields like autonomous driving. However, physical adversarial camouflage poses a significant security risk by altering object textures to deceive detectors. Existing techniques struggle with variable physical environments, facing two main challenges: 1) inconsistent sampling point densities across distances hinder the gradient optimization from ensuring local continuity, and 2) updating texture gradients from multiple angles causes conflicts, reducing optimization stability and attack effectiveness. To address these issues, we propose a novel adversarial camouflage framework based on gradient optimization. First, we introduce a gradient calibration strategy, which ensures consistent gradient updates across distances by propagating gradients from sparsely to unsampled texture points, thereby expanding the attack's effective range. Additionally, we develop a gradient decorrelation method, which prioritizes and orthogonalizes gradients based on loss values, enhancing stability and effectiveness in multi-angle optimization by eliminating redundant or conflicting updates. Extensive experimental results on various detection models, angles, and distances show that our method significantly surpasses the state-of-the-art, with an average attack success rate (ASR) increase of 13.46% across distances and 11.03% across angles. Furthermore, experiments in real-world settings confirm the method's threat potential, highlighting the urgent need for more robust autopilot systems less prone to spoofing.

1 Introduction

Object detection is essential for applications like autonomous driving, where robustness and reliability are crucial due to its integration into safety-critical systems. Adversarial attacks pose a serious threat by manipulating object textures to deceive detection algorithms. This can lead to misclassifications or undetected objects, resulting in potential severe real-world consequences like accidents.

Physical adversarial camouflage presents significant challenges compared to its digital counterparts, as it requires modifying real-world objects in a manner that ensures the attack remains effective when transitioning from the digital domain to uncontrolled physical environments. This process involves optimizing an object's surface texture to deceive detection systems across varying angles and distances, aligning with the inherent variability in how cameras perceive objects under diverse perspectives. However, achieving robust adversarial camouflage necessitates addressing several critical challenges. One key difficulty stems from the localized nature of texture manipulation. Specifically, from any given viewpoint, only the visible portions of the object's texture can be optimized. This viewpoint-specific focus often neglects the broader context of the object's surface, which may lead to suboptimal performance when interactions across different perspectives are considered holistically. Furthermore, inconsistencies in the object's appearance, caused by variations in viewing distance, angle, or environmental conditions, further compound the complexity of this task.

Existing methods typically utilize differentiable renderers to optimize adversarial textures by simulating object appearances across diverse viewpoints and environmental conditions. However, we identify two critical issues within this optimization pipeline that adversely affect attack performance. The first issue arises from distance-dependent sampling density: as the distance between the camera and the object changes, the number of pixels used to render the object varies, as illustrated in Figure 1. This variation leads to inconsistent gradient sparsity during backpropagation. This inconsistency, in turn, results in uneven texture modifications across different distances. The second issue stems from potential redundancy or conflict among gradient directions within a minibatch, as illustrated in Figure 2. Gradients derived from similar viewpoints tend to generate redundant updates, while gradients from distinct viewpoints may contradict and cancel one another, thus impeding effective texture optimization.

To address these identified challenges, we propose a novel adversarial camouflage framework built upon two core strategies: Nearest Gradient Calibration (NGC) and Loss-

[*]Corresponding author.

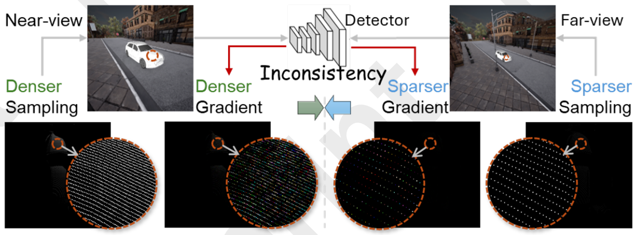


Figure 1: Illustration of gradient inconsistencies due to variations in sampling density across different distances.

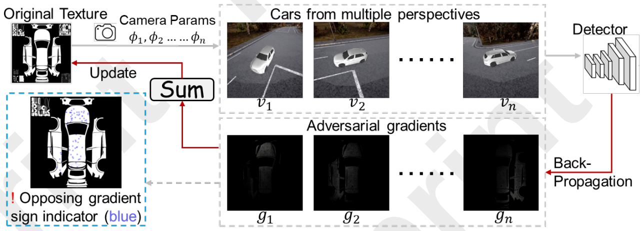


Figure 2: Illustration of gradient conflicts from different perspectives

Prioritized Gradient Decorrelation (LPGD). First, to mitigate the issue of inconsistent gradient sparsity, NGC propagates gradients from sampled points to neighboring unsampled points on the same surface, ensuring local continuity and consistent sparsity in texture updates across varying distances. Second, to resolve the gradient redundancy and conflicts arising from diverse viewpoints within a minibatch, LPGD prioritizes gradients according to their associated adversarial loss and decorrelates them through orthogonalization. This orthogonalization process ensures that each prioritized gradient provides complementary and non-conflicting information, thereby overcoming challenges related to redundancy and cancellation.

We conducted experiments in realistic simulated environments, demonstrating that our adversarial camouflage effectively evades object detection across diverse viewpoints, distances, weather conditions, and multiple detectors. It outperforms state-of-the-art methods by 13.46% in attack effectiveness across distances and 11.03% across angles. Real-world experiments further confirm its practicality and robustness.

Our contributions are as follows:

- We identify two challenges in optimizing adversarial camouflage, including inconsistent gradient sparsity due to distance-dependent sampling density, and redundant or conflicting gradients across viewpoints.
- We propose Nearest Gradient Calibration (NGC) to propagate gradients from sampled to unsampled texture points, ensuring local continuity in gradient updates across varying distances.
- We introduce Loss-Prioritized Gradient Decorrelation (LPGD) to prioritize and decorrelate gradients via orthogonalization, resolving redundancy and conflicts across viewpoints.
- Extensive experiments demonstrate that our approach achieves superior attack performance, enhanced robustness to environmental variations, and improved transferability across diverse object detectors compared to state-of-the-art methods.

2 Related Work

2.1 Object Detection

Object detection plays a pivotal role in computer vision, evolving through various strategies. Anchor-based methods include two-stage detectors [Girshick, 2015; He *et al.*, 2017; Ren *et al.*, 2015], which use region proposals for classification and bounding box regression, and single-stage models [Liu *et al.*, 2016; Redmon and Farhadi, 2018], which frame detection as a regression problem to achieve real-time performance. Conversely, anchor-free methods [Duan *et al.*, 2019] eliminate predefined anchors by directly predicting object centers and dimensions. Recent developments like DETR [Zhu *et al.*, 2021] introduce an end-to-end detection approach utilizing transformer architectures. Nevertheless, these methods are susceptible to adversarial attacks.

2.2 Adversarial Attacks

Adversarial attacks involve the deliberate alteration of images to mislead deep neural networks [Szegedy *et al.*, 2014], resulting in incorrect predictions. These attacks are divided into digital and physical based on the domain of perturbation. Digital attacks directly apply crafted perturbations to digital images, with techniques such as FGSM [Goodfellow *et al.*, 2014] and PGD [Madry *et al.*, 2017] demonstrating notable success, especially in classification. These strategies have been adapted for object detection but lose effectiveness in real-world scenarios. In contrast, physical attacks [Wei *et al.*, 2024; Zhu *et al.*, 2023] introduce perturbations in the real world, encompassing adversarial patches and camouflage. Patch-based attacks apply perturbations to flat surfaces, effective only from certain viewpoints, while adversarial camouflage covers entire object surfaces, allowing for multi-angle attacks.

2.3 Adversarial Camouflage

Current adversarial camouflage methods often utilize differentiable 3D renderers to project 3D objects as rendered images from multiple angles, optimizing 3D textures end-to-end via backpropagation. For instance, Wang et al. [Wang *et al.*, 2022] introduced Full-coverage Camouflage Attack (FCA), optimizing a vehicle's UV textures through a neural renderer for enhanced multi-viewpoint camouflage. Similarly, the Dual Attention Suppression (DAS) attack by Wang et al. [Wang *et al.*, 2021a] reduces visibility to both models and human observers. Another approach involves optimizing 2D texture patterns projected onto vehicle surfaces. Suryanto et al. [Suryanto *et al.*, 2022] proposed the Differentiable Transformer Attack (DTA), employing a neural renderer to simulate realistic effects like shadows. Additionally, ACTIVE [Suryanto *et al.*, 2023] improves camouflage aesthetics with enhanced texture mapping and background color integration. Furthermore, RAUCA [Zhou *et al.*, 2024] utilizes

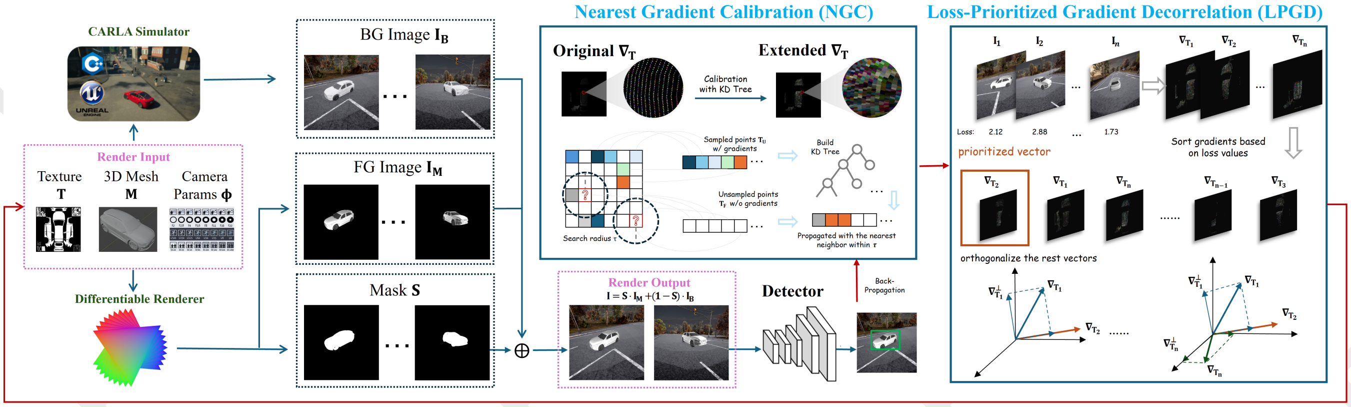


Figure 3: Overview of our proposed method. To optimize the texture, we first render the target vehicle image using the texture with a renderer and integrate it into a realistic background generated by the CARLA Simulator. During optimization, we employ NGC to calibrate gradients in sparsely sampled regions and use LPGD to resolve gradient conflicts through loss-prioritized orthogonalization on a minibatch. The refined gradients are then summed and utilized to update the texture.

advanced rendering to account for environmental factors such as diverse weather conditions. However, existing techniques may not fully resolve challenges arising from sampling inconsistency and conflicts in gradient updates, which can affect overall attack effectiveness.

3 Method

3.1 Problem Definitions

Adversarial camouflage [Wei *et al.*, 2024] aims to optimize the texture of a target object to achieve objectives such as evading detection or inducing misclassification. A 3D object is represented by the tuple $(\mathbf{M}, \mathbf{T})$, where $\mathbf{M}$ denotes the mesh structure and $\mathbf{T}$ represents its texture. The object's image, rendered under various camera parameters $\varphi \in \mathbf{\Phi}$ (*e.g.*, different camera angles and distances), is produced using a differentiable renderer, $\mathcal{R}$. The rendering process first computes the UV coordinates as a function of the camera parameter φ, defined by:

$$\mathbf{UV}(\varphi) = \mathcal{R}(\mathbf{M}, \varphi). \tag{1}$$

The final image of the target object is obtained by sampling color values from the texture $\mathbf{T}$ at the computed UV coordinates:

$$\mathbf{I_M}(\mathbf{T}, \varphi) = \mathcal{F}_S(\mathbf{UV}(\varphi), \mathbf{T}), \tag{2}$$

where $\mathcal{F}_S$ denotes the sampling function. Since the renderer $\mathcal{R}$ does not incorporate background information, a common approach for creating physically realistic images is to overlay the rendered object onto authentic backgrounds. In this study, we utilize background images generated by the CARLA simulator [Dosovitskiy *et al.*, 2017]. The composite image is created by replacing the background in the rendered image using a segmentation mask $\mathbf{S}$:

$$\mathbf{I}(\mathbf{T}, \varphi) = (\mathbf{S} \cdot \mathbf{I_M}(\mathbf{T}, \varphi)) + (1 - \mathbf{S}) \cdot \mathbf{I_B}, \tag{3}$$

where $\mathbf{I_B}$ represents the background image.

To conduct a physical evasion attack, the texture is optimized to mislead the target model, causing detection failure or misclassification. The adversarial texture is derived by solving:

$$\mathbf{T}_{\text{adv}} = \arg\min_{\mathbf{T}} \mathcal{L}\big(\mathbb{F}_\theta(\mathbf{I}(\mathbf{T}, \varphi)),\ y\big), \tag{4}$$

where $\mathbb{F}_\theta$ is the object detector with parameters θ, y is the ground-truth label covering classification and localization, and $\mathcal{L}$ is the suppression loss function. A lower loss implies a reduced probability of predicting the target label correctly. The objective is to hinder the model's ability to accurately predict the target label y.

Existing Obstacles. Achieving effective adversarial camouflage in multi-view settings requires accounting for variations in distance and viewing angles. Current methods [Wang *et al.*, 2022; Zhou *et al.*, 2024] optimize textures by rendering images under diverse conditions using differentiable renderers [Kato *et al.*, 2018]. However, they fail to address two significant challenges: ❶ Varying sampling densities at different distances. This variation causes the gradients used for texture updates to exhibit inconsistent levels of sparsity across distances. ❷ Conflicting texture update directions across different viewing angles. Naively merging these updates undermines the effectiveness of the attack.

3.2 Nearest Gradient Calibration

As illustrated in Equation 2, the rendered image samples colors from the texture $\mathbf{T}$ using UV coordinates. The pixel count representing the target object varies with rendering distance: closer objects occupy more pixels, resulting in denser texture sampling, whereas distant objects are represented by fewer pixels, leading to sparser sampling. Typically, only sampled points are updated during texture optimization. Due to varying rendering distances producing different sets of sampled points, inconsistencies in texture updates can emerge across different distances. This issue arises from emphasizing pixel-level optimization while neglecting the continuity required to preserve the integrity of local surface regions.

To address this problem, we propose a simple yet effective method called **Nearest Gradient Calibration** (NGC). The core idea of NGC is to extend the gradient of each sampled point to its neighboring unsampled points. This propagation ensures that gradient updates over the texture remain locally continuous, thereby preserving the consistency of texture updates across varying distances.

We define the trainable region of the texture $\mathbf{T}$ as $\mathbf{T}'$, which is constrained by a mask $\mathbf{K}$ that restricts updates to specific areas, such as the visible outer surface of a vehicle. Within $\mathbf{T}'$, the sampled subset of points is denoted as $\mathbf{T}_{\mathrm{U}}(\varphi)$, determined by the camera parameters φ. For simplicity, we refer to $\mathbf{T}_{\mathrm{U}}(\varphi)$ as $\mathbf{T}_{\mathrm{U}}$ in the following sections. The remaining unsampled points are defined as $\mathbf{T}_{\mathrm{F}} = \mathbf{T}' \setminus \mathbf{T}_{\mathrm{U}}$. Formally, this relationship is expressed as:

$$\mathbf{T}' = \mathbf{T} \odot \mathbf{K} = \mathbf{T}_{\mathrm{U}} \cup \mathbf{T}_{\mathrm{F}} \tag{5}$$

After each iteration of backpropagation, only the gradients of the sampled points, $\nabla_{\mathbf{T}_{\mathrm{U}}}$, are computed.

$$\nabla_{\mathbf{T}_{\mathrm{U}}} = \nabla_{\mathbf{T}_{\mathrm{U}}} \mathcal{L}\big(\mathbb{F}_\theta(\mathbf{I}(\mathbf{T}, \varphi)),\ y\big) \tag{6}$$

Ideally, adjacent points on the same surface should be updated collectively to maintain continuity. However, the renderer's sampling mechanism may result in sparsely distributed gradients, leaving intermediate points without gradients. To address this, we propagate the sparse gradients to their neighboring unsampled points using a nearest neighbor search, implemented via the *KD Tree* algorithm [Zhou *et al.*, 2008]. For each unsampled point $p \in \mathbf{T}_{\mathrm{F}}$, we calculate the Euclidean distance to all points in $\mathbf{T}_{\mathrm{U}}$ to identify the nearest neighbor q. The gradient of the nearest sampled point $q \in \mathbf{T}_{\mathrm{U}}$ is assigned to p provided that $\|p - q\|_2 \leq \tau$, where τ is the search radius. This ensures that gradients are assigned only to unsampled points within a specified local range, maintaining spatial locality. If no sampled point within the threshold is found, the gradient for p remains zero. Formally, the gradient assignment is defined as:

$$\forall p \in \mathbf{T}_{\mathrm{F}}, \quad \nabla^{p}_{\mathbf{T}_{\mathrm{F}}} = \begin{cases} \nabla^{q}_{\mathbf{T}_{\mathrm{U}}}, & \text{if } \|p - q\|_2 \leq \tau, \\ 0, & \text{otherwise,} \end{cases} \\ \text{where } q = \arg \min_{q' \in \mathbf{T}_{\mathrm{U}}} \|p - q'\|_2. \tag{7}$$

After assigning gradients to $\mathbf{T}_{\mathrm{F}}$, we obtain the final extended gradient $\nabla_{\mathbf{T}'}$. By applying NGC, gradients are smoothly propagated across the texture surface, ensuring local continuity, as illustrated in Figure 3. This approach effectively addresses the inconsistencies caused by varying sampling densities, resulting in more uniform and coherent texture updates across different rendering distances. The detailed procedure for this method is summarized in Algorithm 1.

3.3 Loss-Prioritized Gradient Decorrelation

In multi-view physical adversarial optimization, overlapping texture regions observed from varying azimuth and elevation angles present substantial challenges, primarily due to gradient redundancy and conflicts. Gradient redundancy arises when gradients from different viewpoints exhibit similar update directions, leading to inflated gradient norms that destabilize optimization. Conversely, gradient conflicts occur when gradients act in opposing directions, resulting in partial or complete cancellation of updates and diminishing their effectiveness. If unaddressed, these issues can potentially reduce efficiency of adversarial optimization.

Algorithm 1: Nearest Gradient Calibration (NGC)

```
Input  : Texture $\mathbf{T}$, Mask $\mathbf{K}$, camera parameter set $\Phi$,
         Search radius $\tau$
Output: Extended gradients $\nabla_{\mathbf{T}'}$
Define $\mathbf{T}' = \mathbf{T} \odot \mathbf{K}$;
Initialize KD-Tree $\mathcal{K}$ with $\mathbf{T}_{\mathrm{U}}$;
for each optimization step do
    Sample camera parameter $\varphi \in \Phi$;
    Identify sampled points $\mathbf{T}_{\mathrm{U}}(\varphi)$;
    $\mathbf{T}_{\mathrm{F}} \leftarrow \mathbf{T}' \setminus \mathbf{T}_{\mathrm{U}}$;
    Compute gradients $\nabla_{\mathbf{T}_{\mathrm{U}}}$;
    Query nearest neighbors for $p \in \mathbf{T}_{\mathrm{F}}$ using $\mathcal{K}$;
    for each $p \in \mathbf{T}_F$ do
        Retrieve nearest $q$ and distance $d$;
        if $d \leq \tau$ then
            Assign $\nabla_{\mathbf{T}_{\mathrm{F}}}(p) \leftarrow \nabla_{\mathbf{T}_{\mathrm{U}}}(q)$;
        else
            Assign zero gradient: $\nabla_{\mathbf{T}_{\mathrm{F}}}(p) \leftarrow 0$;
        end
    end
end
return $\nabla_{\mathbf{T}'}$;
```

To address these problems, we propose the **Loss-Prioritized Gradient Decorrelation** (LPGD) method, which integrates gradient orthogonalization with prioritization based on loss values. This method ensures that the optimization focuses on the most challenging viewpoints and resolves redundancy and conflicts among gradients to produce stable and efficient updates.

In the LPGD method, gradients from k viewpoints, corresponding to the trainable part of the texture $\mathbf{T}'$ in Equation 5, are first sorted according to their loss values, $\mathcal{L}(\varphi_i)$, such that $\mathcal{L}(\varphi_1) \geq \mathcal{L}(\varphi_2) \geq \cdots \geq \mathcal{L}(\varphi_k)$. This prioritization ensures that gradients from more challenging viewpoints are processed first. Specifically, the gradients $\{\nabla_{\mathbf{T}'}(\varphi_1), \nabla_{\mathbf{T}'}(\varphi_2), \ldots, \nabla_{\mathbf{T}'}(\varphi_k)\}$ are ordered in this manner before applying gradient orthogonalization.

Once gradients are prioritized, we proceed with gradient orthogonalization to resolve redundancy and conflicts. This step is crucial to ensure that each gradient contributes unique, non-redundant information to the optimization process. We apply a Schmidt orthogonalization [Leon *et al.*, 2013] procedure, which projects each gradient onto the orthogonal complement of the subspace spanned by the previously processed gradients. Formally, given the i-th gradient $\nabla_{\mathbf{T}'}(\varphi_i)$, its orthogonalized counterpart $\nabla^{\perp}_{\mathbf{T}'}(\varphi_i)$ is computed as:

$$\nabla^{\perp}_{\mathbf{T}'}(\varphi_i) = \nabla_{\mathbf{T}'}(\varphi_i) - \sum_{j=1}^{i-1} \alpha_{ij} \nabla^{\perp}_{\mathbf{T}'}(\varphi_j), \\ \text{where } \alpha_{ij} = \frac{\nabla_{\mathbf{T}'}(\varphi_i) \cdot \nabla^{\perp}_{\mathbf{T}'}(\varphi_j)}{\|\nabla^{\perp}_{\mathbf{T}'}(\varphi_j)\|^2}. \tag{8}$$

This recursive projection removes any components of

Algorithm 2: Loss-Prioritized Gradient Decorrelation (LPGD)

```
Input  : Viewpoints $\{\varphi_i\}_{i=1}^{k}$, corresponding gradients
         $\{\nabla_{\mathbf{T}'}(\varphi_i)\}_{i=1}^{k}$, and losses $\{\mathcal{L}(\varphi_i)\}_{i=1}^{k}$
Output: Optimized gradient $\nabla^{*}_{\mathbf{T}'}$
Sort $\nabla_{\mathbf{T}'}(\varphi_i)$ in descending order of $\mathcal{L}(\varphi_i)$;
Initialize $\nabla^{\perp}_{\mathbf{T}'}(\varphi_1) \leftarrow \nabla_{\mathbf{T}'}(\varphi_1)$;
for each $i = 2$ to $k$ do
    for each $j = 1$ to $i - 1$ do
        Compute $\alpha_{ij} = \frac{\nabla_{\mathbf{T}'}(\varphi_i) \cdot \nabla^{\perp}_{\mathbf{T}'}(\varphi_j)}{\|\nabla^{\perp}_{\mathbf{T}'}(\varphi_j)\|^2}$
    end
    Update
     $\nabla^{\perp}_{\mathbf{T}'}(\varphi_i) \leftarrow \nabla_{\mathbf{T}'}(\varphi_i) - \sum_{j=1}^{i-1} \alpha_{ij} \nabla^{\perp}_{\mathbf{T}'}(\varphi_j)$
end
Compute $\nabla^{*}_{\mathbf{T}'} \leftarrow \frac{1}{k} \sum_{i=1}^{k} \nabla^{\perp}_{\mathbf{T}'}(\varphi_i)$
return $\nabla^{*}_{\mathbf{T}'}$;
```

$\nabla_{\mathbf{T}'}(\varphi_i)$ that are redundant or conflicting with respect to previously orthogonalized gradients, ensuring that all gradients are mutually decorrelated. As a result, the gradients provide complementary information about the texture updates, rather than conflicting or overlapping contributions.

Once all gradients are orthogonalized, the final update direction is determined by averaging the orthogonalized gradients. The final update direction is computed as:

$$\nabla^{*}_{\mathbf{T}'} = \frac{1}{k} \sum_{i=1}^{k} \nabla^{\perp}_{\mathbf{T}'}(\varphi_i), \tag{9}$$

where $\nabla^{\perp}_{\mathbf{T}'}(\varphi_i)$ are the orthogonalized gradients. By combining orthogonalization with averaging, the method achieves stable updates that respect the contributions of all viewpoints without being dominated by redundant or conflicting gradients. The detailed procedure for this method is summarized in Algorithm 2.

Notably, our proposed NGC and LPGD methods are compatible and can be combined in a straightforward manner. NGC propagates the gradient of points $q \in \mathbf{T}_{\mathrm{U}}$, i.e., $\nabla_{\mathbf{T}_{\mathrm{U}}}$, to points $p \in \mathbf{T}_{\mathrm{F}}$, i.e., $\nabla_{\mathbf{T}_{\mathrm{F}}}$, to obtain the extended gradient $\nabla_{\mathbf{T}'}$ at each iteration. Subsequently, LPGD decorrelates these extended gradients $\nabla_{\mathbf{T}'}$ within a minibatch via loss-prioritized orthogonalization to obtain the final gradients $\nabla^{*}_{\mathbf{T}'}$.

4 Experiment

4.1 Settings

Implementation Details. We utilize the CARLA simulator [Dosovitskiy *et al.*, 2017] to generate datasets. In line with previous studies [Zhou *et al.*, 2024], we capture simulated images to construct a comprehensive training set. Our training set comprises 20,000 images captured from diverse angles and distances to enhance texture generation. We focus on the Audi E-Tron model, as explored in prior research [Wang *et al.*, 2021a; Wang *et al.*, 2022; Suryanto *et al.*, 2022; Suryanto *et al.*, 2023]. For evaluation, adversarial camouflage is applied to the vehicle within CARLA, with images captured at elevation angles of $\{0°, 5°, 10°, 15°, 20°, 30°, 45°, 60°\}$, along with 2-degree azimuth increments for a thorough 360° sweep. Extended evaluations encompass distances of $\{5, 7.5, 10, 12.5, 15\}$ meters and five distinct weather conditions: noon, sunset, night, foggy, and rainy. In real-world tests, the camouflage patterns are printed on 1:24 scale Audi E-Tron models, with images captured from various angles and distances for further analysis. Texture optimization is conducted using the Adam optimizer with a learning rate of 0.1, employing ModernGL [Dombi, 2020] for differentiable rendering and segmentation mask $\mathbf{S}$ generation. We optimize the textures over three epochs, with all experiments performed on a single NVIDIA A100 80GB GPU.

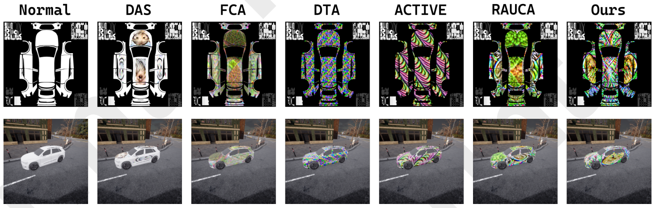


Figure 4: Optimized textures for each baseline method and the corresponding rendered samples.

Comparison Baselines. Our framework is evaluated against several state-of-the-art adversarial camouflage techniques, including DAS [Wang *et al.*, 2021a], FCA [Wang *et al.*, 2022], DTA [Suryanto *et al.*, 2022], ACTIVE [Suryanto *et al.*, 2023], and RAUCA [Zhou *et al.*, 2024]. For our evaluations, as illustrated in Figure 4, we transform both our optimized textures and those of the baselines into standardized UV textures, which are constrained by a mask $\mathbf{K}$ applied to the optimizable regions.

Evaluation Metrics. Following previous studies [Zhou *et al.*, 2024], we assess the attack effectiveness of adversarial camouflage using the AP@0.5 benchmark [Everingham *et al.*, 2015], which is a standard measure capturing both recall and precision at a detection IOU threshold of 0.5.

Target Detection Models. Consistent with prior studies [Zhou *et al.*, 2024], we utilize YOLOv3 [Redmon and Farhadi, 2018] as the white-box target detection model for generating adversarial camouflage. To evaluate the effectiveness of the optimized camouflage, we test it against a range of popular object detection models, treating them as black-box models, except for YOLOv3. This evaluation encompasses models such as the one-stage detector YOLOX [Ge *et al.*, 2021], two-stage detectors Faster R-CNN (FrRCN) [Ren *et al.*, 2016] and Mask R-CNN (MkRCN) [He *et al.*, 2017], as well as transformer-based detectors DETR [Carion *et al.*, 2020] and PVT [Wang *et al.*, 2021b]. Each model is pretrained on the COCO dataset and implemented using the MMDetection framework [Chen *et al.*, 2019].

4.2 Evaluation in Physical Simulation Settings

In this section, we compare our method to state-of-the-art (SOTA) adversarial camouflage approaches. Comprehensive

Methods	**Elevation angle**								Avg
	$\mathbf{0°}$	$\mathbf{5°}$	$\mathbf{10°}$	$\mathbf{15°}$	$\mathbf{20°}$	$\mathbf{30°}$	$\mathbf{45°}$	$\mathbf{60°}$	
Normal	97.57	99.23	99.56	98.23	99.78	96.24	85.42	45.30	90.17
DAS	93.37	95.58	99.45	93.37	98.34	95.58	73.48	45.30	86.81
FCA	93.37	73.48	64.09	65.75	70.17	39.78	10.50	3.31	52.56
DTA	76.80	64.09	60.22	72.38	82.87	56.91	22.10	0.55	54.49
ACTIVE	65.19	66.30	36.46	32.60	37.57	19.34	0.0	0.0	32.18
RAUCA	38.67	23.20	4.97	9.94	13.26	13.81	1.66	0	13.19
Ours	**12.53**	**2.51**	**2.24**	**0.0**	**0**	**0**	**0**	**0**	**2.16**

Table 1: Evaluation results in various elevation angles. Values are AP@0.5 (%) of the target vehicle averaged across different azimuth angles with YOLOv3.

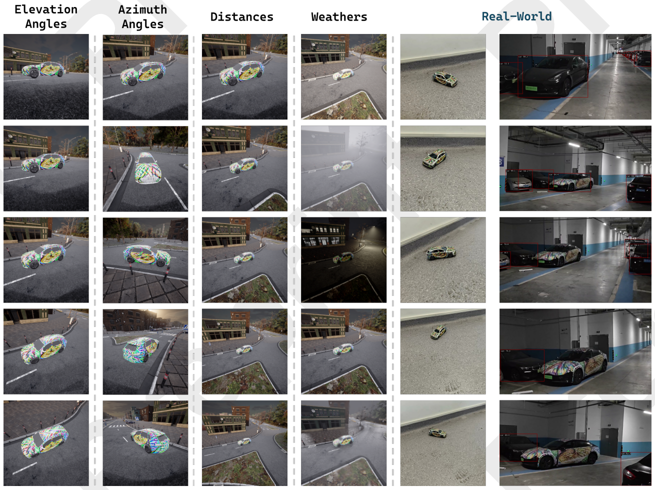


Figure 5: Samples under different evaluation conditions

Methods	**Distance** (m)					Avg
	5	**7.5**	**10**	**12.5**	**15**	
Normal	60.36	90.54	90.17	86.33	84.25	82.33
DAS [Wang *et al.*, 2021a]	63.61	89.44	86.81	81.01	78.66	79.91
FCA [Wang *et al.*, 2022]	51.59	68.65	52.56	32.94	44.82	50.11
DTA [Suryanto *et al.*, 2022]	35.22	68.03	54.49	33.77	44.41	47.18
ACTIVE [Suryanto *et al.*, 2023]	14.64	33.63	32.18	25.34	37.02	28.56
RAUCA [Zhou *et al.*, 2024]	17.06	24.17	13.19	10.43	15.12	15.99
Ours	**1.24**	**2.48**	**2.16**	**2.90**	**3.87**	**2.53**

Table 2: Evaluation results in various distances. Values are AP@0.5 (%) of the target vehicle averaged across different elevation and azimuth angles with YOLOv3.

Methods	**Weather Setting**					Avg
	Noon	Sunset	Night	Foggy	Rainy	
Normal	92.27	90.17	92.34	79.01	93.99	89.56
DAS [Wang *et al.*, 2021a]	86.67	86.81	88.74	81.70	88.54	86.51
FCA [Wang *et al.*, 2022]	66.99	52.56	73.55	59.81	64.64	63.51
DTA [Suryanto *et al.*, 2022]	69.48	54.49	68.99	55.32	67.40	63.14
ACTIVE [Suryanto *et al.*, 2023]	50.28	32.18	38.40	26.87	43.30	40.24
RAUCA [Zhou *et al.*, 2024]	29.63	13.19	27.49	16.02	24.86	22.24
Ours	**5.25**	**2.16**	**1.86**	**2.42**	**2.62**	**2.86**

Table 3: Evaluation results under diverse weather conditions. Values are AP@0.5 (%) of the target vehicle averaged across different azimuth angles with YOLOv3.

experiments are conducted in the CARLA simulation platform [Dosovitskiy *et al.*, 2017] to assess attack performance across various viewpoints, distances, weather conditions, and transferability across different object detectors.

Robustness Across Multiple Angles. We evaluated the effectiveness of adversarial attacks across $\mathbf{360°}$ azimuth angles and elevation angles from $\mathbf{0°}$ to $\mathbf{60°}$. At each elevation, images were captured encircling the target vehicle. Experiments were conducted in a white-box setting with YOLOv3, and results are shown in Table 1. Our findings indicate that attacks are more challenging at lower elevation angles, as evidenced by higher AP@50 scores due to non-camouflaged features like vehicle tires aiding detection. At higher elevations, where such features are less visible and only camouflaged areas remain, detector performance declines, resulting in more successful attacks. Our method consistently achieves the best results, with the lowest AP@50 scores across all angles, demonstrating robust performance across perspectives. Specifically, compared to the state-of-the-art RAUCA [Zhou *et al.*, 2024], our approach improves attack performance by an average of 11.03%.

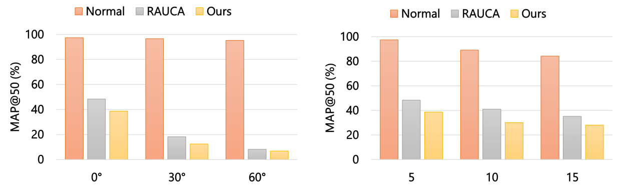
Robustness Across Multiple Distances. We conducted extensive experiments to assess performance across distances from 5 to 15 meters. Using the same protocol as for elevation evaluation, we averaged attack performance over full $\mathbf{360°}$ azimuth and $\mathbf{0°}$–$\mathbf{60°}$ elevation angles. Results in Table 2 reveal two key insights. First, although detection capability weakens with distance, attack performance doesn't necessarily improve due to loss of texture details at greater ranges, which can diminish camouflage effectiveness. Second, our method consistently achieves high attack success rates across all distances, outperforming existing approaches by 13.46%, highlighting the robustness and adaptability of the proposed NGC strategy.

Robustness Under Different Weather Conditions. Following previous studies [Zhou *et al.*, 2024], we evaluated our method under varying weather conditions, including *Noon* (high illumination), *Night* (low illumination), *Sunset* (moderate illumination), as well as *Foggy* and *Rainy* conditions, which reduce visibility. The results, summarized in Table 3, show that most existing methods are sensitive to these conditions. Extreme illumination (*Noon* and *Night*) leads to performance degradation due to overexposure or reduced visibility. Moreover, adverse weather conditions like *Foggy* and *Rainy* also impact performance by obscuring critical visual adversarial features. In contrast, our method demonstrates significantly improved robustness across all tested conditions. Notably, compared to state-of-the-art (SOTA) method RAUCA [Zhou *et al.*, 2024], our method achieves a consistent improvement, with an average gain of 19.38% across all weather scenarios. Despite this, high illumination (*Noon*) still poses challenges due to overexposure, providing an avenue for further enhancement.

Transferability Across Different Object Detectors. To evaluate the ability of our method to transfer across various object detectors, we carried out comprehensive experiments on detectors with diverse architectures, including single-stage, two-stage, and transformer-based models. These experiments compared our approach with several state-of-the-art (SOTA) methods. The outcomes, detailed in Table 4, reveal notable variations in method performance across differ-

Methods	**Single Stage**		**Two Stage**		**Transformer**	
	YOLOv3	YOLOX	FrRCN	MkRCN	DETR	PVT
Normal	90.17	95.30	95.68	97.71	97.56	90.36
DAS [Wang *et al.*, 2021a]	86.81	88.60	85.84	87.50	83.91	79.63
FCA [Wang *et al.*, 2022]	52.56	73.41	67.95	71.34	57.04	54.97
DTA [Suryanto *et al.*, 2022]	54.49	79.63	66.16	79.77	25.90	65.19
ACTIVE [Suryanto *et al.*, 2023]	32.18	49.45	46.13	50.48	31.49	41.09
RAUCA [Zhou *et al.*, 2024]	13.19	30.80	23.34	33.77	13.53	35.43
Ours	**2.16**	**22.65**	**14.57**	**28.11**	**6.28**	**28.25**

Table 4: Evaluating of transferability across various object detectors.



(a) Elevation Angle

(b) Distance (Relative)

Figure 6: Evaluation results in a real-world setting, assessed under varying elevation angles (a) and distances (b).

ent detector types. Notably, our method not only surpasses existing SOTA approaches in the white-box scenario but also achieves consistently stronger results in most black-box tests, highlighting its superior transferability.

4.3 Evaluation in Real-World Settings

In this section, we evaluate our method in real-world settings by printing and applying the optimized texture onto a car, as shown in Figure 5. The evaluation considers varying perspectives and distances. For perspectives, we tested three elevation angles and averaged results over a full 360° azimuth. For distances, we assessed three distinct ranges. Results in Figure 6 show that real-world conditions are more challenging than digital simulations. Lower elevation angles yield weaker performance, aligning with simulation trends. As distance increases, AP drops, indicating improved attack effectiveness—likely due to degraded detector performance at longer ranges. Additionally, while camouflage detail fades with distance, even robust features like uncovered tires become less detectable.

4.4 Computational Complexity

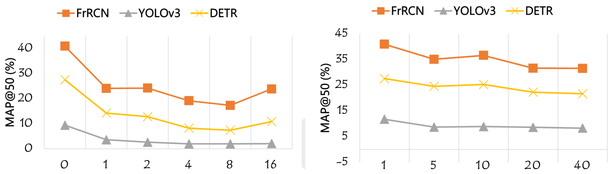
The NGC module has a complexity of $O(M \log N)$, where $M = |T_F|$ and $N = |T_U|$ denote the numbers of unsampled and sampled points, respectively. This stems from KD-Tree-based nearest neighbor search. The LPGD component introduces a complexity of $O(m^2 k)$, due to gradient orthogonalization over a texture of size m^2 across k optimization steps. In runtime tests, NGC introduces an 18.0% overhead (+0.0209s) per iteration, while LPGD adds 4.6% (+0.0054s), compared to the base forward and backward time (0.1163s).

	FP+BP	NGC	LPGD
Complexity	/	$O(M \log N)$	$O(m^2 k)$
Time cost (s)	0.1163	+0.0209	+0.0054

Table 5: Computational complexity and time cost.

Model	Baseline	w/ NGC	w/ LPGD	w/ NGC + LPGD
Yolov3	11.75	3.45	8.63	2.16
Faster RCNN	40.95	18.33	31.63	14.57

Table 6: Ablation study on our proposed components.



(a) Effect of search radius τ

(b) Effect of batchsize k

Figure 7: Ablation study on (a) search radius in NGC and (b) batch size for gradients orthogonalization in LPGD.

4.5 Ablation Studies

In this section, we perform ablation studies to analyze the impact of our proposed NGC and LPGD strategies. As shown in Table 6, both components individually improve attack performance, with their combination yielding the best results. For NGC, we further examine the effect of the search radius τ, which determines the furthest neighbor allowed for gradient propagation. For LPGD, we explore how the batch size used for gradient orthogonalization influences performance. To isolate the effects, we set the other component's effectiveness to zero during each study.

Effect of Search Radius τ. We evaluate our NGC strategy across a range of search radius values τ from 0 to 16. The results in Figure 7(a) show that attack performance initially improves with increasing τ, reaches an optimal point, and then declines. This validates the effectiveness of our method compared to the case where $\tau = 0$. However, excessively large τ allows gradients to propagate beyond the target surface, introducing noise to irrelevant textures. Thus, selecting an appropriate τ is critical for balancing local continuity and noise control.

Effect of Batch Size k. We assess the LPGD strategy by varying the batch size from 1 to 40. The results, depicted in Figure 7 (b), show a trend of increasing attack performance with larger k. This validates our method's effectiveness in reducing conflicts by considering a broader global context and decorrelating redundancy and conflict between gradients.

5 Conclusion

In this study, we identify two key challenges in physical adversarial camouflage: inconsistency in gradient sparsity and conflicting gradient updates. We introduce a novel framework that incorporates Nearest Gradient Calibration (NGC) and Loss-Prioritized Gradient Decorrelation (LPGD). NGC promotes gradient propagation from sampled to nearby unsampled texture points, ensuring local continuity across varying distances. LPGD prioritizes and orthogonalizes gradients to resolve redundancy and conflicts for gradients derived from different viewpoints. Our approach significantly enhances attack performance, robustness across diverse environments, and transferability across different detectors.

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities, Sun Yat-sen University under Grants No. 23xkjc010; the Shenzhen Science and Technology Program (No.KQTD20221101093559018).

References

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[Chen *et al.*, 2019] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open MMLab Detection Toolbox and Benchmark. *CoRR*, abs/1906.07155, 2019.

[Dombi, 2020] Szabolcs Dombi. Moderngl, high performance python bindings for opengl 3.3+. https://github.com/moderngl/moderngl, 2020.

[Dosovitskiy *et al.*, 2017] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[Duan *et al.*, 2019] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019.

[Everingham *et al.*, 2015] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

[Ge *et al.*, 2021] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO Series in 2021. *CoRR*, abs/2107.08430, 2021.

[Girshick, 2015] Ross Girshick. Fast r-cnn. In *International Conference on Computer Vision (ICCV)*, 2015.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision (ICCV)*, 2017.

[Kato *et al.*, 2018] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018.

[Leon *et al.*, 2013] Steven J Leon, Åke Björck, and Walter Gander. Gram-schmidt orthogonalization: 100 years and more. *Numerical Linear Algebra with Applications*, 20(3):492–532, 2013.

[Liu *et al.*, 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016.

[Madry *et al.*, 2017] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9), 2017.

[Redmon and Farhadi, 2018] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015.

[Ren *et al.*, 2016] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[Suryanto *et al.*, 2022] Naufal Suryanto, Yongsu Kim, Hyoeun Kang, Harashta Tatimma Larasati, Youngyeo Yun, Thi-Thu-Huong Le, Hunmin Yang, Se-Yoon Oh, and Howon Kim. DTA: Physical Camouflage Attacks Using Differentiable Transformation Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15305–15314, June 2022.

[Suryanto *et al.*, 2023] Naufal Suryanto, Yongsu Kim, Harashta Tatimma Larasati, Hyoeun Kang, Thi-Thu-Huong Le, Yoonyoung Hong, Hunmin Yang, Se-Yoon Oh, and Howon Kim. 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)*, pages 4305–4314, October 2023.

[Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[Wang *et al.*, 2021a] Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8565–8574, 2021.

[Wang *et al.*, 2021b] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping

Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[Wang *et al.*, 2022] Donghua Wang, Tingsong Jiang, Jialiang Sun, Weien Zhou, Zhiqiang Gong, Xiaoya Zhang, Wen Yao, and Xiaoqian Chen. 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, pages 2414–2422, 2022.

[Wei *et al.*, 2024] Hui Wei, Hao Tang, Xuemei Jia, Zhixiang Wang, Hanxun Yu, Zhubo Li, Shin'ichi Satoh, Luc Van Gool, and Zheng Wang. Physical adversarial attack meets computer vision: A decade survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[Zhou *et al.*, 2008] Kun Zhou, Qiming Hou, Rui Wang, and Baining Guo. Real-time kd-tree construction on graphics hardware. *ACM Transactions on Graphics (TOG)*, 27(5):1–11, 2008.

[Zhou *et al.*, 2024] Jiawei Zhou, Linye Lyu, Daojing He, and Yu Li. Rauca: A novel physical adversarial attack on vehicle detectors via robust and accurate camouflage generation. *arXiv preprint arXiv:2402.15853*, 2024.

[Zhu *et al.*, 2021] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 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.

[Zhu *et al.*, 2023] Wenjun Zhu, Xiaoyu Ji, Yushi Cheng, Shibo Zhang, and Wenyuan Xu. {TPatch}: A triggered physical adversarial patch. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 661–678, 2023.