Deformable and Multi-view Gradient-Aligned Physical Adversarial Camouflage

Jiawei Liang[1], Puning Zhao[1], Tianrui Lou[1], Haoqing Zhang[1], Xianghao Jiao[1], Bozheng Lin[1], Ming Zhang[2], and Xiaochun Cao[1⋆]

[1] Shenzhen Campus of Sun Yat-sen University, Shenzhen, China
caoxiaochun@mail.sysu.edu.cn
[2] National Key Laboratory of Science and Technology on Information System Security, Beijing, China

Abstract. Physical adversarial camouflage poses a significant threat to object detectors in safety-critical applications. However, synthesizing robust 3D textures remains non-trivial due to the structural mismatch between static texture parameters and dynamic physical observations. In this paper, we identify two fundamental optimization hurdles: spatially inconsistent gradients arising from sparse rendering visibility; and cross-view gradient misalignment, where contradictory signals from diverse viewpoints cause destructive interference. To overcome these, we propose a unified framework that reconciles spatial continuity with optimization stability. First, the Deformable Texture Field parameterizes textures as continuous fields via learnable flow grids, enforcing intrinsic smoothness while enabling adaptive geometric deformation to enhance the texture's representational flexibility. Second, a Gradient-Aligned Meta-Optimization strategy leverages multi-step exploration trajectories to implicitly maximize gradient alignment across conflicting viewpoints. Extensive digital and physical experiments demonstrate that our method achieves state-of-the-art robustness and transferability against advanced detectors under diverse viewing conditions.

Keywords: Adversarial camouflage · Object detection · Meta optimization

1 Introduction

Deep neural networks (DNNs) [8, 9] have become the cornerstone of modern object detection [20] systems used in autonomous driving and surveillance. Despite their remarkable performance, these systems are vulnerable to adversarial examples [16], *i.e.*, carefully crafted perturbations that induce catastrophic failures. While digital-domain attacks are well-documented, physical adversarial attacks [30], particularly those utilizing 3D camouflage textures, represent a more

⋆ Corresponding author.

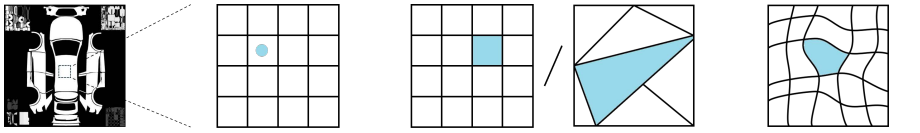


Fig. 1: Comparison of texture parameterization paradigms. Unlike (a) discrete pixel-wise updates that cause noise, or (b) rigid fixed-grid constraints that limit flexibility, (c) our Deformable Texture Field enables adaptive geometric deformation.

tangible threat to real-world security. Unlike 2D patches [34] that are viewpoint-dependent, 3D adversarial camouflage [33] seeks to optimize the entire surface of an object to ensure consistent evasion from omni-directional perspectives.

However, synthesizing robust 3D textures remains non-trivial due to the structural mismatch between static texture parameters (2D) and dynamic physical observations (3D). As illustrated in Fig. 1(a), pixel-wise optimization suffers from severe spatial discontinuity [11], where sparse visibility leads to disjoint gradient updates and fragile high-frequency artifacts susceptible to real-world noise. Furthermore, alternative parameterizations based on fixed grids or triangular faces (Fig. 1(b)) impose rigid geometric constraints. This limits the dynamics of the texture and restricts the optimization solution space, failing to capture the complex patterns required for multi-view evasion.

Furthermore, the multi-view optimization landscape is plagued by cross-view gradient conflicts. As illustrated in Fig. 2, gradients derived from distinct viewpoints often exhibit contradictory directions, *i.e.*, an update suppressing detection in View A may inadvertently increase visibility in View B. In standard sequential optimization (the orange trajectory), these conflicting signals result in destructive interference, causing the parameters to oscillate or diverge rather than converging to the intersection of view-specific manifolds.

To overcome these hurdles, we propose a unified framework that reconciles spatial continuity with optimization stability. Our first contribution is the Deformable Texture Field (DTF). Moving beyond discrete pixel parameterization, we generate the texture from continuous fields defined by two learnable low-resolution grids: a Content Grid for appearance and a Flow Grid for geometry. Through differentiable resampling, this enforces an intrinsic smoothness prior while enabling adaptive deformation for diverse viewing conditions. Our second contribution is a Gradient-Aligned Meta-Optimization (GAMO) strategy. Treating view clusters as distinct tasks, we employ multi-step local exploration with a strict state-reset mechanism. This approach leverages optimization trajectory information to implicitly maximize the directional consistency [17] (i.e., gradient alignment) between conflicting views, ensuring convergence toward a robust, shared parameter manifold.

The main contributions of this work are summarized as follows:

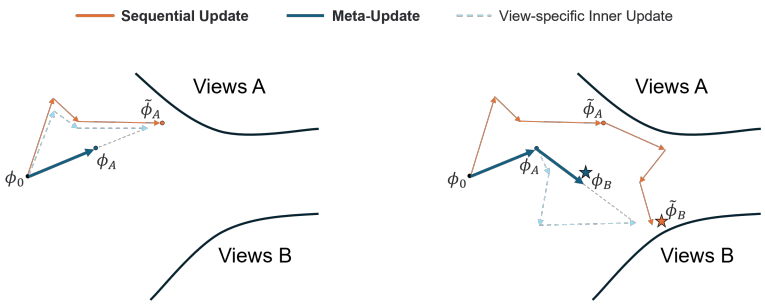


Fig. 2: Cross-view gradient conflict and meta-optimization. Sequential updates can oscillate when view-specific gradients point in conflicting directions. Our meta-optimization aggregates multi-step local trajectories and updates toward a shared region where gradients from different views are better aligned.

– We identify two optimization hurdles in 3D adversarial texture generation: spatially inconsistent gradients and cross-view misalignment leading to destructive interference.
– We propose the Deformable Texture Field, a continuous deformable parameterization method that enforces spatial coherence and enhances representational flexibility via learnable geometric flows.
– We introduce the Gradient-Aligned Meta-Optimization strategy, which leverages multi-step trajectories to implicitly maximize gradient alignment across conflicting viewpoints.
– Extensive digital and physical experiments demonstrate that our method achieves state-of-the-art robustness and transferability against advanced object detectors under diverse viewing conditions.

2 Related Work

2.1 Object Detection

Object detection has undergone a significant paradigm shift from traditional handcrafted features to sophisticated deep learning architectures. Dominant methodologies can be broadly categorized into anchor-based and anchor-free frameworks. Anchor-based methods comprise two-stage detectors such as Faster R-CNN [21] and Mask R-CNN [6] that utilize region proposal networks, as well as single-stage models like SSD [13] and YOLO [20] that prioritize inference speed via regression-based detection. Conversely, anchor-free detectors [3] streamline the pipeline by directly regressing object center points and bounding box dimensions. Recent innovations have introduced Transformer-based models, most notably DETR [35], which leverages self-attention mechanisms for end-to-end

object prediction. While these detectors achieve remarkable accuracy, their inherent reliance on local feature patterns renders them vulnerable to meticulously crafted adversarial perturbations.

2.2 Adversarial Attacks

Adversarial attacks seek to compromise the integrity of deep neural networks by introducing subtle perturbations that induce erroneous outputs [26]. In the digital domain, gradient-based techniques such as FGSM [5] and PGD [16] have demonstrated high potency in fooling classifiers and detectors. However, these digital perturbations often lack robustness when subjected to real-world distortions. Consequently, physical adversarial attacks [30,34] have emerged as a more tangible threat. These are typically manifested as adversarial patches attached to flat surfaces, which are viewpoint-dependent, or adversarial camouflage that envelops the entire 3D geometry of an object. Unlike localized patches, adversarial camouflage provides a multi-angle defense-evasion capability, though it presents significantly higher optimization complexity due to the non-linear nature of 3D-to-2D projections.

2.3 Physical Adversarial Camouflage

Mainstream adversarial camouflage generation relies on differentiable rendering to optimize 3D textures through backpropagation across diverse camera perspectives. Pioneering efforts like FCA [27] and DAS [28] established the feasibility of optimizing UV maps to suppress objectness scores or distract model attention. To enhance environmental robustness, DTA [24] incorporated a transformation network to simulate lighting and shadows, while ACTIVE [25] focused on improving the aesthetic naturalness and resolution of the generated patterns. Recent non-pixel texture parameterizations include adversarial clothing textures [7] and latent-variable camouflage for remote sensing [18]. More recently, RAUCA [33] incorporated advanced rendering modules to account for complex weather conditions, and GRAC [11] applied trainable weights to balance gradient contributions across multiple views. However, these methods generally rely on fixed grid parameterizations, which constrain the optimization space, and struggle to achieve effective gradient alignment across varying views.

3 Preliminaries

In this section, we formally define the problem of multi-view physical adversarial texture generation and analyze the fundamental challenges that motivate our proposed framework.

3.1 Problem Definition

Let $\mathcal{M}$ denote a 3D object mesh equipped with a fixed UV mapping, and $f : \mathbb{R}^{H_{img} \times W_{img} \times 3} \to \mathbb{R}^C$ be a target object detector (e.g., YOLO [20]). Our

objective is to synthesize a single high-resolution texture map $\mathbf{T} \in [0,1]^{H \times W \times 3}$ such that the rendered projections of $\mathcal{M}$ effectively evade detection by f across varying viewing angles.

The imaging process is simulated via a differentiable rendering function $\mathcal{R}(\mathcal{M}, \mathbf{T}, \mathbf{v})$, which projects the textured mesh onto a 2D image plane under a specific camera viewpoint $\mathbf{v}$ sampled from a distribution $\mathcal{V}$. The optimization objective is to minimize the expected adversarial loss over the viewpoint distribution:

$$\min_{\mathbf{T}} \mathbb{E}_{\mathbf{v} \sim \mathcal{V}} \left[\mathcal{L}_{adv}(f(\mathcal{R}(\mathcal{M}, \mathbf{T}, \mathbf{v})), y_{gt}) \right] + \lambda \mathcal{L}_{reg}(\mathbf{T}), \tag{1}$$

where $\mathcal{L}_{adv}$ denotes the adversarial loss designed to minimize the confidence scores of the correct category associated with the target bounding boxes, and $\mathcal{L}_{reg}$ represents the regularization term imposed on the texture parameterization.

3.2 Challenges

Synthesizing robust 3D adversarial textures is fundamentally hindered by two primary optimization hurdles arising from the 3D-to-2D projection process.

Spatially Inconsistent Gradients Pixel-wise optimization is strictly gated by geometric visibility, where gradients are only back-propagated to visible texels. As the viewpoint shifts, active gradient regions fluctuate, causing spatially adjacent pixels to receive asynchronous and discontinuous updates. This disrupts local texture correlations, manifesting as high-frequency artifacts and topological tearing that significantly degrade the physical realizability of the generated camouflage.

Cross-View Gradient Conflict Minimizing a joint adversarial loss over a distribution of viewpoints often results in severe directional conflicts. Gradients derived from distinct viewpoints (e.g., frontal and rear perspectives) frequently exhibit negative cosine similarity. Standard joint optimization, which merely averages these gradients, suffers from destructive interference where valid optimization signals cancel out. This leads to training stagnation or divergence, highlighting the need to find a parameter manifold where gradient directions from diverse views are implicitly aligned for consistent optimization progress.

4 Methodology

Our framework (Fig. 3) aims to resolve the structural mismatch between static texture parameters and dynamic physical observations. To address the Spatially Inconsistent Gradient and Cross-View Gradient Conflict identified in Section 3, we propose a unified optimization paradigm. This paradigm consists of the Deformable Texture Field, which ensures spatial continuity, and the Gradient-Aligned Meta-Optimization, which maximizes the directional consistency of gradients across diverse viewpoints.

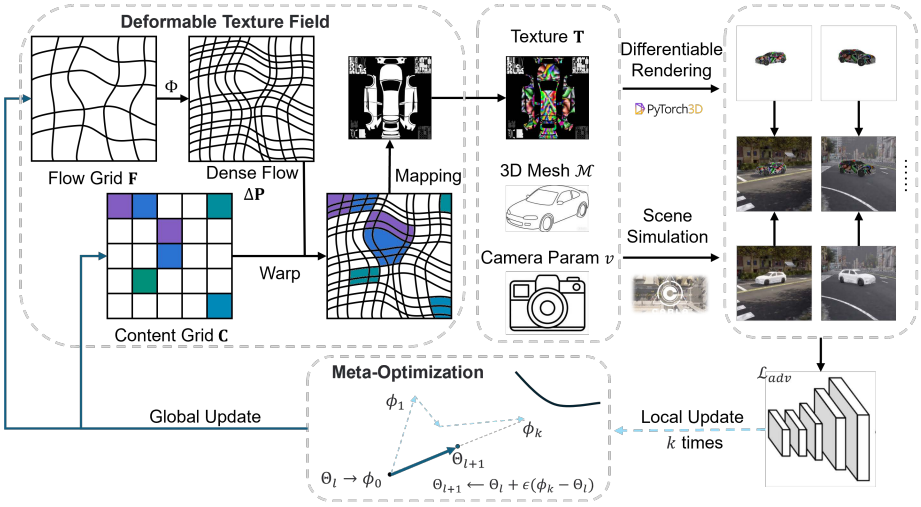


Fig. 3: Overview of the proposed framework. The Deformable Texture Field synthesizes a continuous high-resolution camouflage by warping a learnable content grid with a dense flow field, enforcing spatial coherence. Gradient-Aligned Meta-Optimization resolves cross-view conflicts through multi-step local exploration and global aggregation, aligning gradients across viewpoints.

4.1 Deformable Texture Field

Standard pixel-wise optimization treats textures as collections of independent variables, causing a sparse visibility problem: in each iteration, occluded regions receive no feedback, while updates to visible pixels remain isolated. Over time, this asynchronous process disrupts local spatial correlations, manifesting as high-frequency adversarial patterns that are highly susceptible to real-world noise.

To overcome this, we propose the Deformable Texture Field (DTF). Instead of optimizing discrete pixels, we optimize a low-resolution control grid and synthesize the high-resolution texture via differentiable interpolation. This formulation inherently imposes a smoothness prior to ensure spatial coherence, while allowing adaptive geometric deformations that enhance the texture's representational flexibility.

We decouple the high-resolution texture map $\mathbf{T}$ into two learnable low-resolution components: a Content Grid $\mathbf{C} \in \mathbb{R}^{H' \times W' \times 3}$ and a Flow Grid $\mathbf{F} \in \mathbb{R}^{H' \times W' \times 2}$. Here, H' and W' denote the dimensions of the control grids (e.g., 64×64), which are lower than the target texture resolution $H \times W$. The Content Grid $\mathbf{C}$ encodes the fundamental color patterns, while the Flow Grid $\mathbf{F}$ represents the geometric offsets used to deform these patterns. The final texture $\mathbf{T} \in \mathbb{R}^{H \times W \times 3}$ is synthesized through the following two differentiable steps.

Dense Flow Generation The Flow Grid $\mathbf{F}$ defines sparse control vectors for local geometric deformations. To assign a specific deformation to every pixel

in the target high-resolution texture, we reconstruct a dense deformation field $\Delta\mathbf{P} \in \mathbb{R}^{H\times W\times 2}$. This is achieved by upsampling $\mathbf{F}$ to the target resolution (H, W) using a bilinear interpolation operator Φ:

$$\Delta\mathbf{P}(\mathbf{u}) = \Phi(\mathbf{F}, \mathbf{u}), \quad \forall \mathbf{u} \in \Omega_{tex}, \tag{2}$$

where $\mathbf{u} = (x, y)$ represents the normalized pixel coordinates in the high-resolution texture space $\Omega_{tex} = [0, 1]^2$. By utilizing bilinear interpolation for upsampling, we ensure that $\Delta\mathbf{P}$ varies smoothly across the entire texture surface. This implicitly enforces spatial correlation and prevents adjacent pixels from undergoing disjoint updates even under sparse gradient supervision.

Differentiable Texture Resampling Once the dense deformation field is generated, the color for each pixel in $\mathbf{T}$ is determined via an inverse sampling mechanism. For each target pixel $\mathbf{u} \in \Omega_{tex}$, we compute its corresponding sampling location $\mathbf{p}' = (x', y')$ within the source normalized Content Grid space by applying the learned deformation:

$$\mathbf{p}'(\mathbf{u}) = \mathbf{u} + \Delta\mathbf{P}(\mathbf{u}). \tag{3}$$

To derive the final color, the normalized coordinate $\mathbf{p}'$ is mapped to the discrete $H' \times W'$ coordinate system of $\mathbf{C}$. Let $\hat{\mathbf{p}} = (x' \cdot (W' - 1), y' \cdot (H' - 1))$ denote the point in grid index space. The texture value at $\mathbf{u}$ is calculated as the weighted sum of the four nearest integer neighbors $\mathbf{q}$ surrounding $\mathbf{v}$ in the Content Grid $\mathbf{C}$:

$$\mathbf{T}(\mathbf{u}) = \sum_{\mathbf{q}\in\mathcal{N}(\hat{\mathbf{p}})} \mathbf{C}(\mathbf{q}) \cdot \kappa(\hat{\mathbf{p}} - \mathbf{q}), \tag{4}$$

where $\mathcal{N}(\hat{\mathbf{p}})$ represents the set of four integer coordinates in the $H' \times W'$ grid:

$$\begin{aligned}\mathcal{N}(\hat{\mathbf{p}}) = \{&(\lfloor \hat{p}_x \rfloor, \lfloor \hat{p}_y \rfloor), (\lfloor \hat{p}_x \rfloor, \lceil \hat{p}_y \rceil),\\ &(\lceil \hat{p}_x \rfloor, \lfloor \hat{p}_y \rfloor), (\lceil \hat{p}_x \rceil, \lceil \hat{p}_y \rceil)\}.\end{aligned} \tag{5}$$

The differentiable bilinear sampling kernel k is defined by the product of the distances to these neighbors:

$$\kappa(\delta_x, \delta_y) = \max(0, 1 - |\delta_x|) \cdot \max(0, 1 - |\delta_y|). \tag{6}$$

This formulation ensures that the synthesized texture $\mathbf{T}$ is a continuous function of the control parameters $\mathbf{C}$ and $\mathbf{F}$. By optimizing the Flow Grid $\mathbf{F}$, the framework can adaptively distort the sampling grid to concentrate more representational capacity on salient adversarial regions, effectively providing the geometric flexibility to adapt to diverse viewpoints without increasing the total number of parameters.

Regularization To prevent excessive geometric distortion and ensure physical realizability, we apply an L_2 regularization on the Flow Grid $\mathbf{F}$. As introduced in Eq. (1), we define $\mathcal{L}_{reg}(\mathbf{F}) = \|\mathbf{F}\|_2^2$, which restricts the magnitude of the learned offsets to prevent irregular warping.

4.2 Gradient-Aligned Meta-Optimization

As analyzed in Challenge 2, standard joint training suffers from destructive gradient interference due to conflicting objectives across views. To overcome this, we draw inspiration from first-order meta-learning theory [17]. While originally designed for few-shot learning, we reinterpret its theoretical properties in the context of physical attacks: performing multiple steps of local updates before a global update implicitly optimizes for multi-view collaborative optimization. Building on this insight, we propose a Gradient-Aligned Meta-Optimization strategy. Unlike standard training that averages gradients directly, our approach leverages the high-order derivative terms generated by multi-step trajectories to enforce directional consistency across conflicting viewpoints.

We formally structure this process as a dual-phase optimization involving Global Meta-Parameters $\Theta = \{\mathbf{C}, \mathbf{F}\}$ and local temporary parameters ϕ. Thus, GAMO operates on the DTF parameter space, aligning view-specific updates of the same content and flow grids within a spatially coherent deformable representation. The training iterates through episodes, using ϕ for local exploration and Θ for global robustness aggregation.

Inner Loop: Local Exploration In each training episode, we first initialize the local parameters $\phi_0 \leftarrow \Theta$. We then execute an inner loop consisting of k optimization steps. At each step t ($0 \leq t < k$), we sample a specific viewpoint (or a mini-batch) $\mathbf{v}_t$ and update the local parameters to minimize the adversarial loss. The update rule is formulated as:

$$\phi_{t+1} \leftarrow \phi_t - \eta \nabla_\phi \mathcal{L}(\phi_t, \mathbf{v}_t), \tag{7}$$

where η is the inner learning rate. This process generates a sequence of parameters $\mathcal{T} = \{\phi_0, \ldots, \phi_k\}$ that explores the local loss landscape.

Regarding the optimization strategy, we rigorously reset the inner optimizer state at the start of each episode (before $t = 0$). This ensures the trajectory is driven solely by the currently sampled k viewpoints, preventing historical bias. Within the k steps, we allow the optimizer to accumulate local momentum to efficiently traverse the curvature of the current view's loss basin.

The objective of this procedure is to compute a multi-step optimization trajectory that encapsulates local curvature information. As established in [17], the difference $(\phi_k - \phi_0)$ effectively approximates the gradient of the loss surface with respect to the task distribution. By performing sequential updates, the outer loop leverages this trajectory to implicitly maximize the directional consistency across conflicting views, resolving the gradient conflict observed in standard joint training.

Outer Loop: Maximizing Gradient Alignment Once the local trajectory is computed, the outer loop aggregates this information into the global parameters Θ:

Algorithm 1 Adversarial Texture Optimization Framework

Require: Mesh $\mathcal{M}$, Detector f, Inner steps k, Rates η, ϵ.
Initialize Global Parameters $\Theta = \{\mathbf{C}, \mathbf{F}\}$
while not converged **do**
  Sample viewpoint $\mathbf{v}$
  $\phi \leftarrow \Theta$ {Initialize local parameters}
  Reset optimizer state $\mathcal{S} \leftarrow \emptyset$ {Flush momentum}
  for $t = 0$ to $k - 1$ **do**
    Synthesize $\mathbf{T}$ via Eq. 4 using current ϕ
    Render images $I = \mathcal{R}(\mathcal{M}, \mathbf{T}, \mathbf{v})$
    Compute Loss $\mathcal{L} = \mathcal{L}_{adv}(I) + \lambda \mathcal{L}_{reg}(\mathbf{T})$
    Apply gradients: $\phi \leftarrow \text{OptimStep}(\phi, \nabla_\phi \mathcal{L})$
  end for
  Update Global: $\Theta \leftarrow \Theta + \epsilon(\phi - \Theta)$
end while
return Optimized Texture $\mathbf{T}$ synthesized from Θ

$$\Theta \leftarrow \Theta + \epsilon(\phi_k - \Theta), \tag{8}$$

where ϵ is the outer step size. As derived in [17], this strategy implicitly optimizes a modified objective function. The effective update direction approximates:

$$\Delta\Theta \approx -\mathbb{E}[\mathbf{g}_i] + \underbrace{\frac{\eta(k-1)}{4} \nabla_\Theta \left(\mathbb{E}\left[\mathbf{g}_i \cdot \mathbf{g}_j\right] \right)}_{\text{Implicit Gradient Alignment}}, \tag{9}$$

where $\mathbf{g}_i$ and $\mathbf{g}_j$ denote gradients from different viewpoints. The first term corresponds to the standard adversarial loss minimization. Crucially, the second term emerges from the multi-step nature of the inner loop. It acts as a regularizer that maximizes the inner product of gradients $(\mathbf{g}_i \cdot \mathbf{g}_j)$ between different viewpoints. This ensures that the optimization progresses toward a robust manifold where gradient directions are mutually supportive rather than contradictory. A detailed Taylor-expansion derivation is provided in the supplementary material.

The complete algorithm is summarized in Algorithm 1.

5 Experiment

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.

5.1 Experimental Settings

Implementation Setup. We conduct experiments using the CARLA simulator [2], adopting the Audi E-Tron as the target vehicle [27, 28]. Our dataset

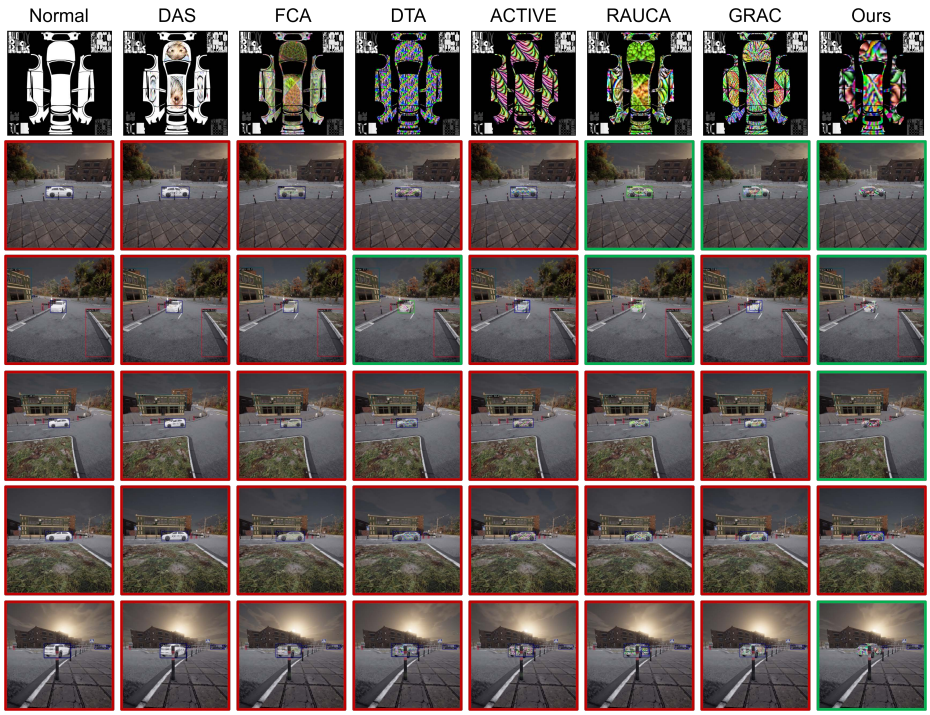
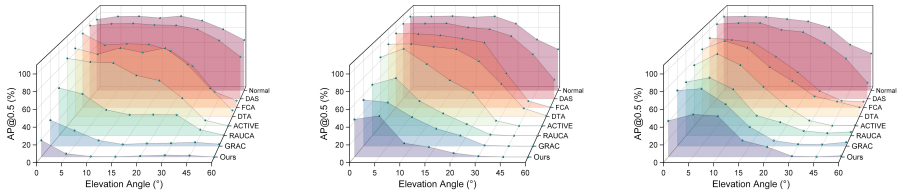


Fig. 4: Qualitative evaluation in the digital environment. The top row compares the optimized 2D textures. The subsequent rows display rendering results under varying distances, camera angles. Green image borders indicate successful evasion, while red borders denote detection failures.

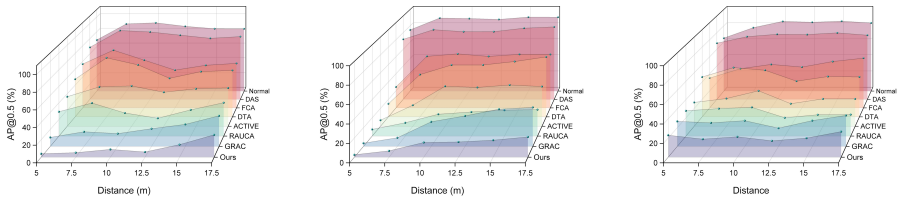
contains 20,000 images captured from randomized perspectives with distances ranging from 5m to 20m. We employ PyTorch3D [19] for differentiable rendering, using binary visibility masks to restrict gradient back-propagation to observable surfaces. To enhance generalization, the textured vehicle is rendered onto diverse background scenes extracted from the CARLA simulator [2]. Additional category-generalization and weather/lighting robustness experiments are provided in the supplementary material.

Hyperparameter Configuration. We initialize the learnable Content and Flow Grids at 64×64 resolution, upsampled to synthesize a 2048×2048 high-resolution texture. For the Gradient-Aligned Meta-Optimization (GAMO), we set the inner loop steps to $k = 16$ and use an SGD optimizer with learning rate $\eta = 0.1$. The supplementary sensitivity analysis supports the choices of 64×64 grid resolution and $k = 16$. The flow regularization λ is set to 0.01 to balance spatial smoothness and deformation capacity. All models are trained for 3 epochs on a single NVIDIA A100 GPU.



(a) YOLOv3 [20] **(b)** Faster R-CNN [22] **(c)** PVT [29]

Fig. 5: Comparison of camouflage effectiveness from various elevation angles. For each elevation, values represent the AP@0.5 (%) of the target car averaged across different azimuth angles. Lower value indicates better camouflage performance.



(a) YOLOv3 [20] **(b)** Faster R-CNN [22] **(c)** PVT [29]

Fig. 6: Evaluation of camouflage effectiveness at various distances. Reported values are AP@0.5 (%) of the target vehicle, averaged across different elevation and azimuth angles. Lower AP@0.5 demonstrates better camouflage performance.

Evaluation Protocol. We rigorously evaluate the adversarial robustness of the generated camouflage under multi-view conditions. For viewpoint evaluation, we test across six elevation angles $\{0°, 10°, 20°, 30°, 45°, 60°\}$, with a full 360° azimuthal sweep performed every 2° at each elevation. Distance robustness is assessed at $\{5, 7.5, 10, 12.5, 15, 17.5\}$ meters. For real-world validation, the optimized textures are printed and applied to 1:24 scale physical vehicle models, where the effectiveness is quantified using video sequences captured from omni-directional perspectives.

Baselines and Metrics. We benchmark our method against state-of-the-art adversarial camouflage techniques, including UV-map-based methods like DAS [28], FCA [27], and RAUCA [33], GRAC [11], as well as projection-based methods such as DTA [24] and ACTIVE [25]. In accordance with standard object detection protocols [4], the Average Precision at an IoU threshold of 0.5 (AP@0.5) is employed as the primary metric.

Target Detection Models. We use YOLOv3 [20] as our white-box surrogate model. To assess transferability, we evaluate our camouflage against diverse black-box architectures implemented via MMDetection [1], all pretrained on COCO. Our evaluation suite includes CNN-based models (Faster R-CNN [22],

Table 1: Transferability across advanced object detectors.

Methods	RTMDet [15]	ConvNeXt [14]	GLIP [10]	DINO [32]	Grounding DINO [12]
Normal	94.49	95.36	96.24	85.95	91.24
DAS [28]	92.61	83.92	95.54	79.11	83.69
FCA [27]	71.56	56.95	82.58	60.26	72.12
DTA [24]	72.18	61.02	80.84	35.12	62.60
ACTIVE [25]	67.69	55.13	81.99	41.42	59.92
RAUCA [33]	51.03	37.68	60.12	28.55	40.96
GRAC [11]	52.90	32.66	**41.42**	23.03	47.39
Ours	**45.44**	**27.84**	43.72	**13.36**	**35.56**

Table 2: Physical evaluation results under varying viewing angles. We report AP@0.5 (%) on white-box YOLOv3 [20] and black-box Faster R-CNN [22]. Lower values indicate better camouflage effectiveness.

Methods	YOLOv3 [20]			Faster R-CNN [22]		
	0°	20°	45°	0°	20°	45°
Normal	98.01	94.70	88.08	99.45	97.42	94.70
DAS [28]	85.43	68.21	73.51	91.39	84.77	80.79
FCA [27]	72.85	52.32	54.30	98.01	84.77	68.87
DTA [24]	68.87	43.71	35.10	89.40	80.79	60.93
ACTIVE [25]	45.77	42.02	29.30	79.47	64.90	41.72
RAUCA [33]	39.74	29.30	23.39	78.15	60.26	30.45
GRAC [11]	41.72	26.44	20.53	74.17	54.30	27.08
Ours	**34.53**	**23.39**	**18.53**	**70.20**	**41.15**	**23.39**

RTMDet [15], ConvNeXt [14]), vision transformers (PVT [29], DINO [32]), and vision-language models (GLIP [10], Grounding DINO [12]).

5.2 Performance in Digital Simulation

Robustness Across Viewing Angles. Fig. 5 reports the AP@0.5 across elevation angles (0°–60°) on YOLOv3 [20] (white-box), Faster R-CNN [22], and PVT [29] (black-box). While detection naturally degrades at higher elevations as the roof's adversarial texture becomes more exposed, our method consistently achieves the lowest AP@0.5 across all angles and architectures. It substantially outperforms RAUCA [33] and GRAC [11] under both oblique and top-down views, demonstrating superior perspective-invariant transferability.

Robustness Across Operational Distances. Fig. 6 evaluates performance over distances from 5 m to 17.5 m. Although detection rates generally decline with distance due to reduced object scale, our method maintains consistently low AP@0.5 across the full range for all detectors. This confirms the robustness of our adversarial texture against scale and resolution variations, effectively avoiding the performance degradation observed in baseline methods at longer ranges and ensuring reliable camouflage regardless of the camera-to-target distance.

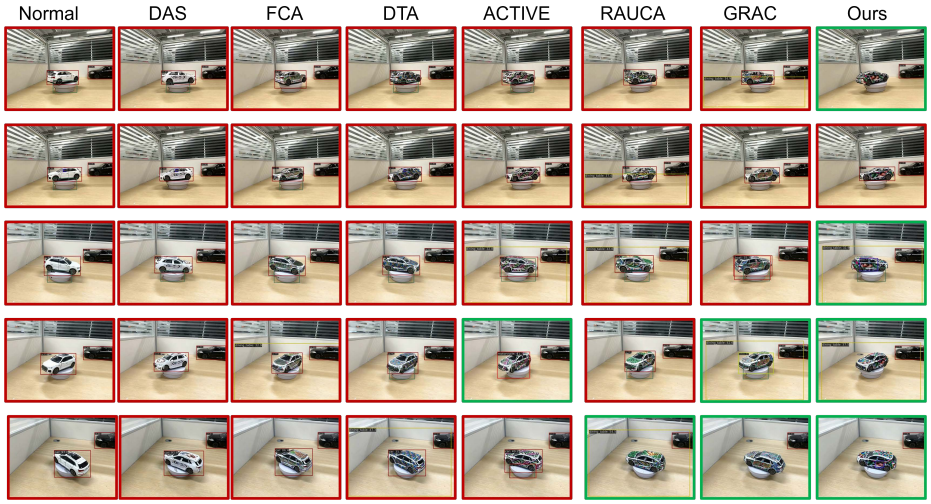


Fig. 7: Qualitative evaluation in the physical world.

Transferability to Advanced Detectors. Table 1 evaluates black-box transferability on five advanced detectors, spanning real-time CNNs, Vision Transformers, and Vision-Language models. Our method achieves the lowest AP@0.5 on four out of five architectures. Specifically, it outperforms the previous best method, GRAC [11], by margins of 7.46% on RTMDet [15], 4.82% on ConvNeXt [14], and 9.67% on DINO [32]. Ours is slightly weaker on GLIP [10], whose language-grounded matching is less sensitive to texture perturbations. Our approach maintains strong effectiveness on its successor, Grounding DINO [12], dropping the AP to 35.56% compared to GRAC's 47.39%. This validates our method's robust generalization across diverse, unknown detection paradigms.

Qualitative Analysis. Fig. 4 visualizes the optimized textures and detection results. While baselines frequently fail under varying distances, perspective distortions, and complex lighting conditions like sunset (indicated by red borders), our method consistently evades detection (green borders), confirming its superior digital robustness.

5.3 Real-World Physical Evaluation

Table 2 presents quantitative physical attack results under varying elevations. Our method achieves the lowest AP@0.5 across all settings. In the white-box YOLOv3 [20] scenario at 0°, we reduce the detection rate to 34.53%, outperforming RAUCA [33] and GRAC [11] by 5.21% and 7.19%, respectively. Importantly, our approach demonstrates superior black-box transferability on Faster

Table 3: Ablation study on DTF and GAMO modules. We report AP@0.5 (%) across three detectors and viewing angles; lower values indicate better effectiveness.

Components		YOLOv3 [20]			Faster R-CNN [22]			PVT [29]		
DTF	**GAMO**	0°	20°	45°	0°	20°	45°	0°	20°	45°
×	×	61.70	42.02	17.06	85.17	41.15	20.70	80.12	46.65	25.82
✓	×	33.06	18.95	9.26	57.92	20.96	15.79	56.42	24.52	11.67
×	✓	40.13	12.08	7.71	66.53	22.92	8.32	62.22	18.06	10.13
✓	✓	19.05	2.19	1.80	42.92	9.01	1.04	41.3	11.67	1.41

R-CNN [22]: at 0°, we achieve an AP of 70.20%, surpassing RAUCA and GRAC by 7.95% and 3.97%. This advantage is amplified at 20°, where we limit the AP to 41.15%, significantly lower than GRAC's 54.30%. Even at 45°, our method maintains leading performance, confirming its real-world robustness against both known and unknown detectors.

As visualized in Fig. 7, our adversarial camouflage evades most detection across diverse camera elevations and azimuths in physical settings. These qualitative results validate the efficacy of our approach in bridging the sim-to-real domain gap, ensuring that the robustness achieved in simulation transfers effectively to successful physical adversarial attacks.

5.4 Ablation Study.

Table 3 analyzes the contributions of the DTF and GAMO modules. Removing both components yields the highest detection rates. Introducing DTF or GAMO individually drops the AP@0.5 on YOLOv3 [20] at 0° from 61.70% to 33.06% and 40.13%, respectively, indicating that both modules independently enhance camouflage. Notably, their combination produces a strong synergistic effect. The full method consistently achieves the lowest detection rates, reaching 19.05% on YOLOv3 [20] at 0° and dropping to 1.80% at 45°. This trend holds for black-box detectors like Faster R-CNN [22], where the full method reduces the baseline AP by 42.25% at 0°. These results confirm that DTF reduces spatial fragmentation caused by sparse visible-texel updates, while GAMO further resolves cross-view gradient conflicts.

Gradient Alignment Analysis. Beyond the module-level ablation, we further isolate the optimizer by keeping the DTF parameterization unchanged. Table 4 compares several gradient-conflict handling strategies under this controlled setting. GAMO achieves the lowest AP@0.5, the highest gradient cosine similarity, and the lowest conflict ratio, indicating that its performance gain is tied to improved cross-view gradient alignment rather than only to the texture representation.

Table 4: Comparison of gradient-conflict handling methods with DTF fixed.

Method	AP@0.5↓	Cosine↑	Conflict↓
DTF	40.26	0.0048	0.4871
DTF + PCGrad [31]	20.25	0.0051	0.4709
DTF + MGDA [23]	13.42	0.0063	0.4507
DTF + GRAC [11]	12.34	0.0055	0.4494
DTF + GAMO	**7.68**	**0.0070**	**0.4345**

Efficiency Analysis. Since GAMO introduces an inner-outer optimization structure, we examine whether this design adds substantial overhead beyond the detector/render back-propagation shared by all methods. Table 5 reports the per-view cost of representative methods. The outer operation in GAMO only forms the trajectory displacement and applies one global update, so detector/render back-propagation remains the dominant cost. As a result, our method uses lower per-view FLOPs and latency than GRAC and RAUCA.

Table 5: Efficiency comparison.

Metric	GRAC [11]	RAUCA [33]	Ours
GFLOPs	8505.60	973.21	**140.84**
ms	430.10	265.67	**135.17**

6 Conclusion

In this work, we present a unified framework to bridge the structural mismatch in generating 3D physical adversarial camouflage. To overcome the twin challenges of spatial discontinuity and cross-view conflicts, we introduce the Deformable Texture Field (DTF) alongside a Gradient-Aligned Meta-Optimization (GAMO) strategy. DTF parameterizes textures via continuous learnable grids to enforce intrinsic smoothness while enabling dynamic geometric deformation for enhanced representational flexibility. GAMO mitigates destructive multi-view interference by implicitly maximizing gradient alignment across diverse optimization trajectories. Extensive experiments confirm that our synergistic approach significantly outperforms state-of-the-art methods in both digital and physical scenarios, achieving superior perspective-invariant robustness. These findings expose critical vulnerabilities in modern detectors, emphasizing the urgent need for physically grounded defenses in safety-critical applications.

Acknowledgements

This work was supported by the Shenzhen Science and Technology Program (No. SYSRD20250529113401002, KJZD20240903095730039), the National Natural Science Foundation of China (No. U2541229), and the Ningbo Science and Technology Innovation 2025 Major Project (2025Z027).

References

1. 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., Lin, D.: MMDetection: Open MMLab Detection Toolbox and Benchmark. CoRR **abs/1906.07155** (2019), `http://arxiv.org/abs/1906.07155`
2. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
3. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6569–6578 (2019)
4. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**, 98–136 (2015)
5. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: International Conference on Computer Vision (ICCV) (2017)
7. Hu, Z., Huang, S., Zhu, X., Sun, F., Zhang, B., Hu, X.: Physically realizable natural-looking clothing textures evade person detectors via 3d modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16975–16984 (2023)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
9. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
10. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10965–10975 (2022)
11. Liang, J., Liang, S., Lou, T., Zhang, M., Li, W., Fan, D., Cao, X.: Gradient-reweighted adversarial camouflage for physical object detection evasion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13880–13889 (2025)
12. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: European conference on computer vision. pp. 38–55. Springer (2024)
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European Conference on Computer Vision (ECCV) (2016)
14. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)
15. Lyu, C., Zhang, W., Huang, H., Zhou, Y., Wang, Y., Liu, Y., Zhang, S., Chen, K.: Rtmdet: An empirical study of designing real-time object detectors. arXiv preprint arXiv:2212.07784 (2022)

16. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. stat **1050**(9) (2017)
17. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018)
18. Peng, Z., Chen, J., Shi, Z., Zou, Z.: Physical adversarial camouflage generation in optical remote sensing images. IEEE Transactions on Information Forensics and Security **20**, 6308–6323 (2025). https://doi.org/10.1109/TIFS.2025.3581771
19. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv preprint arXiv:2007.08501 (2020)
20. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
21. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems (NeurIPS) **28** (2015)
22. Ren, S., He, K., Girshick, R., Sun, J.: 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)
23. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. In: Advances in Neural Information Processing Systems (2018)
24. Suryanto, N., Kim, Y., Kang, H., Larasati, H.T., Yun, Y., Le, T.T.H., Yang, H., Oh, S.Y., 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)
25. Suryanto, N., Kim, Y., Larasati, H.T., Kang, H., Le, T.T.H., Hong, Y., Yang, H., Oh, S.Y., 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)
26. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014), http://arxiv.org/abs/1312.6199
27. Wang, D., Jiang, T., Sun, J., Zhou, W., Gong, Z., Zhang, X., Yao, W., 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. vol. 36, pp. 2414–2422 (2022)
28. Wang, J., Liu, A., Yin, Z., Liu, S., Tang, S., 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)
29. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 568–578 (2021)
30. Wei, H., Tang, H., Jia, X., Wang, Z., Yu, H., Li, Z., Satoh, S., Van Gool, L., Wang, Z.: Physical adversarial attack meets computer vision: A decade survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)

31. Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. In: Advances in Neural Information Processing Systems (2020)
32. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
33. Zhou, J., Lyu, L., He, D., Li, Y.: Rauca: A novel physical adversarial attack on vehicle detectors via robust and accurate camouflage generation. arXiv preprint arXiv:2402.15853 (2024)
34. Zhu, W., Ji, X., Cheng, Y., Zhang, S., Xu, W.: {TPatch}: A triggered physical adversarial patch. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 661–678 (2023)
35. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., 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), https://openreview.net/forum?id=gZ9hCDWe6ke