Physically Realizable Natural-Looking Clothing Textures Evade Person Detectors via 3D Modeling

Zhanhao Hu[1*] Wenda Chu[2,1*] Xiaopei Zhu[3,1] Hui Zhang[4] Bo Zhang[1] Xiaolin Hu[1,5,6†]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
[3]School of Integrated Circuits, Tsinghua University, Beijing, China
[4]Beijing Institute of Fashion Technology, Beijing, China
[5]IDG/McGovern Institute for Brain Research, THBI, Tsinghua University, Beijing, China
[6]Chinese Institute for Brain Research (CIBR), Beijing, China
{huzhanha17, chuwd19, zxp18}@mails.tsinghua.edu.cn
fzyzhh@bift.edu.cn, {dcszb, xlhu}@mail.tsinghua.edu.cn

Abstract

Recent works have proposed to craft adversarial clothes for evading person detectors, while they are either only effective at limited viewing angles or very conspicuous to humans. We aim to craft adversarial texture for clothes based on 3D modeling, an idea that has been used to craft rigid adversarial objects such as a 3D-printed turtle. Unlike rigid objects, humans and clothes are non-rigid, leading to difficulties in physical realization. In order to craft natural-looking adversarial clothes that can evade person detectors at multiple viewing angles, we propose adversarial camouflage textures (AdvCaT) that resemble one kind of the typical textures of daily clothes, camouflage textures. We leverage the Voronoi diagram and Gumbel-softmax trick to parameterize the camouflage textures and optimize the parameters via 3D modeling. Moreover, we propose an efficient augmentation pipeline on 3D meshes combining topologically plausible projection (TopoProj) and Thin Plate Spline (TPS) to narrow the gap between digital and real-world objects. We printed the developed 3D texture pieces on fabric materials and tailored them into T-shirts and trousers. Experiments show high attack success rates of these clothes against multiple detectors.

1. Introduction

Deep Neural Networks(DNNs) have been widely used in many real-world systems such as face recognition and object detection [31, 32, 36, 48]. However, it is well known that DNNs are vulnerable to adversarial examples [16, 35]. Adversarial examples can be crafted by adding small perturbations to the clean inputs, rendering the DNNs' outputs incorrect. Such vulnerabilities could result in severe safety problems when deploying DNN-based systems. This has become a hot research topic recently [7, 11, 23, 26–28].

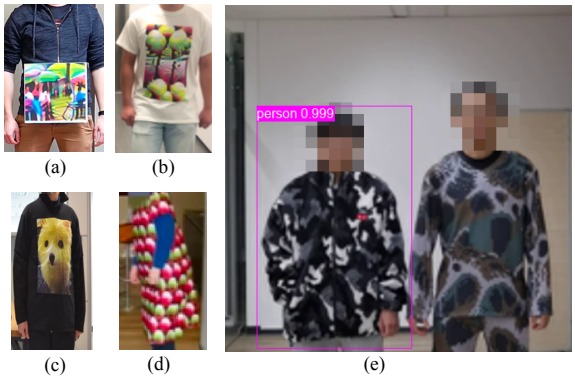


Figure 1. Visualization of several adversarial clothes. (a) Adversarial patch [38]. (b) Adversarial T-shirt [43]. (c) Naturalistic patch [18]. (d) Adversarial Texture [19]. (e) Left: daily camouflage texture; Right: our adversarial camouflage texture.

Adversarial examples were first identified in the digital world. However, adversarial examples also exist in the physical world, posing more risks in real-world scenarios. Recently, many works [1, 12–14, 33, 39–41, 45–47] have designed *physical adversarial examples* to deceive DNNs in the real world. Among them, hiding persons [18–20, 38, 42, 43] from DNN-based object detectors is especially challenging due to the difficulties of modeling non-rigid object surfaces (i.e., clothes). Most works [18, 20, 38, 42, 43] print

*Equal contribution. †Corresponding author.

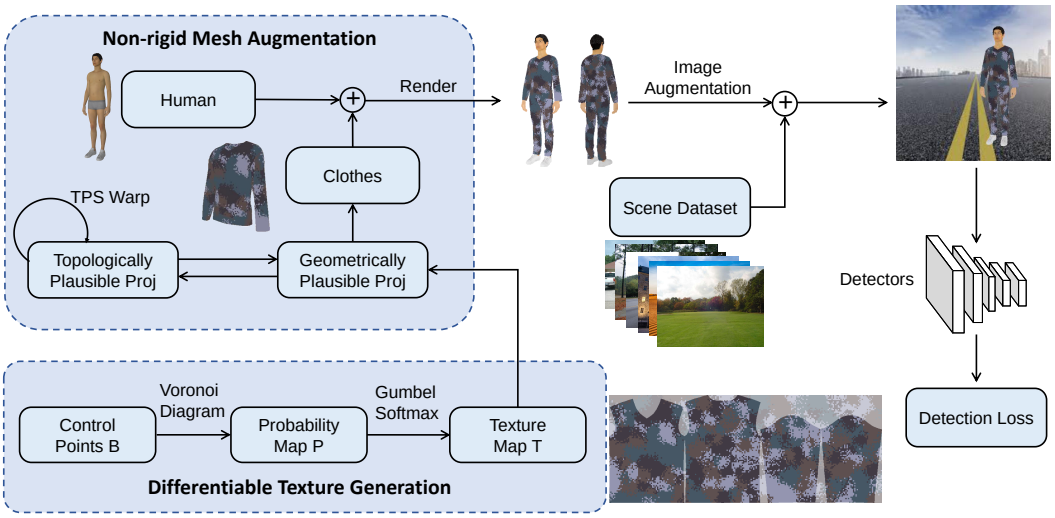


Figure 2. The training pipeline of the adversarial camouflage textures.

adversarial patches on the front side of clothes to hide people from being detected. We call them *patch-based adversarial examples*. These patches are usually conspicuous to humans, making the clothes look strange and easily noticeable by human observers. Efforts have been put on making the adversarial patches more natural-looking [12, 18, 40]. However, these patch-based adversarial clothes can only attack object detectors at a narrow range of viewing angles (i.e., when the camera faces the front of the person). To attack the detector at a wider range of viewing angles, one may print the adversarial patches everywhere on the clothes, which would make the clothes unnatural-looking again. For example, a dog-head-like patch on the front of a T-shirt is natural, but putting this patch everywhere on the T-shirt would make the T-shirt look weird.

Another way to craft physical adversarial examples is to design the textures on the surface of the target objects [1, 13, 19, 39, 44], i.e., crafting *texture-base adversarial examples*. Unlike patch-based adversarial examples, texture-based ones are usually adversarially effective at multiple viewing angles. They are mostly optimized via 3D modeling or using clone networks, and printed on the surface of rigid objects such as turtles [1] and cars [13, 39, 44]. However, it is much harder to realize the 3D textures of non-rigid objects like humans and clothes in the physical world while maintaining their adversarial effectiveness, since there is a huge gap between a 3D human model and a real-world person. To circumvent this difficulty, Hu et al. [19] propose to craft texture-based adversarial clothes by extending patches into textures with repetitive patterns, which does not require 3D modeling. However, their textures are very conspicuous to humans, and obtaining natural-looking textures can be difficult under the constraint of repetitive patterns.

In this paper, we propose a 3D modeling pipeline to produce natural-looking adversarial clothes that are physically realizable and can hide people at multiple viewing angles. Specifically, we craft adversarial camouflage texture (AdvCaT) patterns and apply them on clothes. We choose camouflage texture patterns mainly because they are typical texture patterns widely used in daily clothes, therefore making the clothes more natural-looking In order to make the texture patterns more generalizable when applied to deformed and unseen 3D models, we propose a novel 3D augmentation method combining topologically plausible projection (TopoProj) and thin plate spline (TPS) [3,10,37,43] for non-rigid objects such as clothes.

We optimized several AdvCaT patterns to attack widely used detection models, including YOLOv3 [31], Faster RCNN [32], and deformable DETR [48], and applied the texture patterns on clothes in the physical world. See Fig. 1 for the visualization of our adversarial clothes compared with others. Experiments showed that our adversarial clothes could evade different detectors at multiple viewing angles. A subjective test experiment indicated that the naturalness score of our adversarial clothes covered with AdvCaT is significantly higher than other adversarial clothes and close to daily clothes.

2. Related Work

Early works [7, 16, 23, 35] found that adversarial examples crafted by adding small digital adversarial perturbations on the clean inputs can mislead the DNNs. Some adversarial examples can also be crafted in the physical world to attack different DNN models, including image classification models [1, 5, 14, 33, 45] and detection models [8, 18, 20, 24, 34, 38, 41–43, 47]. Among these works,

patch-based and texture-based attacks are typical ways to craft physical adversarial examples.

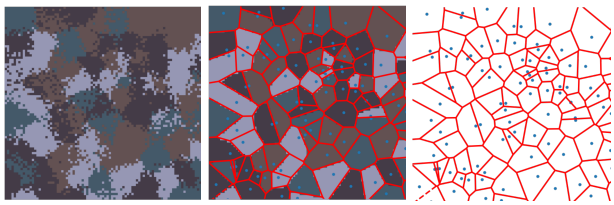
Patch-based attacks [1, 5, 13, 14, 18, 20, 20, 33, 38, 39, 42–44] usually optimize patches and put them on the target objects, and therefore can only work at a narrow range of viewing angles. These works produce different adversarial objects, including glasses frames [33], road signs [14, 15], cars [39, 40, 44] and clothes [18–20, 20, 38, 42, 43]. Among them, hiding persons from object detectors is especially challenging since the adversarial patches on the clothes can be heavily deformed due to their non-rigidity [43]. On the other hand, these adversarial patches are usually conspicuous to humans. To this end, Duan et al. [12] and Wang et al. [40] introduce additional losses to make the adversarial patches less conspicuous. Hu et al. [18] propose to produce more natural-looking patches with GANs [4, 22].

Texture-based attacks [1, 19, 39, 44, 46], on the other hand, optimize the textures on the surface of the target objects to craft physical adversarial examples. Covered with adversarial textures, the object usually can deceive DNNs at multiple viewing angles. These works mainly use 3D modeling or clone networks to optimize textures for rigid objects. Athalye et al. [1] introduce the Expectation over Transformation (EoT) method and produce an adversarial 3D-printed turtle. Zhang et al. [44] and Wang et al. [39] design vehicle camouflage for multi-view adversarial attacks. Hu et al. [19] propose adversarial textures with repetitive structures for non-rigid clothes.

3. Adversarial Camouflage Texture Patterns Generation

In this section, we present the pipeline of generating adversarial camouflage texture (AdvCaT) that can be applied to clothes. As shown in Fig. 2, we adopt 3D meshes to model humans and clothes and define the surface of the clothes according to their 2D texture maps with UV coordinates. We propose two critical techniques to optimize adversarial camouflage texture clothes. The first is to parameterize the camouflage textures on the 2D texture maps with Voronoi diagram [2] and Gumbel softmax trick [21, 25]. The second is to apply a realistic deformation on the 3D meshes with the topologically plausible projection (TopoProj). We render the foreground photos of a 3D person wearing a T-shirt and a trouser using a differential renderer [29]. The foreground photos are synthesized with background images sampled from a scene dataset. Finally, we feed the synthesized photos into the victim detector and minimize an adversarial loss to optimize the parameters of camouflage patterns.

In what follows, we first introduce the differential generation of AdvCaT. Next, we present the novel 3D deformation which can be used to augment the meshes during training to boost the generalizability of the optimized textures.



(a) (b) (c)

Figure 3. (a) Camouflage texture. (b) Color cluster regions. The region of each color cluster can be approximately represented by polygons. (c) Voronoi diagram. The blue points are the control points, and the red lines are the boundaries of the regions.

Finally, we elaborate the loss functions for optimization.

3.1. Differentiable Generation of Camouflage Textures

Camouflage patterns are originally designed for concealing people in the wild and have now become typical textures of ordinary clothes. There are a few kinds of common camouflage patterns. Among them, we choose to imitate a specific type called digital camouflage patterns that consists of small rectangular pixels. As shown in Fig. 3a, digital camouflage patterns are typically irregular in shape and consist of a limited number of colors.

We noticed that the pixels of the camouflage patterns are locally aggregated as clusters, each of which approximately covers a polygon region. See Fig. 3b for illustration. Inspired by the polygon generation ability of Voronoi diagram [2], we use a soft version of Voronoi diagram to generate the cluster regions of the camouflage pixels.

Polygon generation with Voronoi diagram. A Voronoi diagram is a partition of a plane into multiple regions [2]. Each region is controlled by a point, consisting of all the pixels closer to the corresponding control point than to any other point (see Fig. 3c). In this way, the locations and shapes of the polygons can be parameterized by the coordinates of the control points.

However, the locations and shapes of the polygons are not differentiable with respect to the coordinates of the control points if we directly apply this rule. Therefore, we define a soft version of Voronoi diagram by introducing probabilities for each pixel. Suppose the texture map only consists of several discrete colors in a color set $\mathcal{C} = \{c_i = (R_i, G_i, B_i) | i = 1, \dots, N_C\}$. Then, N_P independent control points are assigned to each color, with coordinates $\{b_{ij} \in \mathbb{R}^2, i = 1, 2, \dots, N_C, j = 1, 2, \dots, N_P\}$. For each pixel with coordinates x on the texture map, we assign a discrete distribution $\mathcal{P}^{(x)}$ to describe its probability

of coloring with $\{c_i\}$:

$$p_k^{(x)} = \frac{w_k^{(x)}}{\sum_{i=1}^{N_C} w_i^{(x)}}, k = 1, \ldots, N_C, \quad (1)$$

$$w_i^{(x)} = \sum_{j=1}^{N_P} \exp\left(-\frac{\|x - b_{ij}\|_2}{\alpha}\right), \quad (2)$$

where $p_k^{(x)}$ is the sampling probability of color c_k. According to Eq. (2), the probability of a pixel x colored by c_i increases as it gets closer to a control point b_{ij}. The parameter α is the smoothing radius of the Voronoi diagram. When α approaches zero, the summation in Eq. (2) will be dominated by the closest control point to x, therefore the color of x will be deterministic, which resembles the original hard version of Voronoi diagram. In practice, we define a probability map $\mathcal{P}$ with size $N_C \times H \times W$ for all the pixels on the texture map. We further smooth the probability map $\mathcal{P}$ by a uniform smoothing kernel $\mathcal{S} = \frac{1}{m^2}\mathbb{1}_{m\times m}$ of size $m \times m$. The smoothed probability map is then computed by a convolutional operation: $\mathcal{P}' = \mathcal{P} * \mathcal{S}$.

Sampling discrete colors by Gumbel softmax. Following the procedure stated above, we assign each pixel x on the texture map to a discrete distribution $\mathcal{P}^{(x)}$ guided by a Voronoi diagram, while each pixel should be assigned with a specific color $c^{(x)}$ in the end. However, directly sampling according to $\mathcal{P}^{(x)}$ is not differentiable with respect to $p_i^{(x)}$. Alternatively, using softmax function directly to blend all the colors certainly can not produce discrete colors. Therefore, we leverage the Gumbel-softmax [21,25] reparameterization trick to approximate the discrete sampling process.

Suppose $g_i \sim \text{Gumbel}\,(0, 1)$ are i.i.d. random variables drawn from the standard Gumbel distribution. Given the discrete distribution $\mathcal{P}^{(x)}$, we can equivalently draw the color $c^{(x)}$ by c_k, where

$$k = \arg\max_i (g_i + \log p_i^{(x)}). \quad (3)$$

The equivalency is guaranteed [25] by

$$\Pr(k = i) = p_i^{(x)}. \quad (4)$$

Since the argmax operation is still non-differentiable, we instead use a softmax estimator [21] to approximate it, such that the color $c^{(x)}$ is calculated by

$$c^{(x)} = \sum_{i=1}^{N_C} c_i \cdot \text{Softmax}\left(\frac{g_i + \log p_{c_i}^{(x)}}{\tau}\right), \quad (5)$$

where τ is the temperature coefficient. We have $\lim_{\tau\to 0} c^{(x)} = c_k$. Finally, each pixel x on the texture map T will be colored with $c^{(x)}$.

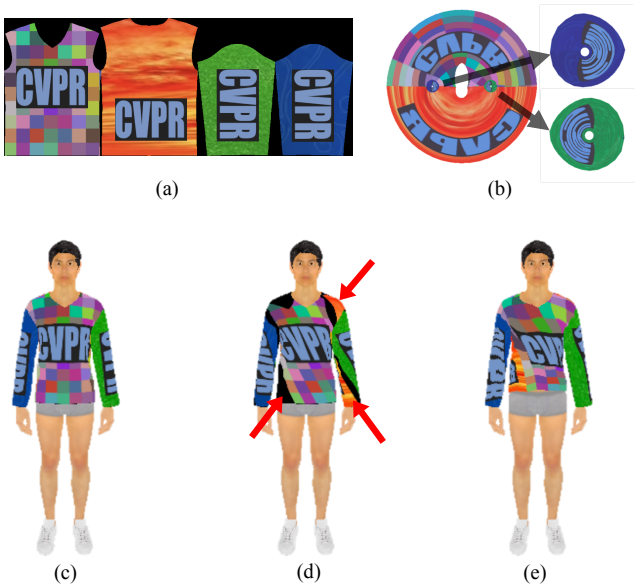


Figure 4. Visualization of the texture augmentation. (a) The texture map of a T-shirt mesh. It is geometrically plausible. (b) A texture map that is topologically plausible, where each point's neighbors in the 2D projection correspond precisely to its neighbors in the 3D mesh. (c-e) Rendered images with different warp methods on the texture map. (c) No warp. (d) Applying a mild shear strain along the texture map's vertical axis. The red arrows indicate pixels that appear at wrong places on the rendered image. (e) Our texture warp base on TopoProj.

In order to enlarge the optimization space, we replace the random seed g_i with a variable g_i'. Since the random seed g_i can be equivalently sampled by inverse transform sampling $g = -\log(-\log(u))$ with $u \sim \text{Uniform}\,(0, 1)$, we define the variable g_i in Eq. (5) as

$$g_i = -\log(-\log(\lambda \cdot u_i^{(\text{fix})} + (1 - \lambda) \cdot u_i^{(\text{train})})), \quad (6)$$

where $u_i^{(\text{fix})} \sim \text{Uniform}\,(0, 1)$ is fixed during the whole training process, and $u_i^{(\text{train})}$ is a trainable variable clipped in range $[0, 1]$. The hyper-parameter $\lambda \in [0, 1]$ controls the ratio of the trainable variables. Putting together, we update the coordinates $\{b_{ij}\}$ and the trainable variable $\{u_i^{(\text{train})}\}$ jointly during optimization.

3.2. Non-rigid Mesh Augmentations

According to Expectation over Transformation (EoT) [1, 33], one can improve the robustness and the generalizability of the physical adversarial examples by applying multiple digital transformations that simulate physical transformations as augmentations during optimization. In order to efficiently simulate the physical warps and movements of the clothes, we apply two augmentations on 3D meshes before applying regular 2D augmentations on the final im-

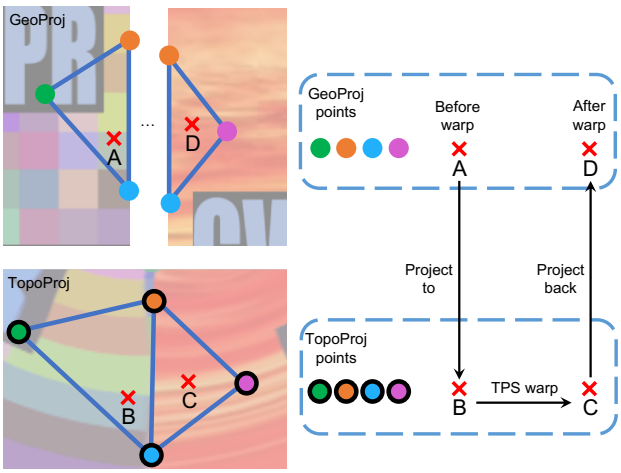


Figure 5. Illustration of the warping at the edge of two pieces. A GeoProj and a TopoProj with textures are shown on the left panel. The two pieces are far away on GeoProj but next to each other on TopoProj. Two triangle elements with blue solid lines at the eadges of the pieces are next to each other on TopoProj. The vertices on TopoProj and GeoProj with the same color are the projections of an identical vertex on the 3D mesh.

ages. The first augmentation aims to warp the texture map of the clothes meshes based on *topologically plausible projections (TopoProj)*. The second augmentation is applied on the mesh vertices' coordinates by 3D Thin Plate Spline (TPS) [37].

Texture warping based on topologically plausible projection. We first obtain the texture maps and UV coordinates of the clothes by Clo3D software[1]. The clothes on the 3D person models are created by pieces of flattened cloth identical to the texture maps. See (Fig. 4a) for an example of a T-shirt mesh's texture map. The local distances (within triangular elements) of the mesh vertices according to 3D coordinates are thus consistent with their local distance on the 2D texture map. Therefore, we call the texture map a *geometrically plausible projection (GeoProj)*. According to this local-distance-preserving property, we can produce the final clothes similar to the 3D simulated ones (Fig. 4c) by printing the texture maps on fabric materials in the real world.

It is challenging to simulate the physical transformations merely by warping on the texture map, since the warping may result in transformations that are far away from the physical ones. As an example, Fig. 4d shows a rendered image after applying a mild shear strain along the vertical axis in Fig. 4a. Plenty of pixels on the image appear at wrong place, e.g., the black background pixels appear on the front side of the clothes, and orange backside pixels appear on the sleeves. The reason is that the coordinates of the points on the texture map cannot reflect their topological relations in the 3D mesh. For example, the bottom-left corner of the T-shirt's front side should be connected to the bottom-right corner of the T-shirt's backside, while the corresponding pixels are far away on the texture map.

To address this problem, we propose a novel warping technique based on the TopoProj (Fig. 4b), which resembles the physical transformation of the clothes (Fig. 4e). A TopoProj is a projection of the mesh vertices that preserve the topological relations between vertices, which allow the pixels appear at reasonable place after the warping. See *Supplementary Material* for the generation process of the TopoProj. However, we can not simply replace the GeoProj with the TopoProj, since it brings difficulties in physical realization: the local distances will no longer be consistent with that of 3D meshes, i.e., we cannot print such patterns and tailor them to produce the final clothes. Moreover, the inconsistency of the local distance will result in extremely uneven resolution of the textures. Therefore, we leverage both GeoProj and TopoProj when applying the warping.

During the original rendering [29], each pixel of the final image corresponds to a certain light path that passing through the camera. The light path may have single or multiple intersections with some triangle elements of the 3D mesh, yet we only consider the closest intersection to the camera. The barycentric coordinate of the intersection in the triangle elements thus can be calculated. Since each vertex of the triangle elements has its correspondent on the texture map, one can calculate the correspondent of the intersection point on the texture map according to its barycentric coordinate. The rendered color of the pixel thus can be calculated according to the position of the intersection point on the texture map.

In order to assign new colors to the pixels of the warped image, we apply additional projections on the coordinates of the intersection points during the rendering. Fig. 5 illustrates the warping pipeline. As mentioned, GeoProj and TopoProj are two projections of all the vertices in the 2D plane for a 3D mesh. For a point in a triangle element, we define a corespondent in each projection, whose position is determined by its barycentric coordinate. The barycentric coordinate is calculated via the original rendering, which is the same in GeoProj and TopoProj. Specifically, we describe the warping process in five steps: (1) given an intersection point A (the red cross on the left piece in Fig. 5) on the GeoProj with its barycentric coordinates; (2) find its correspondent B on TopoProj based on the barycentric coordinates; (3) warp the corresponding point by 2D Thin Plate Spline (TPS) [3, 10] method and get point C; (4) compute the new barycentric coordinates for the warped point C on TopoProj (may be in a new triangle element); (5) find its correspondent D on GeoProj according to the new barycentric coordinates, and compute the color of point D by inter-

[1] https://www.clo3d.com/

polating the texture map. The process is applied on all the pixels of the image.

The TPS warping in step (3) depends on a set of control points (See *Supplementary Material* for the details). We uniformly sample the polar coordinates of each control point with a range of $[-\epsilon_r, \epsilon_r]$ and $[-\epsilon_t, \epsilon_t]$ for the radius and angle respectively.

Vertex augmentation by 3D TPS. We also applied augmentation directly on the 3D vertex coordinates of the meshes by 3D TPS [37]. The vertex coordinates are perturbed according to a set of control points. We uniformly perturb the control points in range $[-\epsilon_{\text{TPS}}, \epsilon_{\text{TPS}}]$. See *Supplementary Material* for the visualization.

Together with the texture warping, we apply mesh augmentations during optimization to reduce the gap between the 3D meshes and the real-world ones.

Other augmentation. Since the colors will change when they are printed on fabric materials, we calibrate the digital color on the texture maps to the physical color following [43]. See *Supplementary Material* for the details. During 3D rendering, we sample the viewing angles of the camera adaptively according to the mean confidence score of the target bounding boxes, where the angles with higher scores are more likely to be sampled. We also choose the simulated lights from ambient lights, directional lights, and point lights uniformly at random. Moreover, we apply other image augmentations on the rendered images following previous works [38, 41], such as randomizing the scales, positions, contrast and brightness.

3.3. Adversarial Loss Function

In this section, we present the objective functions for attacking detectors.

Detection loss. Object detectors predict bounding boxes with confidence scores. Since our goal is to evade the detectors from detecting humans, we minimize the confidence score of the person class in the box which has the maximum Intersection over Union (IoU) score with the ground truth. For each input x, suppose that the victim detector $\mathcal{D}$ outputs a set of bounding boxes $b_i^{(x)}$, each with a confidence $\text{Conf}_i^{(x)}$. We define the detection loss as

$$\mathcal{L}_{det} = \sum_x \text{Conf}_{i^*}^{(x)}, i^* = \arg\max_i \text{IoU}(\text{gt}^{(x)}, b_i^{(x)}), \quad (7)$$

where $\text{gt}^{(x)}$ stands for the ground truth bounding box of the foreground person on the input image x.

Concentration loss for camouflage texture. In order to increase the stability of the camouflage texture generation, we prevent the polygons from being too small by introducing a concentration loss that encourages control points to move away from each other:

$$\mathcal{L}_{con} = \sum_{j=1}^{N_C} \sum_{1 \le k_1 < k_2 \le N_P} \exp\left(-\frac{\|b_{jk_1} - b_{jk_2}\|^2}{\sigma^2}\right), \quad (8)$$

where σ is a constant.

The total adversarial loss for minimization is

$$\mathcal{L} = \mathcal{L}_{det} + \alpha_{con}\mathcal{L}_{con}, \quad (9)$$

where α_{con} is the weight between the two losses.

4. Experiments

4.1. Experimental Setup

Subjects. Three actors (age mean: 26.3; age range: $25-27$; height range: $175-178$ cm) are recruited to collect physical test data. We also recruited 93 subjects (age mean: 30.2; age range: $18-57$) to evaluate the naturalness score of different clothes. The recruitment and study procedures were approved by the Department of Psychology Ethics Committee, Tsinghua University, Beijing, China.

Baseline methods According to the previous work [19], the Attack Success Rates (ASRs) of the adversarial clothes printed with isolated patches will drop catastrophically when the viewing angle changes. Printing repetitively tiled patches on the clothes is helpful to prevent the ASRs from dropping. Therefore, we tiled the patches produced by patch-based attacks for fair comparison. We mainly evaluated three patch-based attacks *AdvPatch* [38], *AdvTshirt* [43], *NatPatch* [18], and a texture-based attack, *AdvTexture* [19]. We also evaluate *RandColor*, a random texture with random colors in a lattice, and *RandCaT*, a random camouflage texture pattern.

See *Supplementary Material* for the datasets, target detectors, evaluation metric and the implementation details.

4.2. Naturalness Score by Subjective Evaluation

Following Hu et al. [18], we conducted a subjective evaluation on the naturalness score of the adversarial clothes. For a fair comparison, we applied different patterns on an identical garment model using FAB3D[2]. We showed eight pictures of different T-shirts (Tab. 1) aggregated on a scrollable page in random orders to the subjects and required them to give a naturalness score for each picture using a 7-level Likert scale (1 = not natural at all to 7 = very natural).

As shown in Tab. 1, the naturalness score of AdvCat targeting Faster RCNN (4.89) is significantly higher than those of the other five adversarial patterns ($p < 0.001$, student's t-test), and is close to the scores of the control group with common textures (the second column, 6.08 and the third column, 5.05 in the table).

[2] https://tri3d.in/

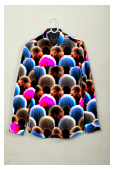
Images								
Score	6.08 ± 1.00	5.05 ± 1.39	2.05 ± 1.06	1.75 ± 1.09	1.72 ± 0.98	1.69 ± 0.95	2.54 ± 1.23	4.89 ± 1.39
Source	common texture	common camouflage	AdvPatch [38]	AdvTshirt [43]	AdvTexture yolo [19]	AdvTexture faster [19]	NatPatch [18]	AdvCaT (ours)

Table 1. Subjective test using a 7-level Likert scale (1 = not natural at all to 7 = very natural).

Method	IoU0.01	IoU0.1	IoU0.3	IoU0.5
RandColor	0.13	0.13	0.13	0.17
RandCaT	1.02	1.02	1.04	1.10
AdvPatch	69.33	72.27	75.80	85.97
NatPatch	42.47	43.66	45.41	67.40
AdvTexture	1.44	21.73	87.05	**99.98**
AdvCaT (ours)	**95.18**	**99.21**	**99.40**	99.52

Table 2. ASRs/% of different methods targeting Faster RCNN in the digital world.

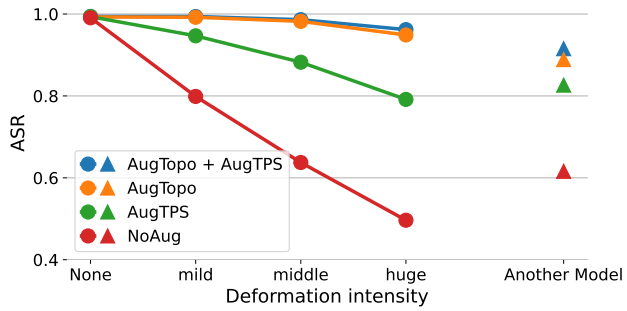


Figure 6. ASR targeting Faster RCNN versus deformation intensity for different augmentation strategies during training. *Another Model* denotes ASRs of the patterns when they are applied on an unseen model without any 3D deformation.

4.3. Digital World Results

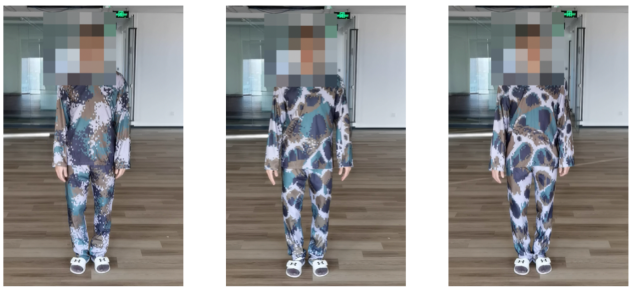
Evaluation with different IoU threshold. We noticed that the IoU threshold τ_{IoU} during evaluation is usually set to 0.5 according to previous works [19, 38] since they mainly evaluate their adversarial patches or textures on the datasets that contains multiple people, e.g. Inria dataset [9]. On such datasets, a relatively high threshold can prevent confusing the boxes of overlapping objects. However, the high threshold could result in an overestimation of the attack's effectiveness. The target detector may output a considerably large bounding box with an IoU score smaller than the threshold, which still provides strong evidence of having detected the person. See *Supplementary Material* for examples. Therefore, we evaluated the ASRs with different IoU thresholds $0.01, 0.1, 0.3$, and 0.5. See Tab. 2 for the ASRs of different methods targeting Faster RCNN. According to Tab. 2, the ASR of the AdvTexture is slightly higher than our method with an IoU threshold of 0.5, while it decreases significantly when the IoU threshold decreases. On the contrary, the ASRs of our method, AdvCaT is consistently high with different IoU threshold, even when the threshold equals to 0.01, which indicates the strong adversarial effectiveness of AdvCaT. Since an IoU threshold of 0.01 is too small that may introduce undesirable noises, we report the ASRs with IoU threshold 0.1 in the rest of our paper unless explicitly mentioned.

Ablation Study of Augmentation Strategies In order to investigate the effect of different 3D model augmentation strategies on the generalizability of the optimized patterns, we optimized four AdvCaT patterns targeting Faster RCNN with different 3D model augmentation strategies The first pattern used no augmentation on 3D meshes, denoted by *NoAug*. The second pattern used 3D TPS augmentation, denoted by *AugTPS*. The third pattern used topologically plausible projection, denoted by *AugTopo*. The final one, *AugTPS+AugTopo*, incorporated both augmentations. Moreover, we used four different intensity of 3D mesh deformation during evaluation, which are denoted by *None* (no deformation), *Mild* ($(\epsilon_r, \epsilon_t, \epsilon_{\text{TPS}}) = (0.1, 50, 0.15)$), *Middle* ($(\epsilon_r, \epsilon_t, \epsilon_{\text{TPS}}) = (0.1, 65, 0.22)$), and *Huge*$(\epsilon_r, \epsilon_t, \epsilon_{\text{TPS}}) = (0.1, 80, 0.3)$, respectively. Note the hyper-parameter used during training was the same as *Mild*. See Fig. 6 for the ASRs of the patterns with different augmentation strategies and deformation intensities. The ASRs of the patterns applied on a new 3D person without any 3D deformations are also plotted in the figure.

As shown in Fig. 6, the ASR of *NoAug* drops significantly when the deformation intensity increases, which implies its catastrophic drop of the adversarial effectiveness in the real world. Using 3D TPS alone can be better, but it still suffers from a considerable drop under huge deformation

Detectors	Faster RCNN	YOLOv3	DDETR
Random	0.85	3.31	5.76
AdvCaT	99.36	94.53	88.77

Table 3. ASRs/% targeting different detectors in the digital world.



Random
ASR=0.00 %

AdvCaT w/o aug
ASR=19.27 %

AdvCaT w/ aug
ASR=85.94 %

Figure 7. Adversarial clothes covered with different patterns in the physical world.

intensity. The ASRs of *AugTopo* that only use TopoProj are high even when the deformation intensity is huge. Combining 3D TPS with TopoProj is slightly better than only using TopoProj. The ASRs of different strategies evaluated on a new unseen 3D model are consistent with the previous observation, which indicates the good generalization ability of the pattern using both augmentations.

Attacking different detectors. We optimized camouflage patterns to attack different detectors including YOLOv3 [31], FasterRCNN [32] and deformable DETR [48] and show their ASRs in Tab. 3. We also used the trained patterns to attack other detectors to study their transferability. The ASR of the AdvCaT trained to attack Faster RCNN was relatively high when targeting MaskRCNN (92.22%) and Deformable DETR (65.11%), but relatively low when targeting YOLOv3 (23.26%). See *Supplementary Material* for the visualization of these AdvCaTs and the full transfer study.

Parameter sensitivity. We varied the value of λ in Eq. (6) during optimization. When λ increased, The ASR increased, while the AdvCaT became less like a camouflage pattern. In addition, we optimized AdvCaT with different styles by using various color combinations c_i, all of which achieves high ASRs targeting Faster RCNN. See *Supplementary Material* for details of these experiments.

4.4. Physical World Results

We produced three clothes covered with different AdvCaT patterns in the physical world. We cropped the different parts of the clothes from the texture map and printed them on fabric materials. These pieces were then tailored

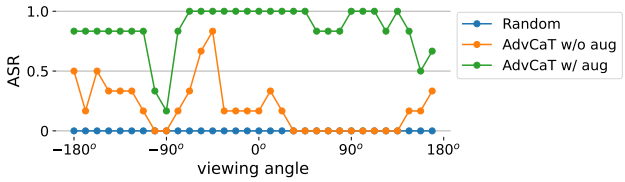


Figure 8. ASRs at different viewing angles in the physical world. The actors were facing the camera when the viewing angle equals to $0°$.

to wearable adversarial clothes. In Fig. 7 we visualized the clothes and presented their ASRs targeting Faster RCNN, where *Random* denotes the clothes covered with random camouflage textures; *AdvCaT w/o aug* and *AdvCaT w aug* denote the clothes that covered with AdvCaT with or without mesh augmentation (i.e. TopoProj and 3D TPS) during optimization, respectively. The ASR of *AdvCaT w aug* (85.94%) was significantly higher than those of *AdvCaT w/o aug* (19.27%) and *Random* (0.00%).

Fig. 8 shows the ASRs at different viewing angles, indicating strong attack ability of the AdvCaT clothes. In addition, we found that our designed clothes were relatively robust to the environment change. When the distance between the actor and the camera was less than 4 m, the ASR stayed high (above 61.5%). See *Supplementary Material* for details of these experiments. We also provide a video demo in *Supplementary Video*.

5. Conclusion

We proposed to optimize clothes textures via 3D modeling to produce natural-looking adversarial clothes that are adversarially effective at multiple viewing angles. The adversarial T-shirt with AdvCaT patterns has a high naturalness score in a subjective test evaluated by a group of subjects. Experimental results indicate that our adversarial clothes can hide people from detectors at multiple viewing angles with high ASRs in the digital and physical world.

Limitations Though the AdvCaT patterns sometimes have a relatively high ASR targeting unseen detectors, their transferability is not universal, since the ASRs targeting some certain detectors are not very good. One can use model ensemble to improve their transferability.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Nos. U19B2034, 62061136001, 61836014).

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Inter-*

national conference on machine learning, pages 284–293. PMLR, 2018. 1, 2, 3, 4

[2] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, sep 1991. 3

[3] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989. 2, 5

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 3

[5] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 2, 3

[6] 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.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (sp)*, pages 39–57. IEEE, 2017. 1, 2

[8] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68. Springer, 2018. 2

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 7

[10] Gianluca Donato and Serge Belongie. Approximate thin plate spline mappings. In *European conference on computer vision*, pages 21–31. Springer, 2002. 2, 5

[11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 1

[12] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1000–1008, 2020. 1, 2, 3

[13] Yexin Duan, Jialin Chen, Xingyu Zhou, Junhua Zou, Zhengyun He, Jin Zhang, Wu Zhang, and Zhisong Pan. Learning coated adversarial camouflages for object detectors. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, pages 891–897, 2022. 1, 2, 3

[14] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 1, 2, 3

[15] Abhiram Gnanasambandam, Alex M Sherman, and Stanley H Chan. Optical adversarial attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 92–101, 2021. 3

[16] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[18] Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, Jun-Cheng Chen, Kai-Lung Hua, and Wen-Huang Cheng. Naturalistic physical adversarial patch for object detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7848–7857, 2021. 1, 2, 3, 6, 7

[19] Zhanhao Hu, Siyuan Huang, Xiaopei Zhu, Fuchun Sun, Bo Zhang, and Xiaolin Hu. Adversarial texture for fooling person detectors in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13307–13316, 2022. 1, 2, 3, 6, 7

[20] Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, Alan L Yuille, Changqing Zou, and Ning Liu. Universal physical camouflage attacks on object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 720–729, 2020. 1, 2, 3

[21] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 3, 4

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 3

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations*, 2017. 1, 2

[24] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019. 2

[25] Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. *Advances in neural information processing systems*, 27, 2014. 3, 4

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016. 1

[27] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015. 1

[28] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The

limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016. 1

[29] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 3, 5

[30] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[31] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 2, 8

[32] 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. 1, 2, 8

[33] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016. 1, 2, 3, 4

[34] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*, 2018. 2

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1, 2

[36] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1701–1708. IEEE Computer Society, 2014. 1

[37] Zhixian Tang, Kun Chen, Mingyuan Pan, Manning Wang, and Zhijian Song. An augmentation strategy for medical image processing based on statistical shape model and 3d thin plate spline for deep learning. *IEEE Access*, 7:133111–133121, 2019. 2, 5, 6

[38] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2, 3, 6, 7

[39] 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. 1, 2, 3

[40] 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. 1, 2, 3

[41] Yi Wang, Jingyang Zhou, Tianlong Chen, Sijia Liu, Shiyu Chang, Chandrajit Bajaj, and Zhangyang Wang. Can 3d adversarial logos cloak humans? *arXiv preprint arXiv:2006.14655*, 2020. 1, 2, 6

[42] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020. 1, 2, 3

[43] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *European Conference on Computer Vision*, pages 665–681. Springer, 2020. 1, 2, 3, 6, 7

[44] Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. Camou: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In *International Conference on Learning Representations*, 2019. 2, 3

[45] Yiqi Zhong, Xianming Liu, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15345–15354, 2022. 1, 2

[46] Xiaopei Zhu, Zhanhao Hu, Siyuan Huang, Jianmin Li, and Xiaolin Hu. Infrared invisible clothing: Hiding from infrared detectors at multiple angles in real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13317–13326, 2022. 1, 3

[47] Xiaopei Zhu, Xiao Li, Jianmin Li, Zheyao Wang, and Xiaolin Hu. Fooling thermal infrared pedestrian detectors in real world using small bulbs. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, 2021. 1, 2

[48] 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. 1, 2, 8