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

1. Supplementary Methods

1.1. Topologically Plausible Projection Generation

See Fig. S1a for the visualization of the vertices of a T-shirt mesh, and Fig. S1b-c for the visualization of a GeoProj and a TopoProj. We generate the topological plausible projection (TopoProj) by a process named *Zipping*. When 3D vertex points are mapped to TopoProj, the connectivity of all point pairs remains unchanged. However, as for GeoProj, only the connectivity of the inner points in each pieces is unchanged, while some separated boundary point pairs are connected according to the 3D mesh. Therefore, we use Zipping to generate TopoProj by connecting the point pairs in GeoProj. See Fig. S1d for an example.

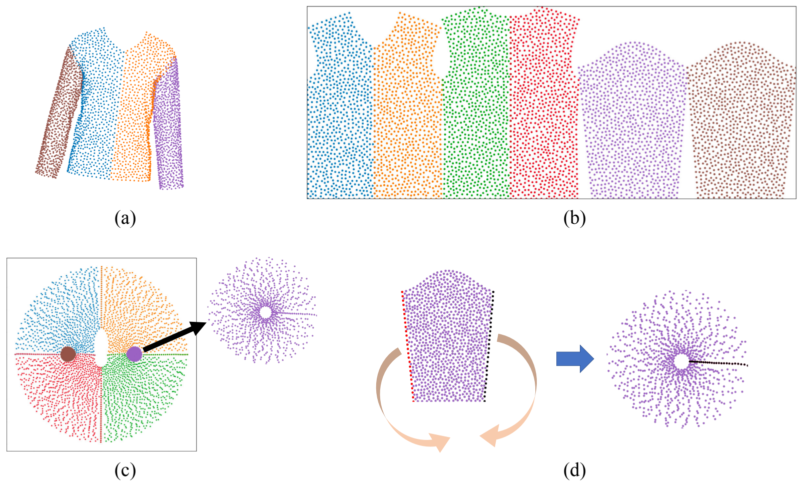


Figure S1. Visualization of different projections. (a) The 3D vertex points of a T-shirt mesh at the front view. (b) A GeoProj. (c) A TopoProj. (d) A TopoProj of a sleeve can be created from its GeoProj by Zipping. Each pair of the points on GeoProj that supposed to be identical are colored red and black, respectively. The points on the left (red) and right (black) boundaries in the left panel are identical according to 3D mesh and therefore have the same coordinates in TopoProj (the right panel).

Zipping is a continuous transformation from GeoProj to TopoProj inspired by physical mechanisms of elastic stress. Intuitively, Zipping is to pull the point pairs (red and black points in Fig. S1d) slowly, until they are overlapped. The triangle elements will be deformed as even as possible, while their chiralities are not supposed to be flipped during zipping. Specifically, we update the coordinates of the points based on carefully designed forces. This force is an inner force caused by the triangle (like a spring) that intended to push the point back to *restore* the triangle's original shape. It restricts the shape of the entire frame from changing too much during Zipping. As illustrated in Fig. S2, the force component $F_{i,k}$ on point i by triangle element k is proportional to the relative displacement $\Delta^{\text{all}}_{i,k}$ of this point in the triangle element. Since each point

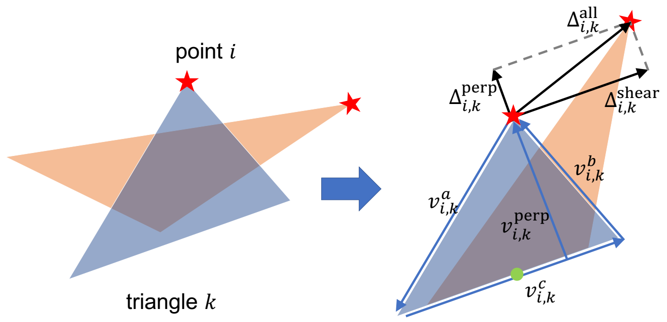

point i
triangle k
$\Delta_{i,k}^{\text{all}}$
$\Delta_{i,k}^{\text{perp}}$
$\Delta_{i,k}^{\text{shear}}$
$v_{i,k}^{a}$
$v_{i,k}^{b}$
$v_{i,k}^{\text{perp}}$
$v_{i,k}^{c}$

Figure S2. Illustration of the forces. The light Salmon orange triangle and the blue triangle denotes the triangle element before and after deformation, respectively. For a specific point, denoted by the red star, we first rotate and translate the triangles to align its bottom edges in the two triangles with overlapped midpoints (colored green). The force is then defined proportional to the displacement vector $\Delta_{i,k}^{\text{all}}$ starting from the point after deformation and ending at the point before deformation.

belongs to multiple triangle elements, the resultant force is

$$F_i = \sum_k F_{i,k} = \sum_k \frac{\Delta_{i,k}^{\text{all}}}{\left\| v_{i,k}^{\text{perp}} \right\|_2}, \tag{1}$$

$$v_{i,k}^{\text{perp}} = \frac{(v_{i,k}^b, v_{i,k}^c)v_{i,k}^a - (v_{i,k}^a, v_{i,k}^c)v_{i,k}^b}{(v_{i,k}^c, v_{i,k}^c)}, \tag{2}$$

where $(\cdot,\cdot)$ is the inner production and $v_{i,k}^a$, $v_{i,k}^b$ and $v_{i,k}^c$ are the edge vectors of the triangle element illustrated in Fig. S2. $v_{i,k}^{\text{perp}}$ is the altitude from the edge $v_{i,k}^c$ to the point i, also illustrated in Fig. S2. We divide the forces by the norm of the altitude $\left\| v_{i,k}^{\text{perp}} \right\|_2$ in Eq. (1) to prevent the triangle element from flipping. Moreover, the points in each point pair (colored red and black in Fig. S1d) are subjected to attractive forces that point to each other. We then iteratively update the coordinates of each point according to the resultant force. Suppose that the resultant force of the point i at step t is $F_i^{(t)}$, and its current coordinate is $x_i^{(t)}$. We update the coordinate by

$$x_i^{(t+1)} = x_i^{(t)} + \beta^{(t)} * F_i^{(t)}, \tag{3}$$

where $\beta^{(t)}$ is an adaptive time interval that prevent the chirality of the triangles from being flipped. Specifically, we have

$$\beta^{(\text{t})} = \gamma \min \left(\mathcal{S}_\beta^{(t)} \cup \{\beta_{\max}\} \right), \tag{4}$$

$$\mathcal{S}_\beta^{(t)} = \left\{ \beta \mid \beta > 0, \text{ and } \exists i, k, \text{ s.t. } v_{i,k}^{\text{perp}} = 0 \text{ when coordinates } x_i = x_i^{(t)} + \beta * F_i^{(t)} \right\}, \tag{5}$$

where we used $\gamma = 0.5$ and $\beta_{\max} = 0.1$ in practice.

With a proper initialization of the points' coordinates (by bending the pieces from GeoProj), we generated the TopoProj for T-shirt and trouser meshes. See the visualization in Fig. S3.

1.2. TPS Warping on TopoProj

We initialize a set of control points and uniformly perturb the polar coordinates of each control point. The warped coordinates of all the other points are calculated according to the control points, as shown in Fig. S4.

1.3. Visualization of 3D TPS

An example of the perturbed mesh is shown in Fig. S5.

1.4. Physical Color Calibration on the 3D Texture Map

As shown in Fig. S6a, we first generate a color palette with $9 \times 9 \times 9 = 729$ different colors in the digital world. We then print it on a piece of cloth, take a picture for them, and extract the corresponding colors. We use a polynomial regression

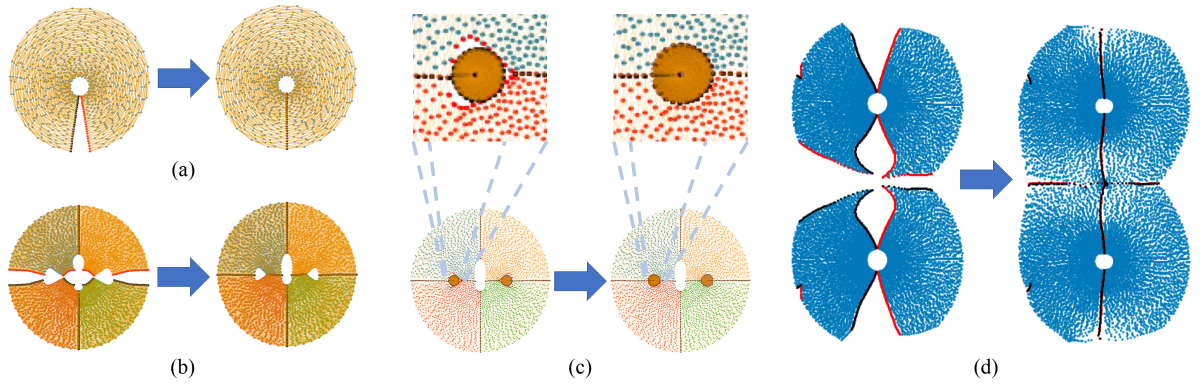


Figure S3. Visualization of Zipping when generating different parts of the clothes. The left panels of all the sub-figures are the initial states of the points, and the right panels are the points after Zipping. (a) One of the sleeves. (b) Zipping the front side of the T-shirt with the back side. (c) Zipping the sleeves with the front side and back side of the T-shirt. (d) The trousers.

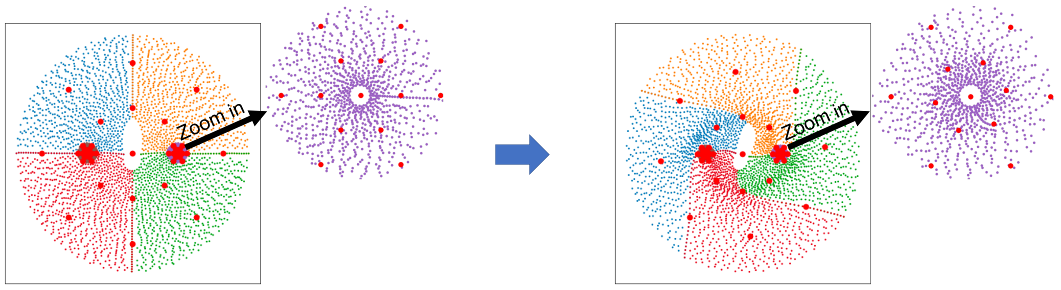


Figure S4. The warping on TopoProj of the T-shirt mesh by 2D TPS. Left: the TopoProj before warping; Right: the TopoProj after warping. The large red dots denote the control points for 2D TPS warping. We zoom in the locations around the sleeves to see the details

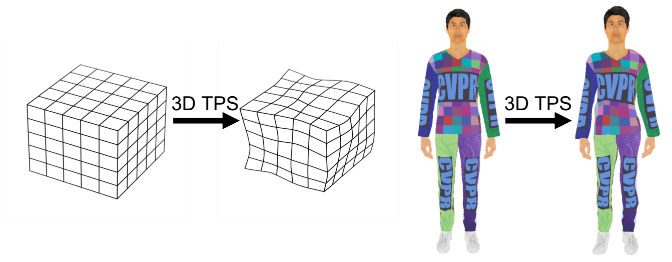


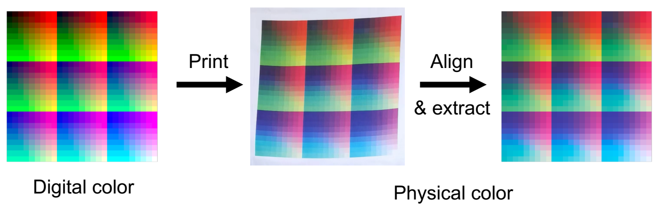
Figure S5. Visualization of 3D TPS warping. We apply 3D TPS warping on the vertex coordinates of the target mesh. Left: warping a cube; Right: warping a person with clothes.

model with six degrees to fit the color transformation. Specifically, suppose the original digital color is $x = (R, G, B)$ and the final physical color is $y^* = (y_1^*, y_2^*, y_3^*) = (R^*, G^*, B^*)$. The regression model is

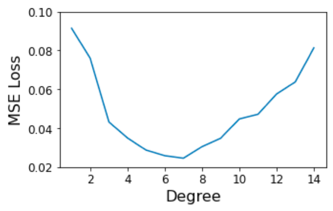
$$y_i = \sum_{a_1,a_2,a_3} w^i_{a_1,a_2,a_3} x_1^{a_1} x_2^{a_2} x_3^{a_3}, i = 1, 2, 3, \tag{6}$$

$$a_1 + a_2 + a_3 \leq \mathrm{d}, \tag{7}$$

where a_1, a_2, a_3 and the degree d are non-negative integers. A linear regression model is then applied to fit the polynomial features to y^* with coefficients $w^i_{a_1,a_2,a_3}$. In order to choose a optimal degree d, we randomly divide the 729 color pairs into training and validation sets, each containing 50% of the pairs. We fit the model on the training set, and calculate the MSE loss on the validation set. As shown in Fig. S6b, the MSE loss is the smallest when the degree is around 6. Therefore, we choose $d = 6$.



(a)



(b)

Figure S6. Physical color calibration. (a) The pipeline of the calibration. The color palette consists of 729 different colors. (b) The MSE loss of the fitting v.s. degree during validation. The MSE loss at each degree is averaged over 20 partitions.

2. Supplementary Results

2.1. Experimental Setup

Datasets and Target Detectors. We collected 506 background images in total from Google Search for 3D rendering. See Sec. 2.2 for some examples. We split them randomly into two sets for training and testing respectively, where the training set consists of 376 images and the test set consists of 130 images.

We attacked three different detectors including YOLOv3 [7], Faster RCNN [8], and Deformable DETR [10] under white box setting. We also attack other detectors including YOLOv2 [6], Mask RCNN [2], and DETR [1] to evaluate the transferability of the adversarial camouflage textures.

Evaluation Metric. We extracted the bounding boxes from the output of the target detector on each input image and filtered out the boxes of which IoU scores with the ground truth box are lower than a specific IoU threshold τ_{IoU}. The threshold was set to 0.1 in our paper except for Tab. 2 in the main text. An image is regarded as an adversarial success example as long as the confidence scores of all the left boxes are lower than a confidence threshold $\tau_{\text{conf}} = 0.5$. The ASR is defined as the proportion of the adversarial success examples among all the test images.

Implementation Details.

We fixed the temperature τ in main text Eq. (5) to 0.3 during training and use discrete sampling ($\tau \to 0$) during evaluation and physical evaluation. We set the parameter $\lambda = 0.7$ in main text Eq. (6). During training, we randomly perturbed the 3D models with the hyper-parameters $(\epsilon_r, \epsilon_t, \epsilon_{\text{TPS}}) = (0.1, 50.0, 0.15)$. We optimized the parameters for 600 epochs with Adam [5] optimizer with learning rate 0.001 for the coordinates b_{ij} in main text Eq. (2) and 0.01 for the trainable Gumbel seed u_i^{train} in main text Eq. (6).

For digital evaluation, we rendered the mesh as we did for training, while using backgrounds sampled from the test set. We averaged the ASR over 37 viewing angles ranging from $-180°$ to $180°$. Note that the 3D person model exactly faces the simulated camera when the viewing angle equals to $0°$.

For physical evaluation, we asked three actors to wear the adversarial clothes and turn a circle slowly in front of a camera, which was fixed at $1.55\ \mathrm{m}$ above the ground and $3.0\ \mathrm{m}$ distant from the actor unless otherwise specified. For each actor and each adversarial clothes, we recorded one video indoor and one outdoor. We extracted 32 frames from each video and therefore collected $32 \times 3 \times 2 = 192$ for each clothes. We labeled the ground truth manually and evaluated the ASRs on these frames as we did in digital evaluation.

2.2. Scene Dataset and Synthesized Images

See Fig. S7a for examples of the background images in the scene dataset, and see Fig. S7b for the synthesized images with rendered 3D person meshes as the foreground images. The 3D person meshes were rendered at different viewing angles, and were stuck onto the background images with random scales and positions.

2.3. Visualization of the Bounding Boxes during Evaluation

See Fig. S8 for the examples of the detection results of different patterns. Previous methods were more likely to output some bounding boxes within the area of the foreground image, which had small but non-negligible IoU scores with the ground truth boxes. See Sec. 4.3 in the main text for the ASRs evaluated with different IoU threshold.

(a)

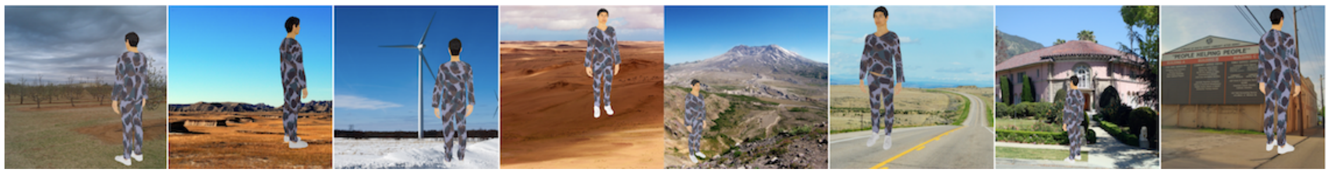

(b)

Figure S7. Visualization of the dataset. (a) The background images. (b) The synthesized images.

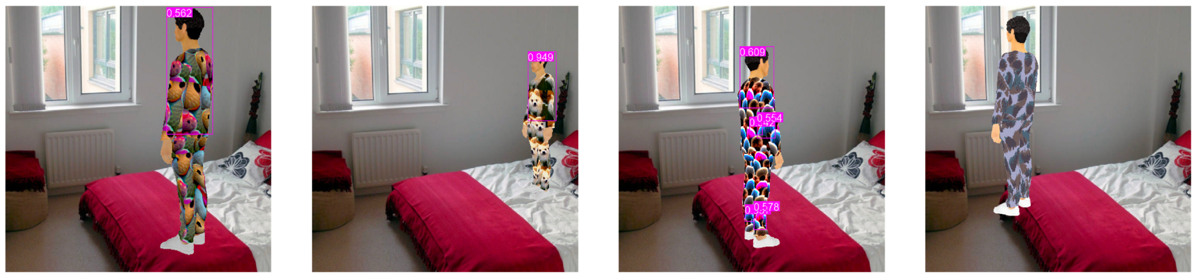

(a) (b) (c) (d)

Figure S8. Visualization of the bounding boxes, with confidence threshold 0.5. (a) AdvPatch [9]. (b) NatPatch [3]. (c) AdvTexture [4]. (d) Ours.

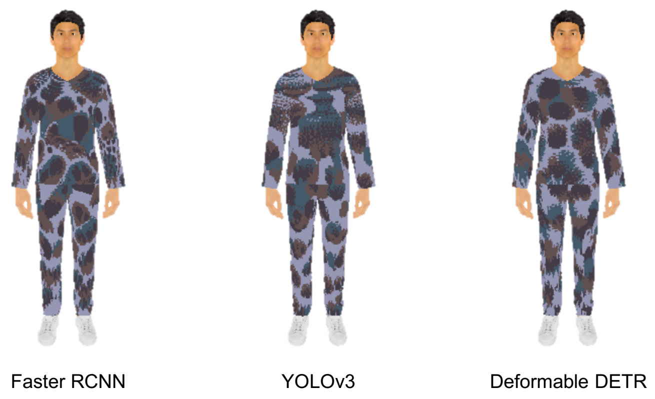


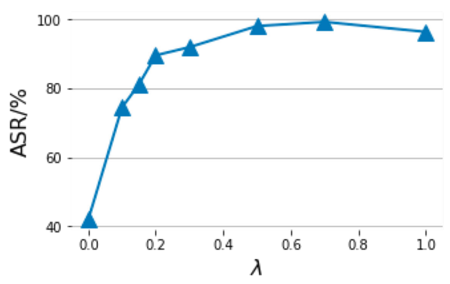
Figure S9. Visualization of the camouflage patterns against different detectors.

2.4. Transfer Study in the Digital World

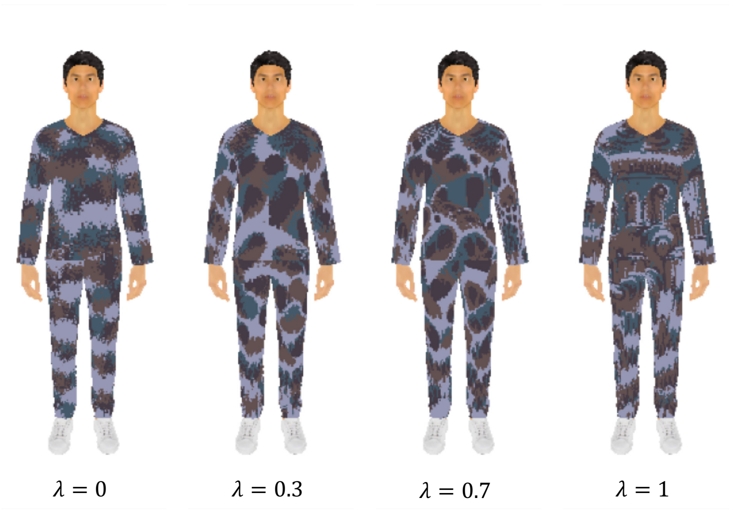
We optimized camouflage patterns against different detectors including YOLOv3 [7], Faster RCNN [8], and Deformable DETR [10]. Fig. S9 shows the camouflage patterns against different detectors. See Tab. S1 for the ASRs of the transfer attacks in the digital world. The ASRs usually drop when the patterns were transferred to attack unseen detectors. In some cases, the ASRs still had high values (source Faster RCNN & target Mask RCNN, 92.22%, source Deformable DETR & targete Faster RCNN, 71.73%).

Source \ Target	Faster RCNN	YOLOv3	Deformable DETR	Mask RCNN	YOLOv2	DETR
Random	0.85	3.31	5.76	0.50	2.10	4.95
Fast rcnn	99.36	28.21	65.11	92.22	18.05	36.05
YOLOv3	23.74	94.59	27.28	12.72	20.33	15.57
Deformable DETR	71.73	23.26	88.77	43.99	16.38	48.19

Table S1. ASRs/% of the transfer attacks in the digital world.



(a)



(b)

Figure S10. Performance of different lambda values. (a) ASR v.s. λ. (b) Visualization of the pattern with different λ values.

2.5. Parameter Sensitivity of λ

We optimized the AdvCaT patterns with different values of λ in Eq.(6) in the main text, which controls the percentage of the trainable variable in the Gumbel seed. As shown in Fig. S10, small λ resulted in poor adversarial effectiveness while large λ resulted in strong adversarial effectiveness. Meanwhile, the AdvCaT pattern with small λ was more like typical camouflage texture patterns, while the pattern with $\lambda = 1$ looked somehow strange. Therefore, we chose $\lambda = 0.7$ as a trade-off between the adversarial effectiveness and the appearance.

2.6. Different Color Combinations of the AdvCaT Patterns

We used different color combinations to produce different styles of the AdvCaT patterns. See Fig. S11 for the visualization. All of the AdvCaT patterns had high ASRs ($> 97\%$) targeting Faster RCNN in the digital world.

2.7. Attack in the Physical World under Different Physical Settings

We evaluated the ASRs of the AdvCaT clothes under different physical settings. See Tab. S2 for the ASRs of the adversarial clothes in the indoor and outdoor scenes. See Sec. 4.1 in the main text for the collection of the physical test set. The ASRs of AdvCaT clothes were high (above 85.4%) both indoor and outdoor. In addition, Fig. S12 shows the ASRs at different distances between the camera and actors. The ASRs stayed high (above 61.5%) when the distance was less than 4 m.

References

[1] 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. 4

[2] 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. 4

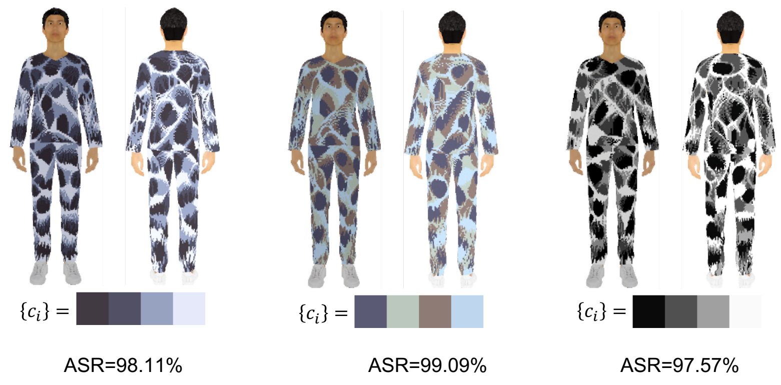


Figure S11. ASRs of the patterns with different color combinations.

	Indoor	Outdoor
Random	0.00	0.00
AdvCaT	86.46	85.42

Table S2. ASRs/% in different environments.

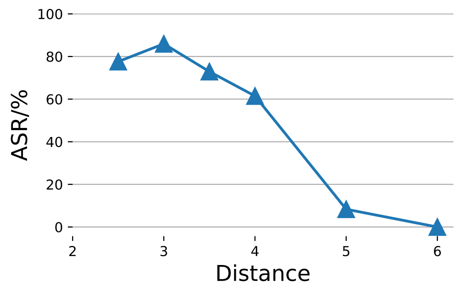


Figure S12. ASRs v.s distances between the camera and actors.

[3] 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. 5

[4] 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. 5

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[6] 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. 4

[7] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 4, 5

[8] 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. 4, 5

[9] 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. 5

[10] 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. 4, 5