Revisiting Adversarial Patch Defenses on Object Detectors: Unified Evaluation, Large-Scale Dataset, and New Insights

Supplementary Material

A. Implementation Details

A.1. Descriptions of Attacks

Transfer-based Self-Ensemble Attack (T-SEA) [7] proposes a black-box attack framework against object detectors, which employs self-ensemble strategies (constrained data augmentation, model ShakeDrop, and patch cutout) to enhance the transferability of adversarial patches. Given the excellent attack performance of T-SEA, we adopt three different patch update strategies from the original work (Adam, mim, and pgd) to derive three attack methods: *T-SEA*, *T-SEA-mim*, and *T-SEA-pgd*.

Adversarial Texture (AdvTexture). [6] presents a generative technique designed for multi-angle adversarial attacks. It initially trains an expandable generator and then optimizes the input latent variable while keeping the generator's parameters fixed, thereby enabling effective manipulation of the generated texture to deceive detectors across various viewpoints. We adopt two patch generation strategies from the original paper, differing in whether a generator is used, resulting in two attack methods: *TC-EGA* and *TCA*.

adversarial patches (AdvPatch) [27] proposes an approach to create adversarial patches that can fool person detection systems. Patches are optimized to reduce the confidence of person detections, with total variation loss employed to generate smoother patches. To ensure physical realizability, non-printability score loss is utilized. Furthermore, transformations such as rotation and scaling are applied to the patch to enhance its robustness. We select the most effective loss function from the original paper, namely the object confidence score, as the optimization objective for the patch, resulting in the attack method: *AdvPatch*.

GAN-based naturalistic adversarial patch (GNAP) [5] proposes a method for generating naturalistic adversarial patches (NAPs) by leveraging pre-trained Generative Adversarial Networks (GANs) [12]. It searches for an input latent vector corresponding to a generated patch, which is visually natural while effectively deceiving the detectors. We adopt the dog pattern showcased in the original paper as the attack patch, resulting in the attack method: *GNAP*.

DM-based naturalistic adversarial patch (DM-NAP) [14] proposes a novel method for generating NAPs using the diffusion model [22]. Adversarial optimization is performed by backpropagating gradients from the victim object detector to iteratively update the diffusion model's latent representation, enhancing the patch's effectiveness for adversarial attacks while preserving its natural appearance. We adopt the Pomeranian image showcased in the original paper as the attack pattern and apply Stable Diffusion v1.4, resulting in the attack method: DM-NAP.

Adversarial Cloaks (AdvCloak) [29] optimizes adversarial patches to suppress person detection by applying thin-plate-spline (TPS) transformations, along with other augmentations like scaling and rotation, to enhance robustness against real-world distortions. Using the YOLOv2 [20] and YOLOv3 [19] as examples, we reproduce the *AdvCloak*.

Adversarial T-shirts (AdvTshirt) [31] presents an optimization-based approach for generating adversarial T-shirts. To fool a single detector, the method uses combining transformations, including perspective and TPS transformations. Using the YOLOv2 [20] as an example, we reproduce the *AdvTshirt*.

Appearing Attack (AA) [32] introduces feature-interference reinforcement and enhanced realistic constraints to improve adversarial example generation. Additionally, it utilizes nested adversarial examples to enhance robustness against real-world factors such as varying distances, angles, backgrounds, and lighting conditions. We set the target of the patch to "person" and reproduce *AA* on the YOLOv2 [20].

Adversarial Sticker (AdvSticker) [2] is an image-independent adversarial patch that, when placed anywhere in a scene, forces a classifier to predict a target class, making it effective for physical-world attacks without environmental constraints. We set the target of the patch to "person" and reproduce *AdvSticker* on the YOLOv3 [19]. **Universal Physical Camouflage (UPC)** [8] is a physical adversarial attack that generates category-agnostic camouflage patterns to deceive object detectors. Unlike instance-specific methods, it disrupts RPN, classification, and regression to induce mislocalization and misclassification. We use the dog-shaped constraint and reproduce *UPC* on the YOLOv3 [19].

A.2. Descriptions of Defenses

Segment and Complete defense (SAC) [17] trains a U-Net [23] as the patch segmenter for defending object detectors against patch attacks through detection and removal of adversarial patches, with a self adversarial training algorithm to improve robustness. In this evaluation, we reproduce the source code using the model weights and parameters provided in the original paper.

Patch-Agnostic Defense (PAD) [10] proposes a novel adversarial patch localization and removal method that does not require prior knowledge or additional training, based on

	YOLOv2	YOLOv3	YOLOv4	YOLOv5	YOLOv7	SSD	CenterNet	RetinaNet	MRCNN	FRCNN	DDETR
T-SEA [7]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
T-SEA-mim [7]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
T-SEA-pgd [7]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TC-EGA [6]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TCA [6]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Advpatch [27]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
GNAP [5]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DM-NAP [14]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
AdvCloak [29]	✓	✓									
AdvTshirt [31]	✓										
AA [32]	✓										
AdvSticker [2]		✓									
UPC [8]		✓									

Table 1. The 94 types of patches in our Adversarial Patch Defense Evaluation (APDE) dataset spanning 13 attack methods and 11 detectors.

semantic independence and spatial heterogeneity. In this evaluation, we select Segment Anything Model (SAM) [13] ViT-L version as the segmenter using the parameters provided in the original paper.

Adversarial YOLO (Adyolo) [9] proposes an efficient and effective plug-in defense component for the YOLO detection system. The core idea is to introduce a patch class into the YOLO architecture, resulting in only a minimal increase in inference time. In this evaluation, to ensure fairness in comparison and maintain a white-box attack scenario, we convert the predictions of the patch class into a mask and apply the same patch filling methods used by other defenses.

NAPGuard [28] provides robust detection capabilities against naturalistic adversarial patches (NAPs) through a meticulously designed critical feature modulation framework: the aggressive feature aligned learning and the natural feature suppressed inference. In this evaluation, we reproduce and train the NAPGuard model using the parameters from the original paper on the GAP dataset [28].

Diffusion-Based Adversarial Defense (DIFFender) [11] proposes a novel defense method that leverages a text-guided diffusion model to defend against adversarial patches. DIFFender uses two well-designed prompts to achieve patch localization and restoration. As the original paper's code is incomplete, we reproduce DIFFender model as faithfully as possible based on the paper, along with the key parts and parameters provided in the available code.

NutNet [16] proposes an innovative model called NutNet for detecting adversarial patches, with high generalization, robustness, and efficiency. Image-splitting and Destructive Training are introduced to enhance the model's ability to reconstruct only the images from a specific clean distribution, which allows for precise identification and masking of patches. In this evaluation, we reproduce NutNet using the source code provided in the original paper and select the AutoEncoder32 [16] model for testing.

Local Gradients Smoothing (LGS) [18] estimates noise locations in the gradient domain and transforms high-activation regions caused by adversarial noise in the image domain, while minimizing the impact on salient objects critical for accurate classification. Since most current patches are optimized using the total variation loss function, pixel-level smoothing at the image domain is insufficient to effectively eliminate patches. For the sake of fairness in evaluation, we adopt the same patch filling methods used by other defenses.

Zmask [24] utilizes Z-score analysis on internal network features to detect and mask pixels corresponding to adversarial objects in the input image, in order to enhance the adversarial robustness of DNNs against physically realizable adversarial attacks. In this evaluation, we reproduce Zmask using the source code provided in the original paper, with model initialization on the INRIA-Person [4] and MS COCO [15] test datasets.

Jedi [26] detects regions with high entropy and uses an autoencoder that can complete patch regions from high entropy kernels to resist patches. In this evaluation, we reproduce the source code using the model weights and parameters provided in the original paper.

ObjectSeeker [30] is a certifiably robust defense against patch attacks in object detection. It uses patch-agnostic masking to remove adversarial patches without prior knowledge of their shape, size, or location, allowing safe detection with any standard detector.

A.3. Additional Experimental Settings

Patch Acquisition. To ensure diverse patch distributions, we construct the large-scale Adversarial Patch Defense Evaluation (APDE) dataset, which includes 2 attack goals, 13 patch attack types, and 11 object detectors. As shown in Tab. 1, a total of 94 adversarial patches are trained. When the APDE dataset is used to retrain defense methods, AdvCloak [29], AdvTshirt [31], and AA [32] are excluded from the training set, allowing for an evaluation of the defense methods' robustness against out-of-domain patches. AdvCloak [29], AdvSticker [2], and UPC [8] are trained to eval-

Model (w/o defense)	**SAC [17]**	**PAD [10]**	**Adyolo [9]**	**NAPGuard[28]**	**DIFFender[11]**	**NutNet [16]**	**LGS [18]**	**Zmask [24]**	**Jedi [26]**
YOLOv2 (82.19)	**82.19**	81.45	**82.19**	**82.19**	76.54	79.08	78.51	79.72	77.36
YOLOv3 (96.87)	**96.87**	95.21	**96.87**	**96.87**	90.05	93.90	91.98	95.35	92.66
YOLOv4 (94.36)	**94.36**	93.48	**94.36**	**94.36**	89.56	92.15	88.68	93.29	89.03
YOLOv5 (94.90)	**94.90**	94.72	**94.90**	**94.90**	92.55	93.17	88.89	93.47	90.30
YOLOv7 (95.58)	**95.58**	90.58	**95.58**	**95.58**	84.26	87.20	88.97	93.40	92.32
SSD (81.28)	**81.28**	80.03	**81.28**	**81.28**	78.89	79.35	73.90	79.70	75.91
CenterNet (92.10)	**92.10**	91.77	**92.10**	**92.10**	85.26	87.44	82.16	89.70	85.04
RetinaNet (96.03)	**96.03**	95.24	**96.03**	**96.03**	89.18	93.35	90.78	94.35	91.49
MRCNN (97.20)	**97.20**	96.42	**97.20**	**97.20**	94.68	96.08	92.60	95.84	94.16
FRCNN (96.92)	**96.92**	96.81	**96.92**	**96.92**	94.79	95.56	92.84	95.55	93.51
DDETR (92.23)	**92.23**	89.82	92.20	92.10	76.31	86.25	83.77	90.55	84.71
Overall (92.70)	**92.70**	91.41	92.69	92.68	86.55	89.41	86.64	90.99	87.86

Table 2. **Comparison of the impact of each defense on clean images, i.e., without patches.** We report person AP@0.5.

uate the robustness of defense methods against adversarial patches of diverse shapes.

Data Acquisition. For this evaluation, we randomly select 1000 positive samples from the testing sets of the widely used INRIA-Person [4] (288 images) and MS COCO [15] (2693 images) datasets for object detection, serving as the source data. These positive samples are utilized to generate hiding attack patches, while negative samples from the testing sets are employed as backgrounds for appearing attack patches, resulting in 94,000 images in total. The dataset is divided into a training set (56,400 images) and a testing set (37,600 images), following a 6:4 ratio.

Data Properties. All images are stored in PNG format with a fixed size of 416 × 416 pixels, achieved through padding or resizing, aligning with the settings described in the respective papers. For all hiding attack patches, we apply the patches to every person labeled in the positive samples, attaching the patch to each person. For the appearing attack, we apply the patches to random locations in the negative samples, with patch sizes ranging from 30 to 80 pixels in length. Instead of heavy human annotations, all adversarial patches are automatically labeled with detection boxes and masks during dataset generation, making them readily available for training or evaluating defense methods.

Adaptive Attacks Properties. Adaptive attacks refer to scenarios where the adversary possesses full knowledge of the defense model's parameters and design details, enabling them to tailor specific attacks that exploit the model's vulnerabilities. The susceptibility of different defense models to adaptive attacks varies significantly. For instance, some defenses are inherently more prone to adaptive attacks due to their architectural characteristics. In this study, we analyze 10 defense methods, among which 5 explicitly discuss adaptive attacks (SAC [17], Zmask [24], Jedi [26], DIFFender [11], and NutNet [16]) against themselves in their original papers. For these 5 methods, we directly replicate the original adaptive attacks as part of our evaluation. For the remaining defense methods, we designed adaptive attacks to maximally compromise their defensive performance, ensuring the most potent adversarial impact under the defined threat model. For defenses based on patch detection or segmentation: Adyolo [9] and NAPGuard [28] have gradient-traceable defense models. Thus, we incorporate their defense model outputs into the loss function to train adaptive attack patches. PAD [10], however, relies on the SAM model with non-traceable gradients. To circumvent this, we focus on its dependency on semantic differences and spatial differences, constraining the adversarial patch to maintain semantic and spatial consistency with its surrounding context. For defenses based on prior knowledge of patches: LGS [18] localizes patches by detecting pixel-level discontinuities. We integrate patch smoothness into the loss function to train adaptive attacks that evade its detection mechanism.

B. Additional Experimental Results

shou dao Here, we provide more detailed experimental results. First, in Sec B.1, we present how defense methods impact clean sample detection. We then assess robustness against diverse patch shapes in Sec B.2, and further analyze cross-dataset generalization in Sec B.3. Additionally, Sec B.4 compares patch erasure techniques, while Sec. B.5 focuses on defenses against appearing attacks. Furthermore, Sec B.6 provides an additional analysis of patches bypassing defenses. Finally, Sec B.7 provides comprehensive results for individual attacks.

B.1. Defenses Impact on Clean Samples

The primary goal of defense methods is to mitigate the impact of adversarial patches on normal detection while minimizing their own effect on the detection of clean images. As shown in Tab. 2, we measure the impact of each defense method on clean images without attack patches. We find that SAC [17], Adyolo [9], and NAPGuard [28] have almost no impact on the detection of clean samples, whereas other defense methods affect the detector's performance to varying degrees. This phenomenon is directly related to the robustness of the defense methods. In the main text, we re-

Defense	T-SEA	AdvCloak	AdvSticker	UPC	LAPs
SAC [17]	48.89	4.17	40.42	52.36	54.2
PAD [10]	79.85	59.04	78.58	**84.19**	**85.28**
Adyolo [9]	67.21	18.29	61.35	69.38	64.54
NAPGuard[28]	78.54	52.21	75.49	74.09	77.36
DIFFender[11]	57.35	27.4	62.74	67.28	61.02
NutNet [16]	**81.94**	**64.14**	**83.12**	82.83	83.42
LGS [18]	67.3	30.56	56.27	71.56	65.69
Zmask [24]	73.35	42.07	70.52	70.06	64.23
Jedi [26]	62.98	35.24	69.24	67.24	64.37

Table 3. **Comparison of different defense methods against adversarial patches of diverse shapes**(T-SEA [7]: square-shaped, AdvCloak [29]: rectangular, AdvSticker [2]: circle-shaped, UPC [8]: dog-shaped, and LAPs [25]: cartoon-patterned). We report AP@0.5 after defenses, where a lower AP indicates better defense performance.

veal that defenses like NutNet [16] and LGS [18] tend to misidentify the background as a patch, exhibiting relatively poor robustness, which in turn affects the detection of clean samples to some extent.

B.2. Defense against Diverse Patch Shapes

Existing evaluations of defense methods predominantly focus on rectangular adversarial patches, yet adversarial patches can exhibit diverse shapes. For example, circular patches are utilized in AdvSticker [2], while naturalistic shapes (e.g., dog-shaped patches in UPC [8] or cartoon-patterned patches in LAPs [25]) further expand the threat landscape. Evaluating defenses against such irregularly shaped patches provides critical insight into their generalization capability. As illustrated in the figure, we benchmark defense performance under four distinct patch shapes. The results indicate that NutNet [16] and PAD [10] demonstrate consistent robustness across all patch shapes. In contrast, NAPGuard [28] shows degraded performance against irregularly shaped patches compared to squares and rectangles. This limitation arises because NAPGuard [28] employs an object detection model that generates bounding boxes restricted to rectangular shapes for patch localization. Furthermore, the defense model was trained exclusively on rectangular patches, significantly limiting its generalization to irregular shapes during evaluation.

B.3. Cross-Dataset Training Comparison

To compare the efficacy of our APDE dataset with prior adversarial patch datasets in training defense models, we select three representative defense methods as test models. For a fair evaluation, the adversarial patches used in testing (i.e., AdvCloak [29] and AdvTshirt [31]) are excluded from the training sets. As shown in Tab. 4, our APDE dataset significantly improves the performance of all three defense methods, outperforming other datasets. Notably, models trained on the Apricot [1] dataset exhibit performance degradation in certain defenses, highlighting its limitations in generalization.

		SAC [17]		**Adyolo** [9]		**NAPGuard** [28]	
		original	retrained	original	retrained	original	retrained
AdvCloak[29]	**Apricot**[1]	4.17	/	18.29	14.43	52.21	43.85
	GAP[28]	4.17	63.19	18.29	21.04	52.21	/
	APDE	4.17	**71.29**	18.29	**22.36**	52.21	**73.16**
AdvTshirt[31]	**Apricot**[1]	34.27	/	8.19	17.31	50.21	41.27
	GAP[28]	34.27	45.06	8.19	29.83	50.21	/
	APDE	34.27	**64.47**	8.19	**37.53**	50.21	**70.89**

Table 4. **Comparison of defense performance before and after retraining on different datasets.** We report the AP@0.5 metric for each defense method before and after retraining.

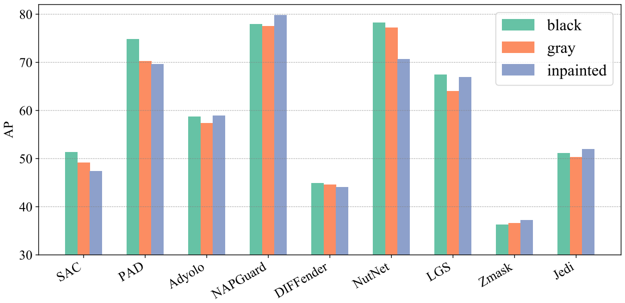


Figure 1. **The impact of three patch erasure methods on defenses**. We report AP@0.5 for black, gray and inpainted patch erasure.

B.4. Patch Erasure Techniques Analysis

Existing defense methods apply different strategies to handle detected patches. The most common approach is mask filling, i.e., setting the pixels of the patch region to black to eliminate the patch's impact on the victim detector. Referring to the pre-processing operations for input images in object detection tasks, padding with a constant pixel value is commonly used. Some methods like NutNet [16] set the patch region to gray, that is, a constant pixel value of [128, 128, 128]. Additionally, some defense methods inpaint the image for coherent restoration in the patch region. For example, DIFFender [11] uses stable diffusion [22] with pre-tuned restoration prompts to restore original pixels. Jedi [26] replaces mask pixels (starting from the boundary and moving inward) with a weighted sum of external pixel values within a certain radius. Since the goal of this step is to eliminate the patch rather than precisely restore the original pixels, it is unnecessary to use pixel-perfect but time-consuming restoration methods. In Fig. 1, we demonstrate how three different filling methods impact defense performance. The results show that black filling performs better than gray filling, while inpainting shows varying results across different defense methods. Overall, the impact of the erasure method on defense effectiveness is not substantial and is largely determined by the performance of the defense method itself.

	YOLOv2	YOLOv3	YOLOv4	SSD	MRCNN	FRCNN
w/o defense	35.43	23.36	12.93	2.99	1.78	10.59
SAC [17]	18.23	24.07	10.96	7.01	1.79	6.2
PAD [10]	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Adyolo [9]	35.13	25.84	14.48	4.65	2.76	13.25
NAPGuard[28]	1.04	0.9	0.23	0.7	0.05	0.03
DIFFender[11]	6.72	11.29	0.81	12.58	0.33	0.86
NutNet [16]	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
LGS [18]	3.13	2.61	0.09	1.48	0.19	0.79
Zmask [24]	6.5	9.66	5.23	12.01	2.16	2.42
Jedi [26]	19.26	11.31	2.93	3.23	1.41	6.03

Table 5. **Results for defense methods against appearing attacks.** We report AP@0.5 after defenses, where a lower AP indicates better defense performance.

B.5. Defenses against Appearing Attack Patches

In our previous experiments, the patches discussed are primarily hiding attack patches, applied to the target object to make it invisible to the victim detectors. However, attackers can also launch appearing attacks (AA), where the patch is misclassified as a specific object. Most defense works focus only on defending against hiding attack patches and have not explored defenses against appearing attack patches. In this study, we replicate the AA patch [32] generated based on YOLOv2 [20]. We apply the patches to negative samples from the INRIA test set. In contrast to hiding attacks, the larger the person AP for appearing attacks, the better the attack and the worse the defense performance. We select several detectors to test the performance of different defense methods, with results shown in Tab. 5. Only the effect after Adyolo [9] defense is worse than before the defense. This is mainly because Adyolo only added the adversarial patch class on the YOLOv2 [20] model while retaining the patch detection model's ability to recognize pedestrians. As a result, in the appearing attack experiment, Adyolo misclassifies the patch as a "person" completely nullifying the defense effect. The main results show that some defense methods that perform well against hiding attacks also offer good defense performance against appearing attack patches. Therefore, **adversarial patch attack goals are not the primary factor influencing the defense performance**.

B.6. More Detailed Analysis on Defense Failures

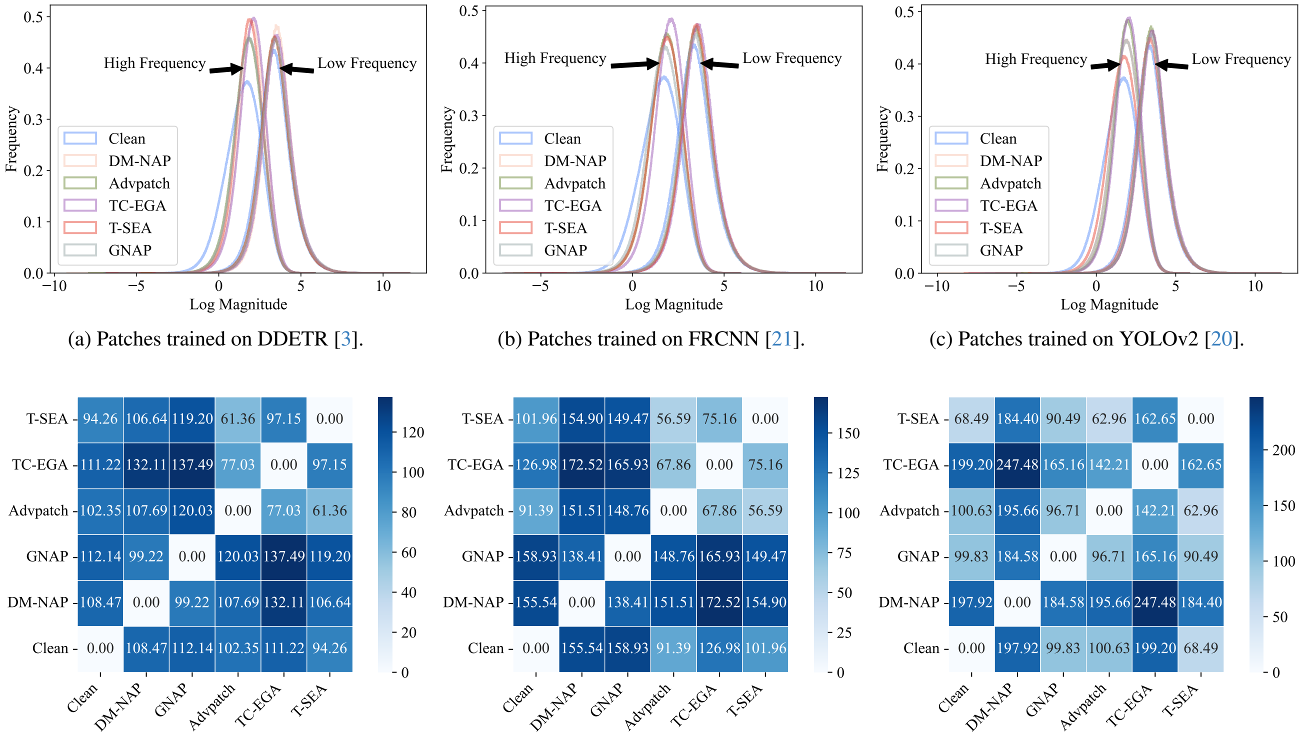
Previous study [28] suggests that, unlike non-NAPs, the high-frequency components of NAPs closely resemble their surroundings, making them more deceptive and challenging to detect accurately. As illustrated in Fig. 2, we present additional frequency-domain component distributions of adversarial patches to further analyze this issue and observe little differences in the high-frequency components between NAPs and non-NAPs. Here, we compute the Fréchet Inception Distance (FID) scores between each of these 5 attacks and clean samples. Our findings reveal that patches trained on DDETR [3] and FRCNN [21] still show that non-NAPs exhibit closer FID scores to each other, suggesting more similar data distributions, whereas NAPs are more distinctly distributed.

Although most cases align with the observations described above, we identified an exception: the non-NAPs generated by TC-EGA [6] on YOLOv2 [20] appear more distinctly distributed compared to other attack patches. As shown in the attack results in Fig. 2, the patches generated by TC-EGA on YOLOv2 effectively bypass defenses like NAPGuard [28] and Adyolo [9]. Therefore, as discussed in the main text, pre-trained defenses heavily rely on the diversity of the training data distribution, highlighting the importance of introducing the APDE dataset with diverse patch distributions.

B.7. Defense Performance on Individual Attacks

As shown in Fig. 3, the PR curves in these subfigures compare the defense performance of different methods. When evaluating individual attacks, the performance of different defenses varies against the specific attack. For example, NAPGuard [28] demonstrates strong defense against T-SEA [7] and AdvPatch [27], achieving 72.87% and 68.44% person AP@0.5, respectively, on the YOLOv2 [20] model. However, its performance against TC-EGA [6] and DM-NAP [14] is relatively weak, with only 47.48% and 55.92% person AP@0.5. Therefore, constructing a large-scale adversarial example dataset in a white-box scenario is crucial for comprehensively and effectively measuring the robustness of defense methods and exploring the most challenging situations they face.

Additionally, Fig. 4 to 11 illustrate the defense effectiveness and patch detection results of all evaluated methods against various attack types. The first row represents images with adversarial patches, while the second row shows patch mask images generated by defenses (Green: the true location of patches. Red: detected patches by the defense. Blue: the background mistakenly identified as patches). The first column without defense results in incorrect predictions (red), while the remaining columns achieve correct results (blue).



(a) Patches trained on DDETR [3].

(b) Patches trained on FRCNN [21].

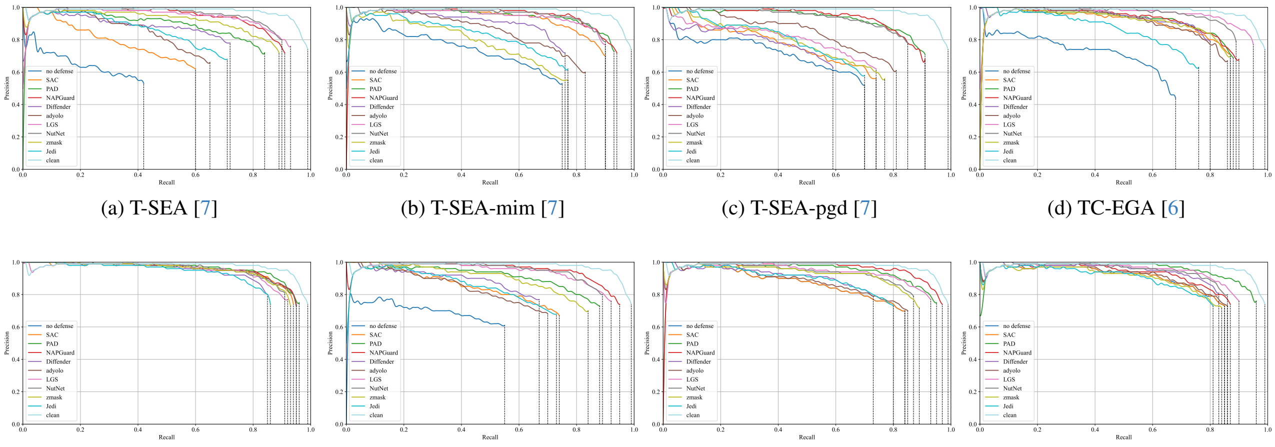
(c) Patches trained on YOLOv2 [20].

(d) Patches trained on DDETR [3].

(e) Patches trained on FRCNN [21].

(f) Patches trained on YOLOv2 [20].

Figure 2. (a)-(c) Frequency domain distribution of patches and (d)-(f) FID scores between patches generated by different attacks.



(a) T-SEA [7]

(b) T-SEA-mim [7]

(c) T-SEA-pgd [7]

(d) TC-EGA [6]

(e) TCA [6]

(f) Advpatch [27]

(g) GNAP [5]

(h) DM-NAP [14]

Figure 3. **Precision-Recall curves for 9 different defenses against 8 different attack methods.**

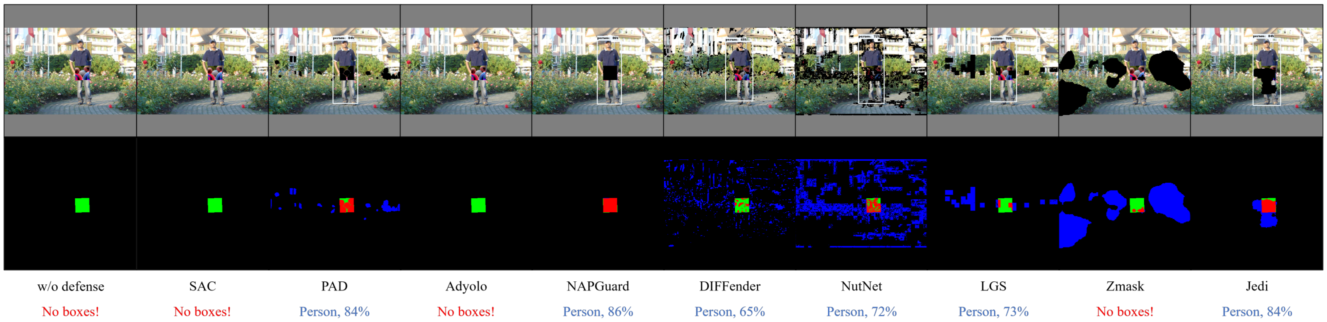


Figure 4. **Comparison of patch detection and defense performance against T-SEA [7] attack patches.**

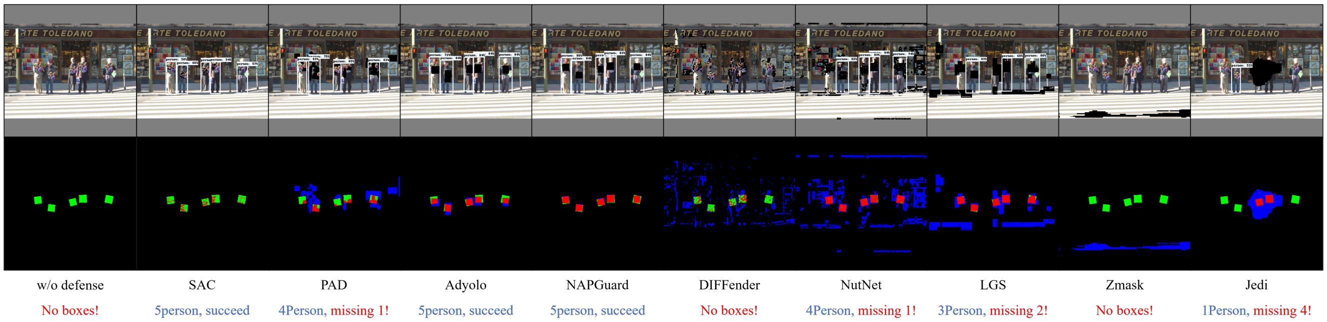


Figure 5. **Comparison of patch detection and defense performance against T-SEA-mim [7] attack patches.**

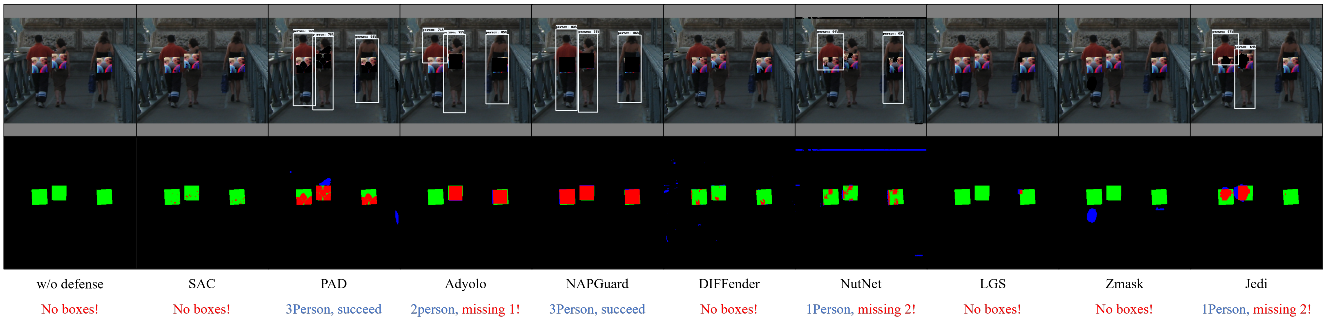


Figure 6. **Comparison of patch detection and defense performance against T-SEA-pgd [7] attack patches.**

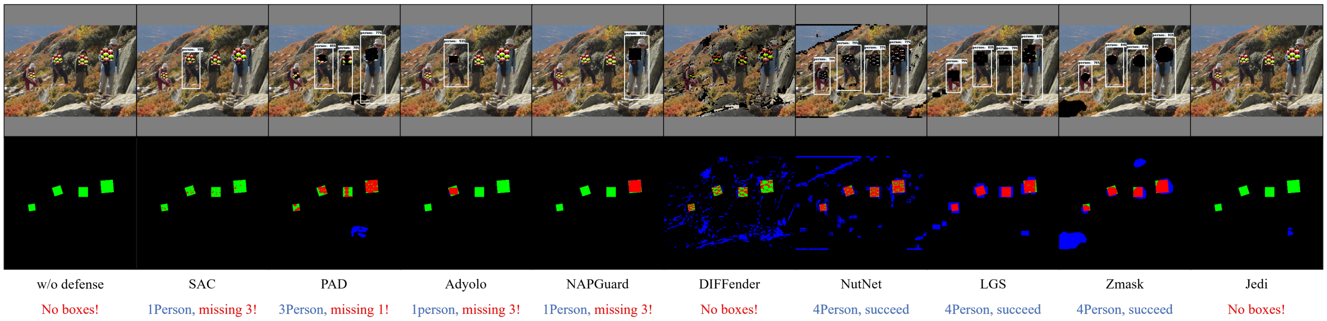


Figure 7. **Comparison of patch detection and defense performance against TC-EGA [6] attack patches.**

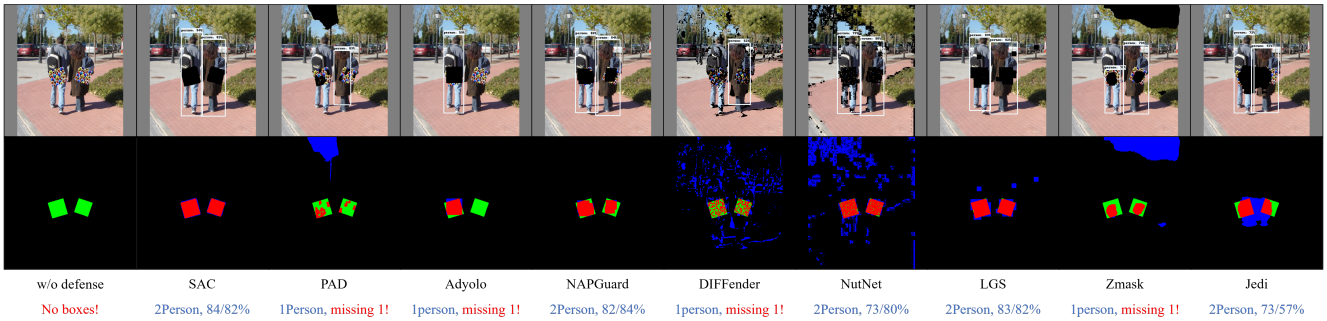


Figure 8. **Comparison of patch detection and defense performance against TCA [6] attack patches.**

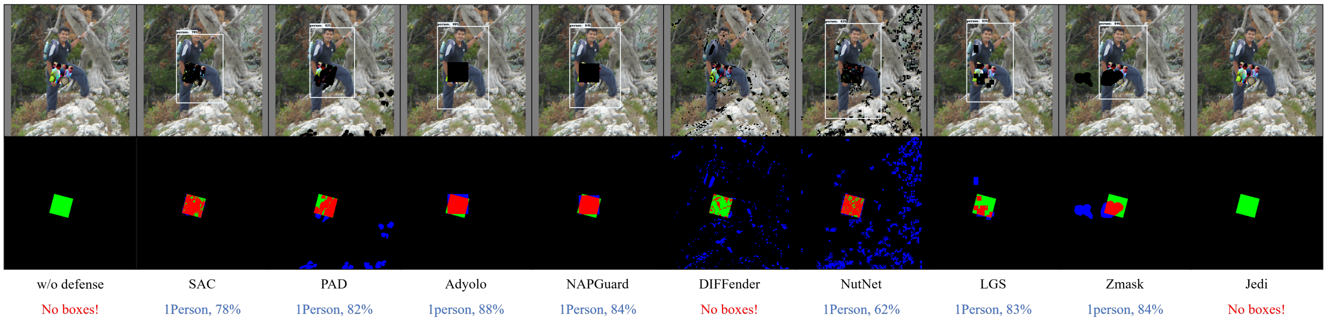


Figure 9. **Comparison of patch detection and defense performance against Advpatch [27] attack patches.**

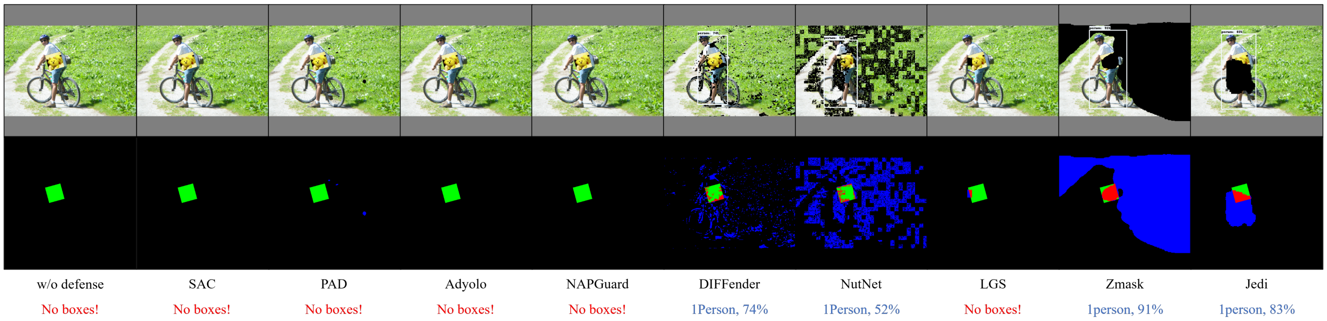


Figure 10. **Comparison of patch detection and defense performance against GNAP [5] attack patches.**

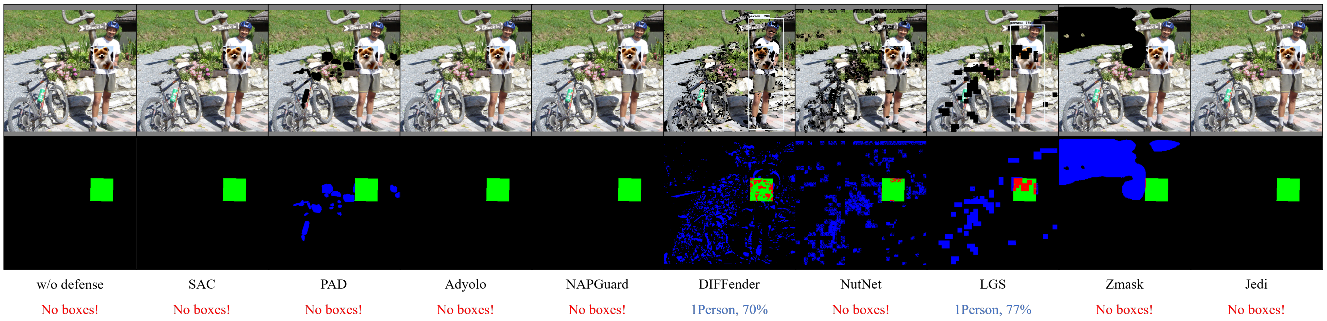


Figure 11. **Comparison of patch detection and defense performance against DM-NAP [14] attack patches.**

References

[1] Anneliese Braunegg, Amartya Chakraborty, and Krumdick. Apricot: A dataset of physical adversarial attacks on object detection. In *European Conference on Computer Vision*, pages 35–50. Springer, 2020. 4

[2] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 1, 2, 4

[3] 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. 5, 6

[4] 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)*, pages 886–893. Ieee, 2005. 2, 3

[5] Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, and Chen. Naturalistic physical adversarial patch for object detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 6, 8

[6] 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, 5, 6, 7, 8

[7] Hao Huang, Ziyan Chen, Huanran Chen, Yongtao Wang, and Kevin Zhang. T-sea: Transfer-based self-ensemble attack on object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20514–20523, 2023. 1, 2, 4, 5, 6, 7

[8] 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, 4

[9] Nan Ji, YanFei Feng, Haidong Xie, Xueshuang Xiang, and Naijin Liu. Adversarial yolo: Defense human detection patch attacks via detecting adversarial patches. *arXiv preprint arXiv:2103.08860*, 2021. 2, 3, 4, 5

[10] Lihua Jing, Rui Wang, Wenqi Ren, Xin Dong, and Cong Zou. Pad: Patch-agnostic defense against adversarial patch attacks. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24472–24481, 2024. 1, 3, 4, 5

[11] Caixin Kang, Yinpeng Dong, Zhengyi Wang, Shouwei Ruan, Hang Su, and Xingxing Wei. Diffender: Diffusion-based adversarial defense against patch attacks in the physical world. *arXiv preprint arXiv:2306.09124*, 2023. 2, 3, 4, 5

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[14] Shuo-Yen Lin, Ernie Chu, Che-Hsien Lin, Jun-Cheng Chen, and Jia-Ching Wang. Diffusion to confusion: Naturalistic adversarial patch generation based on diffusion model for object detector. *arXiv preprint arXiv:2307.08076*, 2023. 1, 2, 5, 6, 8

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, and Perona. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 2, 3

[16] Zijin Lin, Yue Zhao, Kai Chen, and Jinwen He. I don't know you, but i can catch you: Real-time defense against diverse adversarial patches for object detectors. *arXiv preprint arXiv:2406.10285*, 2024. 2, 3, 4, 5

[17] Jiang Liu, Alexander Levine, Chun Pong Lau, Rama Chellappa, and Soheil Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14973–14982, 2022. 1, 3, 4, 5

[18] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1300–1307, 2019. 2, 3, 4, 5

[19] Joseph Redmon. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[20] 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. 1, 5, 6

[21] Shaoqing Ren. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 5, 6

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 4

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 1

[24] Giulio Rossolini, Federico Nesti, Fabio Brau, Alessandro Biondi, and Giorgio Buttazzo. Defending from physically-realizable adversarial attacks through internal over-activation analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12):15064–15072, 2023. 2, 3, 4, 5

[25] Jia Tan, Nan Ji, Haidong Xie, and Xueshuang Xiang. Legitimate adversarial patches: Evading human eyes and detection models in the physical world. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 5307–5315, New York, NY, USA, 2021. Association for Computing Machinery. 4

[26] Bilel Tarchoun, Anouar Ben Khalifa, Mohamed Ali Mahjoub, Nael Abu-Ghazaleh, and Ihsen Alouani. Jedi: Entropy-based localization and removal of adversarial patches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4087–4095, 2023. 2, 3, 4, 5

[27] 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, 5, 6, 8

[28] Siyang Wu, Jiakai Wang, Jiejie Zhao, Yazhe Wang, and Xianglong Liu. Napguard: Towards detecting naturalistic adversarial patches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24367–24376, 2024. 2, 3, 4, 5

[29] 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, 4

[30] Chong Xiang, Alexander Valtchanov, Saeed Mahloujifar, and Prateek Mittal. Objectseeker: Certifiably robust object detection against patch hiding attacks via patch-agnostic masking. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1329–1347. IEEE, 2023. 2

[31] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, and Sun. Adversarial t-shirt! evading person detectors in a physical world. In *European Conference on Computer Vision*, pages 665–681. Springer, 2020. 1, 2, 4

[32] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 1989–2004, 2019. 1, 2, 5