Segment and Complete: Defending Object Detectors against Adversarial Patch Attacks with Robust Patch Detection

Jiang Liu [1], Alexander Levine[2], Chun Pong Lau[1], Rama Chellappa[1], Soheil Feizi[2]
[1]Johns Hopkins University, [2]University of Maryland, College Park
{jiangliu, clau13, rchella4}@jhu.edu, {alevine0,sfeizi}@cs.umd.edu

Abstract

Object detection plays a key role in many security-critical systems. Adversarial patch attacks, which are easy to implement in the physical world, pose a serious threat to state-of-the-art object detectors. Developing reliable defenses for object detectors against patch attacks is critical but severely understudied. In this paper, we propose Segment and Complete defense (SAC), a general framework for defending object detectors against patch attacks through detection and removal of adversarial patches. We first train a patch segmenter that outputs patch masks which provide pixel-level localization of adversarial patches. We then propose a self adversarial training algorithm to robustify the patch segmenter. In addition, we design a robust shape completion algorithm, which is guaranteed to remove the entire patch from the images if the outputs of the patch segmenter are within a certain Hamming distance of the ground-truth patch masks. Our experiments on COCO and xView datasets demonstrate that SAC achieves superior robustness even under strong adaptive attacks with no reduction in performance on clean images, and generalizes well to unseen patch shapes, attack budgets, and unseen attack methods. Furthermore, we present the APRICOT-Mask dataset, which augments the APRICOT dataset with pixel-level annotations of adversarial patches. We show SAC can significantly reduce the targeted attack success rate of physical patch attacks. Our code is available at https://github.com/joellliu/SegmentAndComplete.

1. Introduction

Object detection is an important computer vision task that plays a key role in many security-critical systems including autonomous driving, security surveillance, identity verification, and robot manufacturing [42]. Adversarial patch attacks, where the attacker distorts pixels within a region of bounded size, pose a serious threat to real-world object detection systems since they are easy to implement physically. For example, physical adversarial patches can make a stop sign [40] or a person [41] disappear from object detectors, which could cause serious consequences in security-critical settings such as autonomous driving. Despite the abundance [9, 22, 23, 25, 29, 39–41, 43, 45, 51] of adversarial patch attacks on object detectors, defenses against such attacks have not been extensively studied. Most existing defenses for patch attacks are restricted to image classification [16, 17, 24, 32, 34, 44, 46, 49]. Securing object detectors is more challenging due to the complexity of the task.

Figure 1. We adopt a "detect and remove" strategy for defending object detectors against patch attacks. Left: Predictions on a clean image; middle: predictions on an adversarial image; right: predictions on SAC masked image.

In this paper, we present Segment and Complete (SAC) defense that can robustify any object detector against patch attacks without re-training the object detectors. We adopt a "detect and remove" strategy (Fig. 1): we detect adversarial patches and remove the area from input images, and then feed the masked images into the object detector. This is based on the following observation: while adversarial patches are localized, they can affect predictions not only locally but also on objects that are farther away in the image because object detection algorithms utilize spatial context for reasoning [38]. This effect is especially significant for deep learning models, as a small localized adversarial patch

can significantly disturb feature maps on a large scale due to large receptive fields of neurons. By removing them from the images, we minimize the adverse effects of adversarial patches both locally and globally.

The key of SAC is to robustly detect adversarial patches. We first train a patch segmenter to segment adversarial patches from the inputs and produce an initial patch mask. We propose a self adversarial training algorithm to enhance the robustness of the patch segmenter, which is efficient and object-detector agnostic. Since the attackers can potentially attack the segmenter and disturb its outputs under adaptive attacks, we further propose a robust shape completion algorithm that exploits the patch shape prior to ensure robust detection of adversarial patches. Shape completion takes the initial patch mask and generates a "completed patch mask" that is *guaranteed* to cover the entire adversarial patch, given that the initial patch mask is within a certain Hamming distance from the ground-truth patch mask. The overall pipeline of SAC is shown in Fig. 2. SAC achieves 45.0% mAP under 100×100 patch attacks, providing 30.6% mAP gain upon the undefended model while maintaining the same 49.0% clean mAP on the COCO dataset.

Besides *digital* domains, patch attacks have become a serious threat for object detectors in the *physical* world [9, 23, 39–41, 43, 45]. Developing and evaluating defenses against physical patch attacks require physical-patch datasets which are costly to create. To the best of our knowledge, APRICOT [6] is the only publicly available dataset of physical adversarial attacks on object detectors. However, APRICOT only provides bounding box annotations for each patch without pixel-level annotations. This hinders the development and evaluation of patch detection and removal techniques like SAC. To facilitate research in this direction, we create the *APRICOT-Mask* dataset, which provides segmentation masks and more accurate bounding boxes for adversarial patches in APRICOT. We train our patch segmenter with segmentation masks from APRICOT-Mask and show that SAC can effectively reduce the patch attack success rate from 7.97% to 2.17%.

In summary, our contributions are as follows:

- We propose Segment and Complete, a general method for defending object detectors against patch attacks via patch segmentation and a robust shape completion algorithm.
- We evaluate SAC on both digital and physical attacks. SAC achieves superior robustness under both non-adaptive and adaptive attacks with no reduction in performance on clean images, and generalizes well to unseen shapes, attack budgets, and unseen attack methods.
- We present the APRICOT-Mask dataset, which is the first publicly available dataset that provides pixel-level annotations of physical adversarial patches.

2. Related Work

2.1. Adversarial Patch Attacks

Adversarial patch attacks are localized attacks that allow the attacker to distort a bounded region. Adversarial patch attacks were first proposed for image classifiers [7, 14, 20]. Since then, numerous adversarial patch attacks have been proposed to fool state-of-the-art object detectors including both digital [22, 25, 29, 38, 51] and physical attacks [9, 23, 39–41, 43, 45]. Patch attacks for object detection are more complicated than image classification due to the complexity of the task. The attacker can use different objective functions to achieve different attack effects such as object hiding, misclassification, and spurious detection.

2.2. Defenses against Patch Attacks

Many defenses have been proposed for image classifiers against patch attacks, including both empirical [16, 17, 32–34, 44] and certified defenses [24, 46, 49]. Local gradient smoothing (LGS) [33] is based on the observation that patch attacks introduce concentrated high-frequency noises and therefore proposes to perform gradient smoothing on regions with high gradients magnitude. Digital watermarking (DW) [17] finds unnaturally dense regions in the saliency map of the classifier and covers these regions to avoid their influence on classification. LGS and DW both use a similar detect and remove strategy as SAC. However, they detect patch regions based on predefined criteria, whereas SAC uses a learnable patch segmenter which is more powerful and can be combined with adversarial training to provide stronger robustness. In addition, we make use of the patch shape prior through shape completion.

In the domain of object detection, most existing defenses focus on global perturbations with the l_p norm constraint [8, 10, 50] and only a few defenses [19, 38, 47] for patch attacks have been proposed. These methods are designed for a specific type of patch attack or object detector, while SAC provides a more general defense. Saha [38] proposed Grad-defense and OOC defense for defending blindness attacks where the detector is blind to a specific object category chosen by the adversary. Ji *et al*. [19] proposed Ad-YOLO to defend human detection patch attacks by adding a patch class on YOLOv2 [35] detector such that it detects both the objects of interest and adversarial patches. DetectorGuard (DG) [47] is a provable defense against localized patch hiding attacks. Unlike SAC, DG does not localize or remove adversarial patches. It is an alerting algorithm that uses an objectness explainer that detects unexplained objectness for issuing alerts when attacked, while SAC solves the problem of "detection under adversarial patch attacks" that aims to improve model performance under attacks and is not limited to hiding attacks, although the patch mask detected by SAC can also be used as a signal for issuing attack alerts.

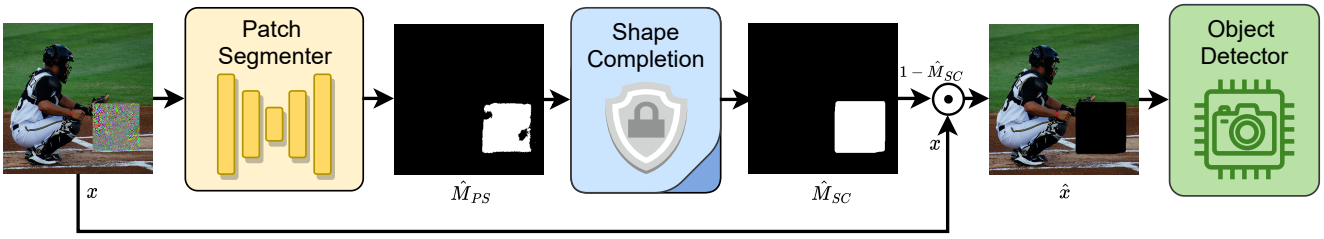


Figure 2. Pipeline of the SAC approach. SAC detects and removes adversarial patches on pixel-level through patch segmentation and shape completion, and feeds the masked images into the base object detector for prediction.

3. Preliminary

3.1. Faster R-CNN Object Detector

In this paper, we use Faster R-CNN [36] as our base object detector, though SAC is compatible with any object detector and we show the results for SSD [28] in the supplementary material. Faster R-CNN is a proposal-based two-stage object detector. In the first stage, a region proposal network (RPN) is used to generate class-agnostic candidate object bounding boxes called region proposals; in the second stage, a Fast R-CNN network [15] is used to output an object class and refine the bounding box coordinates for each region proposal. The total loss of Faster RCNN is the sum of bounding-box regression and classification losses of RPN and Fast R-CNN:

$$\mathcal{L}^{\text{Faster R-CNN}} = \mathcal{L}^{\text{RPN}}_{\text{reg}} + \mathcal{L}^{\text{RPN}}_{\text{cls}} + \mathcal{L}^{\text{Fast R-CNN}}_{\text{reg}} + \mathcal{L}^{\text{Fast R-CNN}}_{\text{cls}} \quad (1)$$

3.2. Attack Formulation

In this paper, we consider image- and location-specific untargeted patch attack for object detectors, which is strictly stronger than universal, location invariant attacks. Let $x \in [0,1]^{H\times W\times 3}$ be a clean image, where H and W are the height and width of x. We solve the following optimization problem to find an adversarial patch:

$$\hat{P}(x,l) = \underset{P\in\{P':\|P'\|_\infty\leq\epsilon\}}{\arg\max} \mathcal{L}(h(A(x,l,P)); y), \quad (2)$$

where h denotes an object detector, $A(x,l,P)$ is a "patch applying function" that adds patch P to x at location l, $\|\cdot\|_\infty$ is l_∞ norm, ϵ is the attack budget, y is the ground-truth class and bounding box labels for objects in x, and $\mathcal{L}$ is the loss function of the object detector. We use $\mathcal{L} = \mathcal{L}^{\text{Faster R-CNN}}$ for a general attack against Faster R-CNN. We solve Eq. (2) using the projected gradient descent (PGD) algorithm [30]:

$$P^{t+1} = \prod_{\mathbb{P}}(P^t + \alpha\,\text{sign}(\nabla_{P^t}\mathcal{L}(h(A(x,l,P^t)); y))), \quad (3)$$

where α is the step size, t is the iteration number, and $\prod$ is the projection function that projects P to the feasible set $\mathbb{P} = \{P : \|P\|_\infty \leq \epsilon \text{ and } A(x,l,P) \in [0,1]^{H\times W\times 3}\}$. The adversarial image x_{adv} is given by: $x_{\text{adv}} = A(x,l,\hat{P}(x,l))$.

We consider square patches $P \in \mathbb{R}^{s\times s\times 3}$, where s is the patch size, and apply one patch per image following previous works [20, 23, 29, 49]. We use an attack budget $\epsilon = 1$ that allows the attacker arbitrarily distort pixels within a patch without a constraint, which is the case for physical patch attacks and most digital patch attacks [7, 22, 25, 29, 38, 51].

4. Method

SAC defends object detectors against adversarial patch attacks through detection and removal of adversarial patches in the input image x. The pipeline of SAC is shown in Fig. 2. It consists of two steps: first, a patch segmenter (Sec. 4.1) generates initial patch masks $\hat{M}_{PS}$ and then a robust shape completion algorithm (Sec. 4.2) is used to produce the final patch masks $\hat{M}_{SC}$. The masked image $\hat{x} = x \odot (1 - \hat{M}_{SC})$ is fed into the base object detector for prediction, where $\odot$ is the Hadamard product.

4.1. Patch Segmentation

Training with pre-generated adversarial images We formulate patch detection as a segmentation problem and train a U-Net [37] as the patch segmenter to provide initial patch masks. Let PS_θ be a patch segmenter parameterized by θ. We first generate a set of adversarial images $\mathcal{X}_{\text{adv}}$ by attacking the base object detector with Eq. (2), and then use the pre-generated adversarial images $\mathcal{X}_{\text{adv}}$ to train PS_θ:

$$\min_\theta \sum_{x_{\text{adv}}\in\mathcal{X}_{\text{adv}}} \mathcal{L}_{\text{BCE}}(PS_\theta(x_{\text{adv}}), M) \quad (4)$$

where M is the ground-truth patch mask, $PS_\theta(x_{\text{adv}}) \in [0,1]^{H\times W}$ is the output probability map, and $\mathcal{L}_{\text{BCE}}$ is the binary cross entropy loss:

$$\mathcal{L}_{\text{BCE}}(\hat{M}, M) = -\sum_i^H \sum_j^W [M_{ij}\cdot\log\hat{M}_{ij} + (1-M_{ij})\cdot\log(1-\hat{M}_{ij})]. \quad (5)$$

Self adversarial training Training with $\mathcal{X}_{\text{adv}}$ provides prior knowledge for PS_θ about "how adversarial patches look like". We further propose a self adversarial training algorithm to robustify PS_θ. Specifically, we attack PS_θ to generate adversarial patch $\hat{P}_{\text{s-AT}} \in \mathbb{R}^{s\times s\times 3}$:

$$\hat{P}_{\text{s-AT}}(x,l) = \underset{P\in\{P':\|P'\|_\infty\leq\epsilon\}}{\arg\max} \mathcal{L}_{\text{BCE}}(PS_\theta(A(x,l,P)),M), \tag{6}$$

which is solved by PGD similar to Eq. (3). We train PS_θ in self adversarial training by solving:

$$\min_\theta \big[\lambda\mathbb{E}_{x\sim\mathcal{D}}\mathcal{L}_{\text{BCE}}(PS_\theta(x),M) + (1-\lambda)\mathbb{E}_{x\sim\mathcal{D},l\sim\mathcal{T}}\mathcal{L}_{\text{BCE}}(PS_\theta(A(x,l,\hat{P}_{\text{s-AT}}(x,l))),M)\big], \tag{7}$$

where $\mathcal{T}$ is the set of allowable patch locations, $\mathcal{D}$ is the image distribution, M is the ground-truth mask, and λ controls the weights between clean and adversarial images.

One alternative is to train PS_θ with patches generated by Eq. (2). Compared to Eq. (2), Eq. (6) does not require external labels since the ground-truth mask M is determined by l and known. Indeed, Eq. (7) trains the patch segmenter in a manner that no external label is needed for both crafting the adversarial samples and training the model; it strengthens PS_θ to detect any "patch-like" area in the images. Moreover, Eq. (6) does not involve the object detector h, which makes PS_θ object-detector agnostic and speeds up the optimization as the model size of PS_θ is much smaller than h.

The patch segmentation mask $\hat{M}_{PS}$ is obtained by thresholding the output of PS_θ: $\hat{M}_{PS} = PS_\theta(x) > 0.5$.

4.2. Shape Completion

4.2.1 Desired Properties

If we know that the adversary is restricted to attacking a patch of a specific shape, such as a square, we can use this information to "fill in" the patch-segmentation output $\hat{M}_{PS}$ to cover the ground truth patch mask M. We adopt a conservative approach: given $\hat{M}_{PS}$, we would like to produce an output $\hat{M}_{SC}$ which *entirely* covers the true patch mask M. In fact, we want to guarantee this property – however, if $\hat{M}_{PS}$ and M differ arbitrarily, then we clearly cannot provide any such guarantee. Because both the ground-truth patch mask M and the patch-segmentation output $\hat{M}_{PS}$ are binary vectors, it is natural to measure their difference as a Hamming distance $d_H(\hat{M}_{PS}, M)$. To provide appropriate scale, we compare this quantity to the total magnitude of the ground-truth mask $\|M\|_H := d_H(\mathbf{0}, M)$. We therefore would like a patch completion algorithm with the following property:

$$\text{If } \frac{d_H(\hat{M}_{PS},M)}{\|M\|_H} \leq \gamma \text{ then } \forall i,j : \hat{M}_{SC\,(i,j)} \geq M_{(i,j)} \tag{8}$$

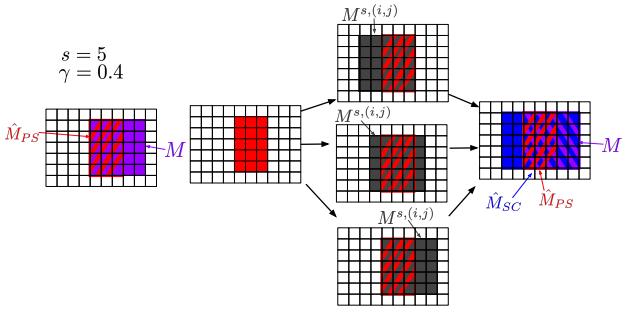


Figure 3. Construction of $\hat{M}_{SC}$ in Eq. (9): $\hat{M}_{SC}$ is the *union* of all candidate masks $M^{s,(i,j)}$ which are γ-close to $\hat{M}_{PS}$. If M is γ-close to $\hat{M}_{PS}$, this guarantees that M is covered by $\hat{M}_{SC}$.

4.2.2 Proposed Method

If the size of the ground-truth patch is known, then we can satisfy Eq. (8), *minimally*, by construction. In particular, suppose that M is known to be an $s \times s$ patch, and let $M^{s,(i,j)}$ refer to the mask of an $s \times s$ patch with upper-left corner at (i,j). Then Eq. (8) is minimally satisfied by the following mask:

$$\hat{M}_{SC\,(i,j)} := \begin{cases} 1 & \text{if } \exists i',j' : M^{s,(i',j')}_{(i,j)} = 1 \text{ and } \\ & \frac{d_H(\hat{M}_{PS},M^{s,(i',j')})}{s^2} \leq \gamma \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

where we have used that $\|M\|_H = s^2$. In other words, we must cover *every* pixel within *any* $s \times s$ patch $M^{s,(i',j')}$ that is γ-close to the observed mask $\hat{M}_{PS}$, because any such patch may in fact be M: a mask consisting of only these pixels is therefore the minimal mask necessary to satisfy Eq. (8). See Fig. 3 for an example. While Eq. (9) may appear daunting, there is a simple dynamic programming algorithm that allows the entire mask $\hat{M}_{SC}$ to be computed in $O(H \times W)$ time: this is presented in the supplementary material.

4.2.3 Unknown Patch Sizes

In Eq. (9), we assume that the ground-truth patch size s is known; and is further parameterized by the distortion threshold γ. Let $\hat{M}_{SC}(s,\gamma)$ represent this parameterized mask, as defined in Eq. (9). If we do not know s, but instead have a set of possible patch sizes S such that the true patch size $s \in S$, then we can satisfy Eq. (8) by simply combining all of the masks generated for each possible value of s:

$$\hat{M}_{SC}(S,\gamma)_{(i,j)} := \bigvee_{s\in S} \hat{M}_{SC}(s,\gamma)_{(i,j)} \tag{10}$$

Eq. (10) is indeed again the minimal mask required to satisfy the constraint: a pixel (i,j) is included in $\hat{M}_{SC}(\gamma)$ if and only if there exists some $M^{s,(i',j')}$, for some $s \in S$, such

that (i, j) is part of $M^{s,(i',j')}$ and $M^{s,(i',j')}$ is $\gamma-$close to $\hat{M}_{PS}$. In practice, this method can be highly sensitive to the hyperparameter γ. To deal with this issue, we initially apply Eq. (10) with low values of γ, and then gradually increase γ if no mask is returned – stopping when either some mask is returned or a maximum value is reached, at which point we assume that there is no ground-truth adversarial patch. The details can be found in the supplementary material.

4.2.4 Unknown Patch Shapes

In some cases, we may not know the shape of the patch. Since the patch segmenter is agnostic to patch shape, we use the union of $\hat{M}_{PS}$ and $\hat{M}_{SC}$ as the final mask output: $\hat{M} = \hat{M}_{PS} \bigcup \hat{M}_{SC}$. We empirically evaluate the effectiveness of this approach in Sec. 5.3.4.

5. Evaluation on Digital Attacks

In this section, we evaluate the robustness of SAC on digital patch attacks. We consider both non-adaptive and adaptive attacks, and demonstrate the generalizability of SAC.

5.1. Evaluation Settings

We use COCO [27] and xView [21] datasets in our experiments. COCO is a common object detection dataset while xView is a large public dataset of overhead imagery. For each dataset, we evaluate model robustness on 1000 test images and report mean Average Precision (mAP) at Intersection over Union (IoU) 0.5. For attacking, we iterate 200 steps with a set step size $\alpha = 0.01$. The patch location l is randomly selected within each image. We evaluate three rounds with different random patch locations and report the mean and standard deviation of mAP.

5.2. Implementation Details

All experiments are conducted on a server with ten GeForce RTX 2080 Ti GPUs. For base object detectors, we use Faster-RCNN [36] with feature pyramid network (RPN) [26] and ResNet-50 [18] backbone. We use the pretrained model provided in `torchvision` [31] for COCO and the model provided in `armory` [1] for xView. For patch segmenter, we use U-Net [37] with sixteen initial filters. To train the patch segmenters, for each dataset we generate 55k fixed adversarial images from the training set with patch size 100×100. Training on pre-generated adversarial images took around three hours on a single GPU. For self adversarial training, we train each model for one epoch by Eq. (7) using PGD attacks with 200 iterations and step size $\alpha = 0.01$ with $\lambda = 0.3$, which takes around eight hours on COCO and four hours on xView using ten GPUs. For patch completion, we use a square shape prior and the possible patch sizes $S = \{25, 50, 75, 100\}$ for xView and $S = \{25, 50, 75, 100, 125\}$ for COCO. More details can be found in the supplementary material.

5.3. Robustness Analysis

5.3.1 Baselines

We compare the proposed method with vanilla adversarial training (AT), JPEG compression [13], spatial smoothing [48], and LGS [33]. For AT, we use PGD attacks with thirty iterations and step size 0.067, which takes around twelve hours per epoch on the xView training set and thirty-two hours on COCO using ten GPUs. Due to the huge computational cost, we adversarially train Faster-RCNN models for ten epochs with pre-training on clean images. More details can be found in the supplementary material.

5.3.2 Non-adaptive Attack

The defense performance under non-adaptive attacks is shown in Tab. 1, where the attacker only attacks the object detectors. SAC is very robust across different patch sizes on both datasets and has the highest mAP compared to baselines. In addition, SAC maintains high clean performance as the undefended model. Fig. 4 shows two examples of object detection results before and after SAC defense. Adversarial patches create spurious detections and hide foreground objects. SAC masks out adversarial patches and restores model predictions. We provide more examples as well as some failure cases of SAC in the supplementary material.

5.3.3 Adaptive Attack

We further evaluate the defense performance under adaptive attacks where the adversary attacks the whole object detection pipeline. To adaptively attack preprocessing-based baselines (JPEG compression, spatial smoothing, and LGS), we use BPDA [4] assuming the output of each defense approximately equals to the original input. To adaptively attack SAC, we use straight-through estimators (STE) [5] when backpropagating through the thresholding operations, which is the strongest adaptive attack we have found for SAC (see the supplementary material for details). The results are shown in Tab. 1. The performances of preprocessing-based baselines drop a lot under adaptive attacks. AT achieves the strongest robustness among the baselines while sacrificing clean performance. The robustness of SAC has little drop under adaptive attacks and significantly outperforms the baselines. Since adaptive attacks are stronger than non-adaptive attacks, we only use adaptive attacks for the rest of the experiments.

5.3.4 Generalizability of SAC

Generalization to unseen shapes We train the patch segmenter with square patches and use the square shape prior

Table 1. mAP (%) under non-adaptive and adaptive attacks with different patch sizes. The best performance of each column is in **bold**.

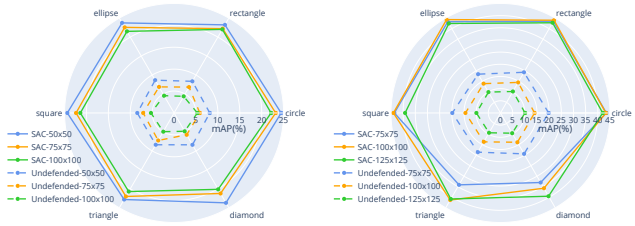
Dataset	Method	Clean	Non-adaptive Attack			Adaptive Attack		
			75×75	100×100	125×125	75×75	100×100	125×125
COCO	Undefended	**49.0**	19.8±1.0	14.4±0.6	9.9±0.5	19.8±1.0	14.4±0.6	9.9±0.5
	AT [30]	40.2	23.5±0.7	18.6±0.8	13.9±0.3	23.5±0.7	18.6±0.8	13.9±0.3
	JPEG [13]	45.6	39.7±0.3	37.2±0.3	33.3±0.4	22.8±0.9	18.0±0.8	13.4±0.7
	Spatial Smoothing [48]	46.0	40.4±0.6	38.1±0.6	34.3±0.1	23.2±0.7	17.5±1.0	13.5±0.6
	LGS [33]	42.7	36.8±0.1	35.2±0.6	32.8±0.9	20.8±0.7	15.9±0.5	12.2±0.9
	SAC (**Ours**)	**49.0**	**45.7±0.3**	**45.0±0.6**	**40.7±1.0**	**43.6±0.9**	**44.0±0.3**	**39.2±0.7**
Dataset	Method	Clean	Non-adaptive Attack			Adaptive Attack		
			50×50	75×75	100×100	50×50	75×75	100×100
xView	Undefended	**27.2**	8.4±1.6	7.1±0.4	5.3± 1.1	8.4±1.6	7.1±0.4	5.3± 1.1
	AT [30]	22.2	12.1±0.4	8.6±0.1	7.2±0.7	12.1±0.4	8.6±0.1	7.15±0.7
	JPEG [13]	23.3	19.3±0.4	17.8±1.0	15.9±0.4	11.2±0.3	9.5±1.0	8.3±0.3
	Spatial Smoothing [48]	21.8	16.2±0.7	14.2±1.1	12.4±0.8	11.0±0.7	7.9±0.6	6.5±0.2
	LGS [33]	19.1	11.9±0.5	10.9±0.3	9.8±0.5	8.2±0.8	6.5±0.4	5.4±0.5
	SAC (**Ours**)	**27.2**	**25.3±0.3**	**23.6±1.2**	**23.2±0.3**	**24.4±0.8**	**23.0±0.9**	**22.1±0.6**

(a) Ground-truth on clean image. (b) Predictions on clean image. (c) Predictions on adversarial image with a 100 × 100 patch. (d) Predictions on SAC masked image.

Figure 4. Visualization of object detection results with examples from COCO dataset (top) and xView dataset (bottom). Adversarial patches create spurious detections, and make the detector ignore the ground-truth objects. SAC masks out the patch and restores model predictions.

in shape completion. Since adversarial patches may not always be square in the real world, we further evaluate square-trained SAC with adversarial patches of different shapes while fixing the number of pixels in the patch. The details of the shapes used can be found in the supplementary material. We use the union of $\hat{M}_{PS}$ and $\hat{M}_{SC}$ as described in Sec. 4.2.4. The results are shown in Fig. 5. SAC demonstrates strong robustness under rectangle, circle, diamond, triangle, and ellipse patch attacks, even though these shapes mismatch with the square shape prior used in SAC.

Generalization to attack budgets In Eq. (2), we set $\epsilon = 1$ that allows the attacker to arbitrarily modify the pixel values within the patch region. In practice, the attacker may lower the attack budget to generate less visible adversarial patches to evade patch detection in SAC. To test how SAC generalizes to lower attack budgets, we evaluate SAC trained on $\epsilon = 1$ under lower ϵ values on the xView dataset. We set



(a) xView dataset. (b) COCO dataset.

Figure 5. Performance of SAC under adaptive attacks with different patch shapes and sizes. SAC demonstrates strong robustness under rectangle, circle and ellipse patch attacks, even though these shapes mismatch with the square shape prior used in SAC.

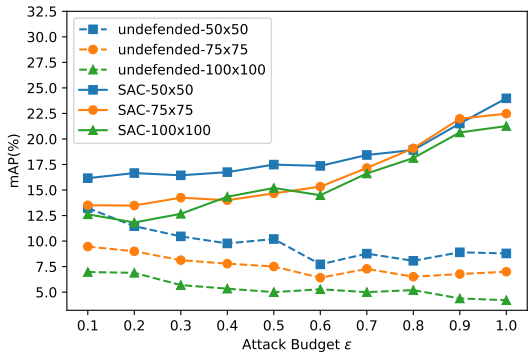


Figure 6. SAC performance under different attack budgets on xView dataset. SAC is trained with $\epsilon = 1$.

iteration steps to 200 and step size to $\epsilon/200$. Fig. 6 shows that SAC remains robust under a wide range of ϵ. Although the performance of SAC degrades when ϵ becomes smaller because patches become imperceptible, SAC still provides significant robustness gain upon undefended models. In addition, SAC is flexible and we can use smaller ϵ in training to provide better protection against imperceptible patches.

Generalization to unseen attack methods In the previous sections, we use PGD (Eq. (3)) to create adversarial patches. We further evaluate SAC under unseen attack methods, including DPatch [29] and MIM [12] attack. We use 200 iterations for both attacks, and set the learning rate to 0.01 for DPatch and decay factor $\mu = 1.0$ for MIM. The performance is shown in Tab. 2. SAC achieves more than 40.0% mAP on COCO and 21% mAP on xView under both attacks, providing strong robustness upon undefended models.

5.4. Ablation Study

In this section, we investigate the effect of each component of SAC. We consider three models: 1) patch segmenter trained with pre-generated adversarial images (PS); 2) PS further trained with self adversarial training (self AT); 3) Self AT trained PS combining with shape completion (SC), which is the whole SAC defense. The performance of these models under adaptive attacks is shown in Tab. 3. PS alone achieves good robustness under adaptive attacks (comparable or even better performance than the baselines in Tab. 1) thanks to the inherent robustness of segmentation models [3, 11]. Self AT significantly boosts the robustness, especially on the COCO dataset. SC further improves the robustness. Interestingly, we find that adaptive attacks on models with SC would force the attacker to generate patches that have more structured noises trying to fool SC (see supplementary material).

Table 2. mAP (%) under adaptive unseen attack methods with different patch sizes.

	Attack	Method	75×75	100×100	125×125
COCO	DPatch [29]	Undefended	33.6±0.8	29.1±0.6	25.0±1.7
		SAC (**Ours**)	45.3±0.3	44.1±0.6	42.1±0.8
	MIM [12]	Undefended	20.1±1.2	14.2±0.8	10.5±0.2
		SAC (**Ours**)	42.2±0.9	43.5±1.0	40.0±0.2
	Attack	Method	50×50	75×75	100×100
xView	DPatch [29]	Undefended	16.0±0.5	13.4±0.9	11.1±0.9
		SAC (**Ours**)	25.3±0.5	22.7±1.1	21.8±0.5
	MIM [12]	Undefended	8.3±0.4	7.3±0.8	6.5±1.5
		SAC (**Ours**)	24.7±0.7	23.0±0.9	22.1±0.6

Table 3. mAP (%) under adaptive attacks of ablated models.

	Method	75×75	100×100	125×125
COCO	Undefended	19.8±1.0	14.4±0.6	9.9±0.5
	PS	23.3±0.7	18.7±0.3	13.1±0.3
	+ self AT	41.5±0.2	40.5±0.6	36.6±0.1
	+ SC	**43.6±0.9**	**44.0±0.3**	**39.2±0.7**
	Method	50×50	75×75	100×100
xView	Undefended	8.4±1.6	7.1±0.4	5.3± 1.1
	PS	16.8±0.6	13.6±0.4	11.1±0.3
	+ self AT	20.6±0.4	17.6±0.5	15.4±0.6
	+ SC	**24.4±0.8**	**23.0±0.9**	**22.1±0.6**

6. Evaluation on Physical Attack

In this section, we evaluate the robustness of SAC on physical patch attacks. We first introduce the APRICOT-Mask dataset and further demonstrate the effectiveness of SAC on the APRICOT dataset.

6.1. APRICOT-Mask Dataset

APRICOT [6] contains 1,011 images of sixty unique physical adversarial patches photographed in the real world, of which six patches (138 photos) are in the development

set, and the other fifty-four patches (873 photos) are in the test set. APRICOT provides bounding box annotations for each patch. However, there is no pixel-level annotation of the patches. We present the APRICOT-Mask dataset [1], which provides segmentation masks and more accurate bounding boxes for adversarial patches in the APRICOT dataset (see two examples in Fig. 7). The segmentation masks are annotated by three annotators using Labelbox [2] and manually reviewed to ensure the annotation quality. The bounding boxes are then generated automatically from the segmentation masks. We hope APRICOT-Mask along with the APRICOT dataset can facilitate the research in building defenses against physical patch attacks, especially patch detection and removal techniques.

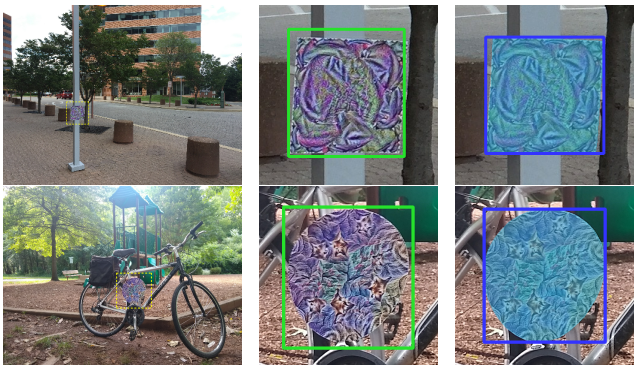


Figure 7. Images and patch annotations from the APRICOT and APRICOT-Mask datasets. Left: adversarial images from the APRICOT dataset; middle: patch bounding boxes provided by the APRICOT dataset; right: patch bounding boxes and segmentation masks provided by the APRICOT-Mask dataset.

6.2. Robustness Evaluation

Evaluation Metrics We evaluate the defense effectiveness by the targeted attack success rate. A patch attack is "successful" if the object detector generates a detection that overlaps a ground truth adversarial patch bounding box with an IoU of at least 0.10, has a confidence score greater than 0.30, and is classified as the same object class as the patch's target [6].

Evaluation Results We train the patch segmenter on the APRICOT test set using the segmentation masks from the APRICOT-Mask dataset. The training details can be found in the supplementary material. Since APRICOT patches are generated from three detection models trained on the COCO dataset targeting ten COCO object categories, we use a Faster-RCNN model pretrained on COCO [31] as our base object detector, which is a black-box attack setting with target and substitute models trained on the same dataset. We evaluate the targeted attack success rate on the development set and compare SAC with the baselines as in Sec. 5.3.1. For AT, we use the Faster-RCNN model adversarially trained on the COCO dataset (AT-COCO) as the size of the APRICOT dataset is not enough to retrain an object detector. The results are shown in Fig. 8. SAC significantly brings down the targeted attack success rate of the undefended model from 7.97% to 2.17%, which is the lowest among all defense methods. AT has a slightly higher targeted attack success rate than the undefended model, which may be due to the domain gap between COCO and APRICOT datasets.

[1] https://aiem.jhu.edu/datasets/apricot-mask

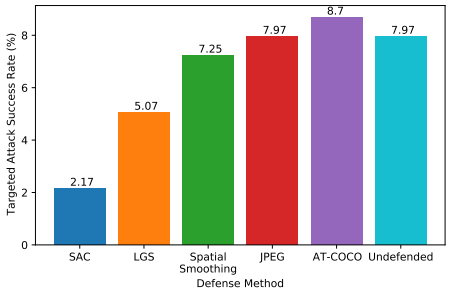


Figure 8. Targeted attack success rates on the APRICOT dataset.

7. Discussion and Conclusion

In this paper, we propose the Segment and Complete defense that can secure any object detector against patch attacks by robustly detecting and removing adversarial patches from input images. We train a robust patch segmenter and exploit patch shape priors through a shape completion algorithm. Our evaluation on digital and physical attacks demonstrates the effectiveness of SAC. In addition, we present the APRICOT-Mask dataset to facilitate the research in building defenses against physical patch attacks.

SAC can be improved in several ways. First, although SAC does not require re-training of base object detectors, fine-tuning them on images with randomly-placed black blocks can further improve their performance on SAC masked images. Second, in this paper, we adopt a conservative approach that masks out the entire patch region after we detect the patch. This would not cause information loss when the attacker is allowed to arbitrarily distort the pixels and destroy all the information within the patch such as in physical patch attacks. However, in the case where the patches are less visible, some information may be preserved in the patched area. Instead of masking out the patches, one can potentially impaint or reconstruct the content within the patches, which can be the future direction of this work.

Acknowledgment This work was supported by the DARPA GARD Program HR001119S0026-GARD-FP-052.

References

[1] Armory testbed. https://armory.readthedocs.io/. Accessed: 11/8/2021. 5

[2] Labelbox. https://labelbox.com/. Accessed: 11/8/2021. 8

[3] Anurag Arnab, Ondrej Miksik, and Philip HS Torr. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 888–897, 2018. 7

[4] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018. 5

[5] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 5

[6] A Braunegg, Amartya Chakraborty, Michael Krumdick, Nicole Lape, Sara Leary, Keith Manville, Elizabeth Merkhofer, Laura Strickhart, and Matthew Walmer. Apricot: A dataset of physical adversarial attacks on object detection. In *European Conference on Computer Vision*, pages 35–50. Springer, 2020. 2, 7, 8

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

[8] Pin-Chun Chen, Bo-Han Kung, and Jun-Cheng Chen. Class-aware robust adversarial training for object detection. *arXiv preprint arXiv:2103.16148*, 2021. 2

[9] 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. 1, 2

[10] Ping-yeh Chiang, Michael J Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. Detection as regression: Certified object detection by median smoothing. *arXiv preprint arXiv:2007.03730*, 2020. 2

[11] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6980–6990, Red Hook, NY, USA, 2017. Curran Associates Inc. 7

[12] 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. 7

[13] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of JPG compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016. 5, 6

[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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 2

[15] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 3

[16] Thomas Gittings, Steve Schneider, and John Collomosse. Vax-a-net: Training-time defence against adversarial patch attacks. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 1, 2

[17] Jamie Hayes. On visible adversarial perturbations & digital watermarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1597–1604, 2018. 1, 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[19] 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

[20] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. In *International Conference on Machine Learning*, pages 2507–2515. PMLR, 2018. 2, 3

[21] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xView: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018. 5

[22] Dapeng Lang, Deyun Chen, Ran Shi, and Yongjun He. Attention-guided digital adversarial patches on visual detection. *Security and Communication Networks*, 2021, 2021. 1, 2, 3

[23] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019. 1, 2, 3

[24] Alexander Levine and Soheil Feizi. (De) Randomized Smoothing for Certifiable Defense against Patch Attacks. *arXiv preprint arXiv:2002.10733*, 2020. 1, 2

[25] Yuezun Li, Xiao Bian, Ming-Ching Chang, and Siwei Lyu. Exploring the vulnerability of single shot module in object detectors via imperceptible background patches. *arXiv preprint arXiv:1809.05966*, 2018. 1, 2, 3

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 5

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 5

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 3

[29] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. DPatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018. 1, 2, 3, 7

[30] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 3, 6

[31] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery. 5, 8

[32] Michael McCoyd, Won Park, Steven Chen, Neil Shah, Ryan Roggenkemper, Minjune Hwang, Jason Xinyu Liu, and David Wagner. Minority reports defense: Defending against adversarial patches. In Jianying Zhou, Mauro Conti, Chuadhry Mujeeb Ahmed, Man Ho Au, Lejla Batina, Zhou Li, Jingqiang Lin, Eleonora Losiouk, Bo Luo, Suryadipta Majumdar, Weizhi Meng, Martín Ochoa, Stjepan Picek, Georgios Portokalidis, Cong Wang, and Kehuan Zhang, editors, *Applied Cryptography and Network Security Workshops*, pages 564–582, Cham, 2020. Springer International Publishing. 1, 2

[33] 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. IEEE, 2019. 2, 5, 6

[34] Sukrut Rao, David Stutz, and Bernt Schiele. Adversarial training against location-optimized adversarial patches. *arXiv preprint arXiv:2005.02313*, 2020. 1, 2

[35] 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. 2

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 91–99, Cambridge, MA, USA, 2015. MIT Press. 3, 5

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 3, 5

[38] Aniruddha Saha, Akshayvarun Subramanya, Koninika Patil, and Hamed Pirsiavash. Role of spatial context in adversarial robustness for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 784–785, 2020. 1, 2, 3

[39] 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. CCS '16, page 1528–1540, New York, NY, USA, 2016. Association for Computing Machinery. 1, 2

[40] 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. 1, 2

[41] 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

[42] Abdul Vahab, Maruti S Naik, Prasanna G Raikar, and Prasad SR. Applications of object detection system. *International Research Journal of Engineering and Technology (IRJET)*, 6(4):4186–4192, 2019. 1

[43] Yajie Wang, Haoran Lv, Xiaohui Kuang, Gang Zhao, Yu-an Tan, Quanxin Zhang, and Jingjing Hu. Towards a physical-world adversarial patch for blinding object detection models. *Information Sciences*, 556:459–471, 2021. 1, 2

[44] Tong Wu, Liang Tong, and Yevgeniy Vorobeychik. Defending against physically realizable attacks on image classification. *arXiv preprint arXiv:1909.09552*, 2019. 1, 2

[45] 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

[46] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking. In *30th USENIX Security Symposium (USENIX Security)*, 2021. 1, 2

[47] Chong Xiang and Prateek Mittal. Detectorguard: Provably securing object detectors against localized patch hiding attacks. *arXiv preprint arXiv:2102.02956*, 2021. 2

[48] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017. 5, 6

[49] Ping yeh Chiang*, Renkun Ni*, Ahmed Abdelkader, Chen Zhu, Christoph Studor, and Tom Goldstein. Certified defenses for adversarial patches. In *International Conference on Learning Representations*, 2020. 1, 2, 3

[50] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 421–430, 2019. 2

[51] Yusheng Zhao, Huanqian Yan, and Xingxing Wei. Object hider: Adversarial patch attack against object detectors. *arXiv preprint arXiv:2010.14974*, 2020. 1, 2, 3