NAPGuard: Towards Detecting Naturalistic Adversarial Patches

Siyang Wu[1], Jiakai Wang[2,*], Jiejie Zhao[2], Yazhe Wang[2], Xianglong Liu[1,2,3]
[1]State Key Laboratory of Complex & Critical Software Environment, Beihang University, Beijing, China
[2]Zhongguancun Laboratory, Beijing, China
[3]Institute of data space, Heifei Comprehensive National Science Center, Anhui, China
{wusiyang, xlliu}@buaa.edu.cn, {wangjk, zhaojiejie, wangyz}@zgclab.edu.cn

Abstract

Recently, the emergence of naturalistic adversarial patch (NAP), which possesses a deceptive appearance and various representations, underscores the necessity of developing robust detection strategies. However, existing approaches fail to differentiate the deep-seated natures in adversarial patches, i.e., aggressiveness and naturalness, leading to unsatisfactory precision and generalization against NAPs. To tackle this issue, we propose NAPGuard to provide strong detection capability against NAPs via the elaborated critical feature modulation framework. ***For improving precision****, we propose the aggressive feature aligned learning to enhance the model's capability in capturing accurate aggressive patterns. Considering the challenge of inaccurate model learning caused by deceptive appearance, we align the aggressive features by the proposed pattern alignment loss during training. Since the model could learn more accurate aggressive patterns, it is able to detect deceptive patches more precisely.* ***To enhance generalization****, we design the natural feature suppressed inference to universally mitigate the disturbance from different NAPs. Since various representations arise in diverse disturbing forms to hinder generalization, we suppress the natural features in a unified approach via the feature shield module. Therefore, the models could recognize NAPs within less disturbance and activate the generalized detection ability. Extensive experiments show that our method surpasses state-of-the-art methods by large margins in detecting NAPs (improve* ***60.24%*** *AP@0.5 on average).*[1]

1. Introduction

Deep neural networks (DNNs) have been widely applied in real-world scenarios [9, 29–31, 49]. Despite their remarkable performance, DNNs are known for their vulnerability to adversarial attacks [7, 18, 35], which hinders their reliability. Adversary patches, as a crucial form of physical adversarial attacks, pose a serious threat to the security of computer vision models and applications, including object detection [38, 42, 43, 47], crowd counting [27], vision transformer [44], x-ray detection [24], *etc*.

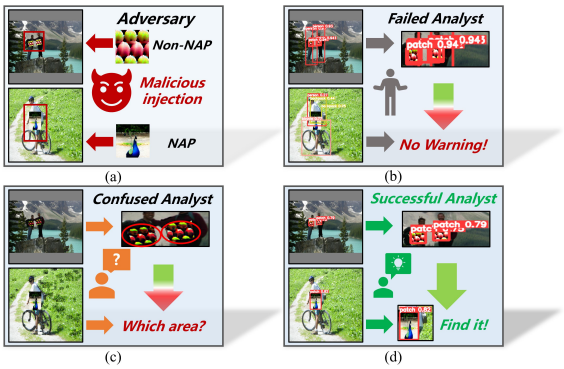


Figure 1. (a) displays the adversarial examples generated by Non-NAP (*i.e.*, AdvTexture [11]) and NAP (*i.e.*, GNAP [10]). (b) shows the insufficient generalization of current model-assisted method for NAPs (*i.e.*, Ad-YOLO [14]). (c) shows the limited precision of image analysis method for NAPs (*i.e.*, LGS [33]). (d) shows the success of our NAPGuard on both Non-NAPs and NAPs.

In the past years, numerous efforts have been made to detect physical adversarial patches [2, 14, 16, 26, 32, 33, 37, 46, 48]. Previous methods focus solely on detecting the presence of adversarial patches [2, 32, 46], neglecting their precise location, which reduces the task to a classification problem. To address this issue, current detection methods aim to locate the adversarial patches, which plays a crucial role in accurately detecting the source of malicious content within an image. Moreover, these methods can be served as a pre-processing step in existing defense methods (*e.g.*, image denoising), thereby enhancing their potential for practical applications. Generally, current detection methods can be divided into two categories: image analysis approaches [16, 33, 37] and model-assisted approaches [14, 26, 48].

[1]Our code is available at https://github.com/wsynuiag/NAPGaurd.

*Corresponding author

The former detects through analyzing anomalies in the image, such as gradients [33] and features [16], while the latter involves training a deep learning model, such as an image segmentation model [26, 48] or an object detection model [14], to aid in the detection of adversarial patches.

Though showing certain results, existing detection methods still face notable limitations in practice: low precision and insufficient generalization (Fig. 1) on naturalistic adversarial patches (NAPs) [10, 20, 36] generated by current methods. In comparison to non-naturalistic adversarial patch (Non-NAP), NAP possesses two prominent characteristics: (1) a deceptive appearance, which resembles natural images and could potentially deceive models into learning inaccurate patterns, resulting in low precision, and (2) various representations, which stem from the abundant object categories in nature, leading to diverse disturbing forms that pose challenges to generalization. However, these characteristics remain insufficiently considered by current defense methods [14, 26, 33, 48], resulting in their failure to effectively detect NAPs.

To address this problem, this paper proposes NAPGuard to provide strong detection capability against NAPs via the elaborated critical feature modulation framework. **To improve precision**, we propose the aggressive feature aligned learning strategy to enhance the model's capability in capturing accurate aggressive patterns. Considering the challenge of inaccurate model learning posed by the deceptive appearance, we align the aggressive features by the proposed pattern alignment loss during training. Inspired by previous findings [6, 34, 41] that aggressive features primarily reside in the high-frequency components, we realize this alignment from a high-frequency perspective. This alignment helps the model recognize aggressive patterns more accurately, thus enabling it to precisely detect deceptive patches. **For enhancing generalization**, we introduce the natural feature suppressed inference strategy to universally mitigating the disturbance from different NAPs. In view of the diverse disturbing forms caused by various representations, we suppress the natural features in a unified approach. In practice, inspired by the pop-out effect in biology, wherein attention rapidly detects features that significantly deviate from others in a visual display [39], we consider to amplify the differences between natural and aggressive features by designing the feature shield module. Under a condition with less disturbance, the model could better capture aggressive features and activate the generalized detection ability. As shown in Fig. 1, our proposed framework achieves better detection performance for NAPs.

Our main **contributions** can be summarized as follows:

- To the best of our knowledge, we are the first to explore this issue from the perspective of aggressive and natural features, which allows us to revisit the natures of NAPs.
- We propose the NAPGuard, an elaborated critical feature modulation framework to effectively detect NAPs by aligning aggressive features and suppressing natural features during training and inference, respectively.
- We construct the first generalized adversarial patch detection (GAP) dataset, which contains 25 distinct adversarial patches and over 9000 images, to facilitate future investigations in physical adversarial patch detection.
- Extensive experiments demonstrate that our method surpasses state-of-the-art methods by large margins in detecting NAPs (**60.24%** AP@0.5 improvement).

2. Related Works

2.1. Physical Adversarial Patch Attack

Extensive studies have shown that DNNs are vulnerable to adversarial patch attacks [1, 17, 22, 23, 28, 38]. These localized patches could easily manipulate the predictions made by the models. Early studies have introduced numerous methods [1, 22, 23, 28] to generate localized adversarial patches in the digital world. Beyond digital world, adversarial patch techniques have also been applied to real-world scenarios [11, 12, 17, 38, 47], posing a greater threat to the reliable application of DNNs. Recently, researchers have focused significantly on wearable physical adversarial patches similar to an "invisibility cloak" [11, 45, 47] by combining image warping methods, which effectively evade model detection.

Besides wearability, another research hotspot of physical adversarial patch attack is naturalness, which aims to evade human eyes [4, 10, 20, 36]. In order to enhance the naturalness of adversarial patches, researchers have leveraged the power of generative networks, including generative adversarial network (GAN) [4, 10, 36] and diffusion models [20] to generate patches that closely resemble real-world examples. Several related works have emerged [5, 13, 22], which have significantly advanced the concealment capabilities of adversarial patches. In summary, NAPs have presented a formidable challenge, necessitating the development of more powerful defense strategies.

2.2. Physical Adversarial Patch Detection

Researchers have extensively explored adversarial attacks and developed defense methods [8, 19, 25, 31]. As for adversarial patches, previous detection methods mainly concentrated on detecting adversarial examples [2, 32, 46], but lacked precise patch localization. To tackle this issue, current methods focus on detecting these patches before model input. These methods can be mainly divided into two mainstreams: (1) image analysis approaches, which detect through abnormal components of the image, including gradients [33], features [16], entropy [37], *etc*. (2) model-assisted approaches [14, 26, 48], which involve training a deep learning model, such as an image segmentation model

[26, 48] or an object detection model [14], to aid in the detection of adversarial patches.

However, current detection methods fail to differentiate between aggressiveness and naturalness in adversarial patches, resulting in their failure when detecting NAPs. In this paper, we focus on these deep-seated natures and design a detection framework to effectively detect NAPs.

3. Methodology

In this section, we first provide the definition of the problem and then elaborate on our proposed framework.

3.1. Problem Definition

Given a victim model $\mathbb{F}_\theta$ and an input clean image $\mathbf{I}$ with the ground truth label y, an adversarial example $\mathbf{I}_{adv}$ can mislead the model to provide wrong predictions, *i.e.*, $\mathbb{F}_\theta(\mathbf{I}_{adv}) \neq y$. As for adversarial patch attack, let $\mathbf{P}$ denote an adversarial patch. The attacker applies the adversarial patch $\mathbf{P}$ to the input image $\mathbf{I}$ through an applier $\mathcal{A}$ with transformations $t \in \mathbb{T}$ (rotation, scaling, *etc.*). Then, the adversarial example can be denoted as $\mathbf{I}_{adv} = \mathcal{A}(\mathbf{I}, \mathbf{P}, t)$.

In this study, we acquire an adversarial patch detection model (*i.e.*, the "**detector**") by redefining category 0 of an object detection model as the "patch" category, discarding other categories and training it with an adversarial example dataset. Given a patch detector $\mathbb{M}_\theta$ with loss function $\mathcal{L}$ and a training set of adversarial examples $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ with ground truth labels $\{y_1, y_2, ..., y_n\}$, then the training process can be defined as

$$\min_\theta E_{x_i \sim \mathcal{X}}[\mathcal{L}(\mathbb{M}_\theta(x_i), y_i)]. \tag{1}$$

Given an image x^* from real-world, then the inference process can be defined by $\mathbb{M}_\theta(x^*) = y^*$, where y^* denotes the patch locations. In this paper, we consider both training and inference processes to provide a comprehensive detection framework against NAPs.

3.2. Framework Overview

In order to developing effective detection methods against NAPs, we divide the feature space into two distinct categories: aggressive features, which contribute to adversarial behaviors, and natural features, which correlate to naturalness. Then, we propose the NAPGuard, an elaborated critical feature modulation framework to effectively detect NAPs. The overall framework can be found in Fig. 2.

For improving the precision, we propose the *aggressive feature aligned learning* (**AFAL**) strategy. Given the inaccurate model learning challenge posed by deceptive appearance, we align the aggressive features by the proposed pattern alignment loss during training. Inspired by previous findings [6, 34, 41] that aggressive features of adversarial examples primarily reside in the high-frequency components, we realize this alignment from a high-frequency perspective. By introducing this alignment, the detector can better recognize aggressive patterns, thus improving its capability to precisely detect more deceptive NAPs.

For enhancing the generalization, we introduce the *natural feature suppressed inference* (**NFSI**) strategy. Considering the diverse disturbing forms caused by various representations, we adopt a unified approach to suppress the natural features, universally mitigating the disturbance from different NAPs. Inspired by the observation that attention rapidly detects features that significantly deviate from others in a visual display [39], we amplify the differences between natural and aggressive features by designing the feature shield module. By applying this module during inference, we can provide a condition that enables the model to capture aggressive features with less disturbance, thus activating the generalized detection ability for various NAPs.

3.3. Aggressive Feature Aligned Learning

Several previous studies have pointed out that aggressive features of adversarial examples primarily reside in the high-frequency components [6, 34, 41]. Since adversarial patches are localized adversarial attacks, it is reasonable to assume that the aggressive features of adversarial patches exhibit similar characteristics. From this perspective, we observe that the high-frequency components of NAPs are more similar to the surroundings compared to Non-NAPs, which makes them more deceptive and leads to inaccurate learning (see discussions in Sec. 5.3.2). Based on this observation, we come up with an intuitive idea: if the detector could recognize aggressive patterns within environments aligned with naturalistic adversarial examples (NAEs), where adversarial patches share similar high-frequency components with the surroundings, it may perform better in detecting deceptive NAPs. Therefore, to improve precision, we consider to enhance the detector's capability in capturing accurate aggressive patterns by aligning the aggressive features during the training process.

In practice, given a set of adversarial examples $x = \{x_1, x_2, ..., x_n\}$ from a training set $\mathcal{X}$, a feature extraction function $\mathcal{F}(\cdot)$ (*e.g.*, modules or backbone of neural networks), we utilize a Laplacian operator ∇^2 to enhance the high-frequency components within the image, thus increasing the similarity between adversarial patches and the surroundings (see discussions in Sec. 5.3.2). Then, we modify the detector's loss function to help the detector capture accurate aggressive patterns during training. Since the mean squared error (MSE) is widely used to quantify differences of features (*i.e.*, lower MSE indicates higher similarity between features), we introduce the pattern alignment loss, an MSE loss between feature maps obtained from normal images and aligned images. Formally, our pattern alignment

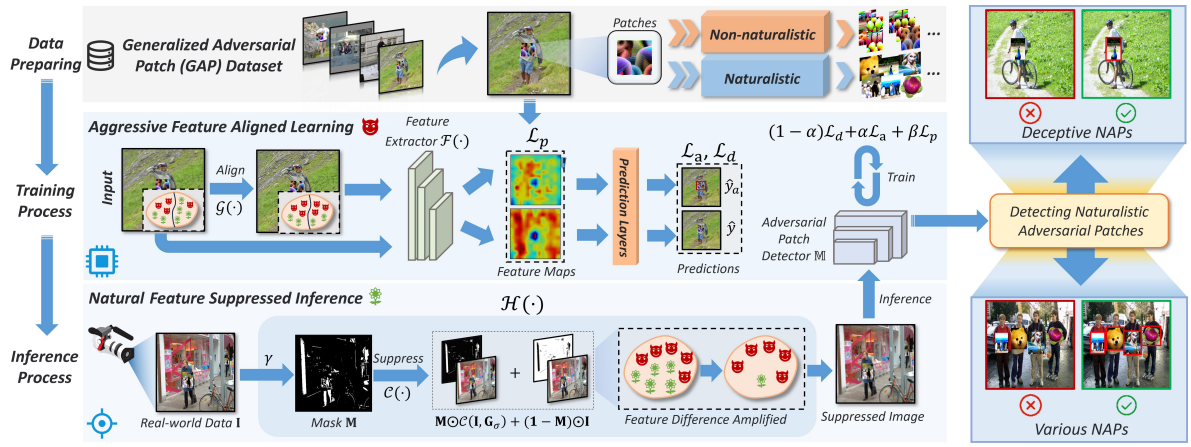


Figure 2. The framework of our NAPGuard. We first enhance the detector's capability in capturing accurate aggressive patterns via aggressive feature aligned learning strategy. Further, we universally mitigate the disturbance from different NAPs via natural feature suppressed inference strategy. Benefiting from synergistic modulation, the proposed NAPGuard can effectively detect NAPs.

loss can be formulated as

$$\mathcal{L}_p = \frac{1}{n}\sum_{i=1}^{n}\|\mathcal{F}(x_i) - \mathcal{F}(\mathcal{G}(x_i, \nabla^2))\|^2, \\ \mathcal{G}(x_i, \nabla^2) = x_i - \nabla^2 x_i. \tag{2}$$

By minimizing $\mathcal{L}_p$, we facilitate the detector to accurately capture aggressive patterns from environments aligned with NAEs, thus enabling it to precisely detect deceptive patches. For further alignment, we input $\mathcal{G}(x_i, \nabla^2)$ as an auxiliary branch into the detector and incorporate its detection loss into the existing optimization process. In summary, we can train a stronger detector by minimizing the following loss function:

$$\min(1-\alpha)\mathcal{L}_d + \alpha\mathcal{L}_a + \beta\mathcal{L}_p, \\ \mathcal{L}_a = \mathcal{L}_d(\mathcal{G}(x_i, \nabla^2)), \tag{3}$$

where $\mathcal{L}_d$ is the original detection loss of the detector, $\mathcal{L}_a$ is the auxiliary detection loss, while α and β controls the contribution of the terms $\mathcal{L}_a$ and $\mathcal{L}_p$, respectively.

3.4. Natural Feature Suppressed Inference

As adversarial patch attack techniques continue to advance, it is inevitable that more NAPs with various representations will emerge in the future. These patches may introduce unknown and diverse disturbing forms to the detector, posing a challenge to its generalized detection ability. To enhance generalization, we aim to mitigate these diverse disturbance universally by suppressing the natural features in a unified approach during the inference process.

Since aggressive features primarily reside in the high-frequency components [6, 34, 41], we delve into the low-frequency domain to suppress natural features. Inspired by the pop-out effect in biology [39], we design a feature shield module $\mathcal{H}(\cdot)$ to universally mitigate the diverse disturbance by suppressing natural features, thus amplifying the differences between natural and aggressive features. For an image sampled in real-world scenarios, we first obtain the filtered image with low-frequency components by applying a low-pass filter. Then, we create a mask by selecting regions that contain rich natural features for further processing. Last, we smooth out the natural details in these regions, thereby amplifying the difference between natural and aggressive features.

Specifically, given an adversarial example $\mathbf{I}(x, y)$, where (x, y) denotes the pixel position, we first obtain its frequency domain representation $\mathbf{F}(u, v)$ by applying a two-dimensional Fast Fourier Transform (FFT). To separate the low-frequency components, we perform an element-wise multiplication between $\mathbf{F}(u, v)$ and a circular low-pass filter $\mathbf{R}_L(u, v)$. Last, we perform an inverse FFT on the result, yielding the filtered image $\mathbf{I}_{low}(x, y)$. This process can be formulated as

$$\mathbf{F}(u, v) = \mathcal{T}(\mathbf{I}(x, y)), \\ \mathbf{I}_{low}(x, y) = \mathcal{T}^{-1}(\mathbf{F}(u, v) \odot \mathbf{R}_L(u, v)), \tag{4}$$

where $\mathcal{T}$ denotes FFT and $\odot$ is the element-wise multiplication. In order to enable directional smoothing of natural features, we improve the Gaussian blurring function by a region selecting method. This allows for more targeted suppression of natural features while preserving the saliency of aggressive features in other regions. In detail, given a filtered image $\mathbf{I}_{low}(x, y)$ calculated by Eq. (4), we generate a mask $\mathbf{M}(i, j)$ through selecting regions that contain rich natural features by a thresholding operation:

$$\tau = \mu_{low} + \gamma\sigma_{low}, \\ \mathbf{M}(i, j) := \begin{cases} 1, if \quad |\mathbf{I}_{low}(x, y)| > \tau \\ 0, otherwise \end{cases}, \tag{5}$$

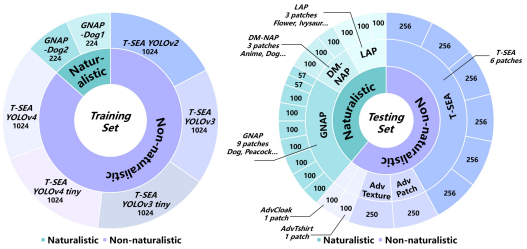


Figure 3. The statistics of GAP dataset. Each part of the outermost ring represents the number of adversarial examples generated by the corresponding adversarial patch.

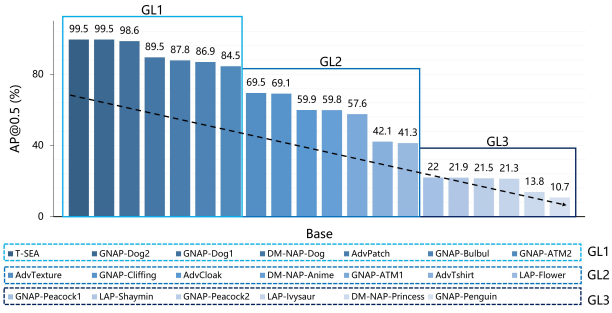


Figure 4. The generalization (AP@0.5) of the base detector (defined in Sec. 5.1) on each type of adversarial patches, sorted in descending order. We also display the distribution of different GLs, divided according to the generalizable performance.

where μ_{low} and σ_{low} denote the mean and standard variation of $\mathbf{I}_{low}$, respectively, $|\cdot|$ represents the absolute value operation, γ is an empirical weight to appropriately control the threshold. Then, we suppress the natural features according to the mask as

$$\mathbf{I}_s = \mathcal{H}(\mathbf{I}) = \mathbf{M} \odot \mathcal{C}(\mathbf{I}, \mathbf{G}_\sigma) + (\mathbf{1} - \mathbf{M}) \odot \mathbf{I}, \quad (6)$$

where $\mathcal{C}$ denotes a two-dimensional convolution, $\mathbf{I}_s$ is the suppressed image, and $\mathbf{G}_\sigma$ is a Gaussian kernel constructed using a standard deviation parameter σ to determine the spread of the Gaussian distribution.

In summary, through utilizing the feature shield module $\mathcal{H}(\cdot)$ during inference, we suppress the natural features in a unified approach, thus facilitating the detector to recognize NAPs and improving the generalizable performance.

Overall, we first enhance the detector's capability in capturing aggressive patterns by jointly optimizing the loss terms $\mathcal{L}_d$, $\mathcal{L}_p$ and $\mathcal{L}_a$ during training. During inference, we universally mitigate the disturbance of natural features by utilizing the feature shield module $\mathcal{H}(\cdot)$ to improve the generalization.[2]

4. Generalized Adversarial Patch Dataset

To address the lack of datasets in physical adversarial patch detection, we introduce the Generalized physical Adversarial Patch detection (GAP) dataset, aiming to provide an evaluation benchmark for future detection approaches.

4.1. Construction Principles

We construct our GAP dataset following the four principles:

- **Data Legality.** All images and adversarial patches used in this dataset are sourced exclusively from open datasets and published papers, complying with data legality regulations. Specifically, the images are derived from the testing set of INRIA-Person [3] and MS COCO [21] datasets.
- **Extensive Diversity.** GAP contains 25 types of distinct adversarial patches from 8 methods including 15 NAPs and 10 Non-NAPs, allowing for a comprehensive evaluation of models' generalizable performance across various types of adversarial patches. In practice, we choose 9 patches from GNAP [10], 6 patches from T-SEA [12], 3 patches each from DM-NAP [20] and LAP [36], and 1 patch each from AdvPatch [38], AdvCloak [45], AdvTshirt [47] and AdvTexture [11].
- **Professional Annotation.** GAP contains professionally annotated adversarial patches, ensuring accurate and reliable labeling, which can serve as a high-quality resource for evaluating adversarial patch detection methods.
- **Explicit Task.** The construction of the dataset should align with the specific problem to be solved so that relevant experimental results can evaluate the effectiveness for addressing this problem. To better evaluate the generalized detection ability, especially for NAPs, we categorize the testing set into three subsets according to the generalizable performance and name them in grades: Generalization Level 1 (GL1), Generalization Level 2 (GL2) and Generalization Level 3 (GL3), where higher level represents poorer generalization. The generalizable performance of each adversarial patch and the distribution of each subset are shown in Fig. 4.

4.2. Data Properties

The GAP dataset contains 9266 images and 25 types of adversarial patches in total. Every adversarial patch is located with a bounding-box annotation. The statistics of adversarial patches in training set and testing set are shown in Fig. 3. All images are stored in PNG format with a fixed size of 416 $\times$ 416 pixels by padding or resizing, which align with the settings described in the respective papers. The dataset is partitioned into a training set (5617 images) and a testing set (3649 images), following a ratio of 6:4. Note that GAP dataset aims to evaluate models' generalizable performance comprehensively, so the testing set contains a diverse range of adversarial patches, which is why the training-to-testing ratio is 6:4 rather than the typical 4:1 ratio.

[2]The overall algorithm can be found in Supplementary Materials.

Table 1. The experimental results (AP@0.5↑) of our proposed NAPGuard and compared baselines on both Non-NAPs and NAPs. The **bold** values represent the highest value in each column, *i.e.*, best performance. "Mixture" represents a mixture set of all these patch types.

Method	Patch Type								
	Non-NAPs					NAPs			Mixture
	T-SEA [12]	AdvPatch [38]	AdvCloak [45]	AdvTshirt [47]	AdvTexture [11]	GNAP [10]	DM-NAP [20]	LAP [36]	
LGS [33]	2.95	8.33	10.14	12.85	13.13	4.09	6.38	4.39	5.71
APE [16]	2.24	4.27	4.26	54.70	31.83	12.09	9.34	7.28	9.36
SAC [26]	0.00	5.05	0.00	18.97	41.54	0.00	0.00	0.00	20.31
PatchZero [48]	4.08	0.00	0.00	0.00	11.65	0.00	0.00	0.00	10.70
Ad-YOLO [14]	94.79	47.69	10.29	35.89	77.03	34.04	40.51	17.74	54.88
Base	**99.42**	87.83	59.94	42.07	69.51	68.00	54.48	36.23	76.84
Ours	98.37	**96.95**	**92.24**	**69.53**	**94.20**	**88.27**	**98.66**	**86.07**	**92.24**

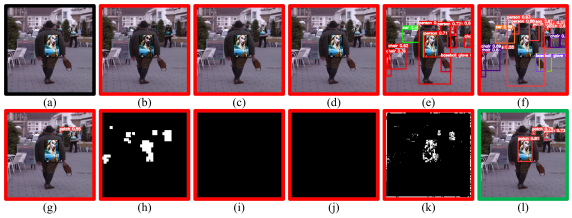


Figure 5. Detection results of our proposed method and compared baselines. (a): Original adversarial example, sampled from GAP. (b) and (h): LGS and its mask. (c) and (i): SAC and its mask. (d) and (j): PatchZero and its mask. (e) and (k): APE and its mask. (f) Ad-YOLO. (g): Base detector. (l) Ours. Our NAPGuard precisely detects NAPs. Best in view.

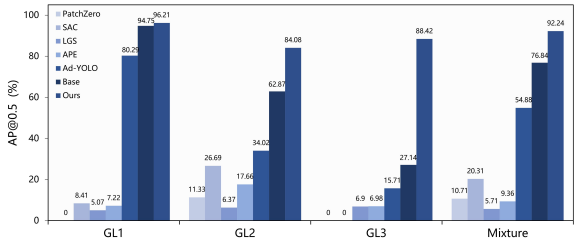


Figure 6. Evaluation results of our method and compared baselines (AP@0.5 ↑) on our GAP dataset. "Ours" achieves better performance than all compared baselines.

5. Experimental Results

In this section, we first outline our experimental settings, then report the effectiveness of our proposed detection framework in various settings.

5.1. Experimental Settings

5.1.1 Models and Datasets

We conduct experiments on our proposed GAP dataset. As for the model, we obtain a patch detector through converting the common used object detection model YOLOv5 [15] to a single-class model and training it on our GAP dataset. Note that the "Base" model refers to the patch detector directly trained on our GAP dataset, which serves as a baseline. As for evaluation metrics, we select the widely used Average Precision (AP@0.5) from detection task, which reflects both the IoU and precision information.

5.1.2 Compared Baselines

We choose several state-of-the-art adversarial patch defense methods as the compared baselines, including LGS [33], Ad-YOLO [14], APE [16], SAC [26] and PatchZero [48]. For Ad-YOLO, which considers the task as an object detection task similar to our approach, we use YOLOv5 as the network architecture and train it on a GAP-adjusted dataset, which adds the original labels to that of adversarial examples. Given that LGS [33] is an image analysis method, we set the block size to 15, overlap to 5, threshold to 0.17, smoothing factor to 2.3 and directly evaluate this method on our GAP dataset. For APE [16], we keep its original setting and evaluate it on our GAP dataset. Since SAC [26] and PatchZero [48] require an image segmentation dataset with pixel-level annotations, which is not aligned with our GAP dataset, we train the models using their original settings. For methods that generate masks, we convert the generated masks into bounding boxes to calculate the AP@0.5.

5.1.3 Detailed Experimental Settings

During training, we empirically set $\alpha = 0.4$ and $\beta = 10$. An SGD optimizer is used with an initial learning rate of 0.01, a momentum value of 0.937 and a weight decay of 0.0005. The batch size is 16, and the detector is trained for a maximum of 200 epochs. As for the inference stage, we empirically set $\gamma = 2$ and the standard deviation $\sigma = 3$ of a 3×3 Gaussian kernel. Additionally, the radius of the circular low-pass filter $\mathbf{R}_L$ is set as one-fourth of the image's width. All codes are implemented in Python 3.7 using PyTorch and all experiments are conducted on an NVIDIA GeForce RTX 2080Ti GPU cluster. [3]

5.2. Detection Performance on Adversarial Patches

In this section, we first evaluate the detection performance of our proposed method. To provide a detailed evaluation, we divide the testing set into 8 subsets according to the attacking methods. The visualization of detection results between our NAPGuard and compared baselines can be found

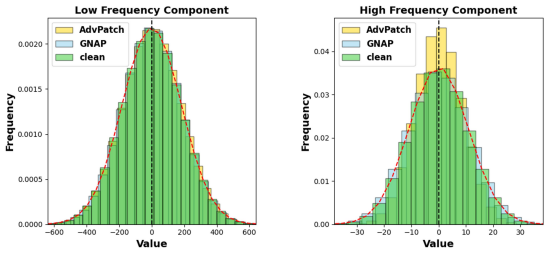


Figure 7. The distributions of frequency components among Non-NAEs (AdvPatch [38]), NAEs (GNAP [10]) and clean examples.

in Fig. 5. As illustrated in Tab. 1, we can draw several conclusions as follows:

(1) For NAPs, it can be clearly observed that our method significantly improves the performance compared with the base detector. For example, our method yields remarkable **38.10%** improvement on average for NAPs. Specifically, we improve the generalizable performance on DM-NAP and LAP by **44.18%** and **49.84%**, respectively.

(2) For Non-NAPs, our method also shows great improvement compared with the base detector (*i.e.*, **+32.30%** for AdvCloak and **+27.46%** for AdvTshirt), indicating the effectiveness and strong generalization capability of our method for Non-NAPs.

(3) Besides, we can witness that our method outperforms the compared baselines by large margins, achieving an AP@0.5 improvement of **60.24%** for NAPs on average. Further, we observe the limited precision of LGS and APE as it generates numerous masks in non-patch regions (as shown in Fig. 5), resulting in a low average AP@0.5 of only 4.95% for NAPs on average. SAC and PatchZero exhibit some success in detecting Non-NAPs (*i.e.*, 41.54% and 11.65% for AdvTexture, respectively), but struggle to generalize their performance to NAPs due to the deceptive appearance. Regarding Ad-YOLO trained on our GAP-adjusted dataset, though it successfully detects other categories (as shown in Fig. 5), its average AP@0.5 on NAPs only achieves 30.76%, whereas the base detector achieves 52.09%. We attribute this disparity to the incorporation of various categories during the learning process of Ad-YOLO, which hinders its ability to accurately capture aggressive patterns. In comparison, our method achieves the highest AP@0.5 of **91.00%** for NAPs on average. [3]

In summary, our NAPGuard achieves remarkable performance on both NAPs and Non-NAPs, *i.e.*, with an average AP@0.5 of **91.00%** for NAPs, **90.26%** for Non-NAPs, and **92.24%** for the mixture dataset, surpassing the compared baselines by large margins.

5.3. Discussion and Analysis

In this section, we evaluate our method on the GAP dataset, discuss our two strategies and propose an alternative feature shield module utilizing the high-frequency components.

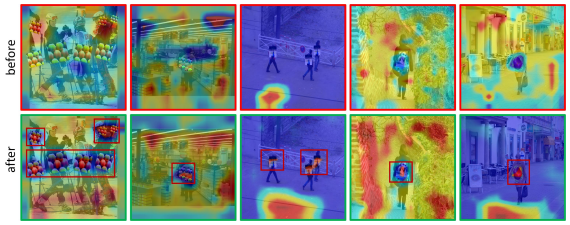


Figure 8. The model attention analysis. After using our AFAL strategy, the detector can recognize aggressive patterns better.

5.3.1 GAP Dataset Evaluation

To evaluate the generalization of our method, we conduct experiments on different GLs of GAP dataset. From Fig. 6, there is a significant decline in the performance of the detection methods, as the GL increasing, demonstrating the effectiveness of GAP dataset to benchmark generalization. Experiment results show that our method achieves a higher generalizable performance than the base detector (*i.e.*, **+21.21%** on GL2, **+61.28%** on GL3), which significantly outperforms other compared methods.

5.3.2 Training Strategy Analysis

First, we demonstrate that the aggressive features of adversarial patches primarily reside in high-frequency components. Fig. 7 shows the greater disparity in the distribution of high-frequency components compared to low-frequency components, supporting our viewpoint. Further, we observe that compared with non-naturalistic adversarial examples (Non-NAEs), NAEs exhibit similar high-frequency components to clean images, indicating that NAPs resemble to the surroundings more closely.

To verify the alignment, we provide visual evidence from the high-frequency domain. As shown in Fig. 9, we successfully enhance the similarity of high-frequency components between adversarial patches and the surroundings, which aligns with that of NAEs.

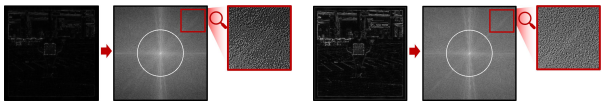
Further, we conduct experiments to analyze the model attention between the base detector and a detector trained using our AFAL strategy. By comparing the visual attention patterns of the two models using CAM [50], we can assess the impact of our AFAL strategy on model's capability to focus on patch regions. Fig. 8 visually demonstrates the focused attention of our enhanced detector on adversarial patches. In other words, our AFAL strategy successfully enhances the detector's capability to recognize aggressive patterns in deceptive patches.

5.3.3 Inference Strategy Analysis

In this part, we utilize t-SNE [40] to demonstrate the effectiveness of our NFSI strategy. Specifically, We visualize the features extracted by the base detector from an NAE (*e.g.*,

Table 2. The experimental results (AP@0.5↑) of using alternative $\mathcal{H}^*$ ("NFSI-A") on our GAP dataset.

Method	GL1	GL2	GL3	Mixture
Base	94.75	62.87	27.14	76.84
+NFSI-A	94.52	65.42	27.64	77.20
+AFAL & NFSI-A	**96.43**	**84.13**	**90.56**	**92.76**



(a) (b)

Figure 9. Visualization results of high-frequency components before and after AFAL.

GNAP [10]) before and after applying our NFSI strategy. To analyze the impact on natural features, we compare both of them with the features extracted from its corresponding clean image. As shown in Fig. 10, the features deviate further from those of the clean image after applying NFSI, indicating that our NFSI strategy effectively suppresses natural features within the image.

5.3.4 Alternative Feature Shield Module

In our NFSI strategy, we investigate the potential of utilizing high-frequency components for region selection from the perspective of aggressive features. To achieve this, we substitute the low-pass filter $\mathbf{R}_L(u,v)$ with a high-pass one $\mathbf{R}_H(u,v)$. Given the more dispersed distribution of high-frequency components, we adjust the threshold γ to a larger value (*e.g.*, 3) for precise region selection. To ensure the preservation of aggressive features, we replace the mask $\mathbf{M}$ with $(\mathbf{1}-\mathbf{M})$. We denote this alternative module as $\mathcal{H}^*$. This reformulation allows us to modify Eq. (6) as follows:

$$\mathbf{I}_s = \mathcal{H}^*(\mathbf{I}) = (\mathbf{1}-\mathbf{M}) \odot \mathcal{C}(\mathbf{I}, \mathbf{G}_\sigma) + \mathbf{M} \odot \mathbf{I}. \quad (7)$$

We conduct experiment on our GAP dataset to evaluate the effectiveness of $\mathcal{H}^*$. Results in Tab. 2 demonstrate its capability to enhance generalization (*i.e.*, +2.55% on GL2). Additionally, when combined with our AFAL strategy, we observe a further enhancement (*i.e.*, +62.92% on GL3), confirming that $\mathcal{H}^*$ serves as a viable alternative module.

5.4. Ablation Studies

In this section, we provide ablation studies to further investigate the contributions of different strategies. As shown in Tab. 3, the AP@0.5 shows a significant rise (*i.e.*, **+38.01%** for NAPs), indicating that our AFAL strategy can significantly improve the precision. Additionally, our NFSI strategy yields a modest improvement in AP@0.5 (*i.e.*, +0.56% for Non-NAPs and +1.14% for NAPs), demonstrating the enhanced generalization. These experimental results show

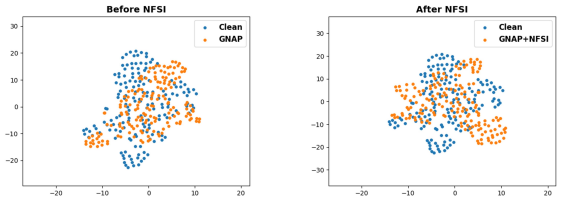


Figure 10. Visualizing the NAE's features extracted by the base detector before and after applying NFSI, alongside the features extracted from a clean image.

Table 3. The ablation study results (AP@0.5↑) on our training and inference strategies. "+AFAL" and "+NFSI" represent using the training and inference strategy alone.

Method			Base	+AFAL	+NFSI	Ours
Patch Type	Non-NAPs	T-SEA [12]	99.42	98.32	**99.45**	98.37
		AdvPatch [38]	87.83	96.89	89.46	**96.95**
		AdvCloak [45]	59.94	91.22	60.17	**92.24**
		AdvTshirt [47]	42.07	65.95	44.23	**69.53**
		AdvTexture [11]	69.51	93.71	68.30	**94.20**
	NAPs	GNAP [10]	68.00	88.00	68.19	**88.27**
		DM-NAP [20]	54.48	98.59	57.93	**98.66**
		LAP [36]	36.23	**86.13**	36.02	86.07
	Mixture	Mixture	76.84	91.89	77.19	**92.24**

that our two strategies contribute to the AP@0.5 individually, while combining them shows further improvement, highlighting the synergistic effect of these strategies. Besides, **we also conduct ablation studies on different loss terms and the choices of hyper-parameters.** [3]

6. Conclusion

In this paper, we propose the NAPGuard, a novel framework to effectively detect NAPs by directionally modulating both aggressive and natural features during training and inference, respectively. Further, we propose the first GAP dataset to prompt future research on benchmarking adversarial patch detection. Extensive experiments demonstrate that our method achieves state-of-the-art performance on NAPs, outperforming other present methods by large margins (*e.g.*, **60.24%** AP@0.5 on average).

Though achieving remarkable performance, there are still some limitations of this framework. In the future, we plan to apply this framework to various models and expand our GAP dataset, facilitating to build a comprehensive benchmark in this domain. Moreover, we are interested in exploring the detection capabilities of our framework in real-world scenarios to assess the robustness of our approach. Further, we would like to deploy our NAPGuard framework as a pre-processing step in existing defense methods, *e.g.*, image denoising.

Acknowledgement. This work is supported by grants No.KZ46009501.

[3] Please refer to the Supplementary Material for more details.

References

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

[2] Edward Chou, Florian Tramèr, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops, SP Workshops, San Francisco, CA, USA, May 21, 2020*, pages 48–54. IEEE, 2020. 1, 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1: 886–893 vol. 1, 2005. 5

[4] Bao Gia Doan, Minhui Xue, Shiqing Ma, Ehsan Abbasnejad, and Damith C Ranasinghe. Tnt attacks! universal naturalistic adversarial patches against deep neural network systems. *IEEE Transactions on Information Forensics and Security*, 17:3816–3830, 2022. 2

[5] Ranjie Duan, Xingjun Ma, Yisen Wang, J. Bailey, A. K. Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 997–1005, 2020. 2

[6] 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. 2, 3, 4

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[8] Jun Guo, Wei Bao, Jiakai Wang, Yuqing Ma, Xinghai Gao, Gang Xiao, Aishan Liu, Jian Dong, Xianglong Liu, and Wenjun Wu. A comprehensive evaluation framework for deep model robustness. *Pattern Recognition*, 2023. 2

[9] Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1567–1574, 2021. 1

[10] Yuqing Hu, Jun-Cheng Chen, Bo-Han Kung, K. Hua, and Daniel Stanley Tan. Naturalistic physical adversarial patch for object detectors. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7828–7837, 2021. 1, 2, 5, 6, 7, 8

[11] Zhan Hu, Siyuan Huang, Xiaopei Zhu, Xiaolin Hu, Fuchun Sun, and Bo Zhang. Adversarial texture for fooling person detectors in the physical world. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13297–13306, 2022. 1, 2, 5, 6, 8

[12] 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. 2, 5, 6, 8

[13] Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, A. Yuille, C. Zou, and Ning Liu. Universal physical camouflage attacks on object detectors. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 717–726, 2019. 2

[14] Nan Ji, YanFei Feng, Haidong Xie, Xueshuang Xiang, and Naijin Liu. Adversarial yolo: Defense human detection patch attacks via detecting adversarial patches. *ArXiv*, abs/2103.08860, 2021. 1, 2, 3, 6

[15] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, (Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, 2022. 6

[16] Taeheon Kim, Youngjoon Yu, and Yong Man Ro. Defending physical adversarial attack on object detection via adversarial patch-feature energy. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1905–1913, 2022. 1, 2, 6

[17] Mark Lee and J. Z. Kolter. On physical adversarial patches for object detection. *ArXiv*, abs/1906.11897, 2019. 2

[18] Simin Li, Huangxinxin Xu, Jiakai Wang, Aishan Liu, Fazhi He, Xianglong Liu, and Dacheng Tao. Hierarchical perceptual noise injection for social media fingerprint privacy protection. *arXiv preprint arXiv:2208.10688*, 2022. 1

[19] Simin Li, Shuning Zhang, Gujun Chen, Dong Wang, Pu Feng, Jiakai Wang, Aishan Liu, Xin Yi, and Xianglong Liu. Towards benchmarking and assessing visual naturalness of physical world adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12324–12333, 2023. 2

[20] 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, 2023. 2, 5, 6, 8

[21] Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, Deva Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. pages 740–755, 2014. 5

[22] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-sensitive gan for generating adversarial patches. In *33rd AAAI Conference on Artificial Intelligence*, 2019. 2

[23] Aishan Liu, Jiakai Wang, Xianglong Liu, Bowen Cao, Chongzhi Zhang, and Hang Yu. Bias-based universal adversarial patch attack for automatic check-out. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 395–410. Springer, 2020. 2

[24] Aishan Liu, Jun Guo, Jiakai Wang, Siyuan Liang, Renshuai Tao, Wenbo Zhou, Cong Liu, Xianglong Liu, and Dacheng Tao. X-adv: Physical adversarial object attacks against x-ray prohibited item detection. *arXiv preprint arXiv:2302.09491*, 1, 2023. 1

[25] Aishan Liu, Shiyu Tang, Xinyun Chen, Lei Huang, Haotong Qin, Xianglong Liu, and Dacheng Tao. Towards defending multiple lp-norm bounded adversarial perturbations via

gated batch normalization. *International Journal of Computer Vision*, 2023. 2

[26] Jiangjiang Liu, Alexander Levine, Chun Pong Lau, Ramalingam Chellappa, and S. Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962, 2021. 1, 2, 3, 6

[27] Shunchang Liu, Jiakai Wang, Aishan Liu, Yingwei Li, Yijie Gao, Xianglong Liu, and Dacheng Tao. Harnessing perceptual adversarial patches for crowd counting. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2055–2069, 2022. 1

[28] 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. 2

[29] Xianglong Liu, Shihao Bai, Shan An, Shuo Wang, Wei Liu, Xiaowei Zhao, and Yuqing Ma. A meaningful learning method for zero-shot semantic segmentation. *Science China Information Sciences*, 66(11):210103, 2023. 1

[30] Yuqing Ma, Xianglong Liu, Shihao Bai, Lei Wang, Dailan He, and Aishan Liu. Coarse-to-fine image inpainting via region-wise convolutions and non-local correlation. In *IJCAI*, pages 3123–3129, 2019.

[31] Yuqing Ma, Xianglong Liu, Shihao Bai, Lei Wang, Aishan Liu, Dacheng Tao, and Edwin R Hancock. Regionwise generative adversarial image inpainting for large missing areas. *IEEE Transactions on Cybernetics*, 2022. 1, 2

[32] J. H. Metzen and Maksym Yatsura. Efficient certified defenses against patch attacks on image classifiers. *ArXiv*, abs/2102.04154, 2021. 1, 2

[33] Muzammal Naseer, Salman Hameed Khan, and F. Porikli. Local gradients smoothing: Defense against localized adversarial attacks. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1300–1307, 2018. 1, 2, 6

[34] Sibo Song, Yueru Chen, Ngai-Man Cheung, and C-C Jay Kuo. Defense against adversarial attacks with saak transform. *arXiv preprint arXiv:1808.01785*, 2018. 2, 3, 4

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[36] 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*, pages 5307–5315, 2021. 2, 5, 6, 8

[37] 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*, pages 4087–4095, 2023. 1, 2

[38] Simen Thys, Wiebe Van Ranst, and T. Goedemé. Fooling automated surveillance cameras: Adversarial patches to attack person detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 49–55, 2019. 1, 2, 5, 6, 7, 8

[39] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980. 2, 3, 4

[40] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 7

[41] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8684–8694, 2020. 2, 3, 4

[42] Jiakai Wang. Adversarial examples in physical world. In *IJCAI*, pages 4925–4926, 2021. 1

[43] Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8565–8574, 2021. 1

[44] Yuxuan Wang, Jiakai Wang, Zixin Yin, Ruihao Gong, Jingyi Wang, Aishan Liu, and Xianglong Liu. Generating transferable adversarial examples against vision transformers. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5181–5190, 2022. 1

[45] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 1–17. Springer, 2020. 2, 5, 6, 8

[46] Chong Xiang and Prateek Mittal. Detectorguard: Provably securing object detectors against localized patch hiding attacks. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021. 1, 2

[47] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. pages 665–681, 2019. 1, 2, 5, 6, 8

[48] Ke Xu, Yao Xiao, Zhaoheng Zheng, Kaijie Cai, and Ram Nevatia. Patchzero: Defending against adversarial patch attacks by detecting and zeroing the patch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4632–4641, 2023. 1, 2, 3, 6

[49] Xiaowei Zhao, Xianglong Liu, Yuqing Ma, Shihao Bai, Yifan Shen, Zeyu Hao, and Aishan Liu. Temporal speciation network for few-shot object detection. *IEEE Transactions on Multimedia*, 2023. 1

[50] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 7