Adversarial Patch EXterminator: Zero-Shot and Patch-Agnostic Defense Framework Against Adversarial Patch Attacks

Jiayimei Wang[1] Tao Ni[2] Guowen Xu[3] Qingchuan Zhao[1]✉ Cong Wang[1]
[1]*City University of Hong Kong*
[2]*King Abdullah University of Science and Technology*
[3]*University of Electronic Science and Technology of China*

Abstract

Adversarial patch attacks pose a serious threat to modern computer vision systems. Although existing defense solutions attempt to mitigate such attacks by developing certifiable models or patch identification pipelines, they generally rely on prior knowledge or extensive training data, show insufficient robustness across varying physical conditions, and present limited performance against challenging cases (*e.g.*, tiny, irregular, or highly background-coherent patches). To address such limitations, we propose APEX, a zero-shot, patch-agnostic three-stage adversarial patch defense framework. Specifically, APEX first concentrates patch regions through bounding-box extraction, then integrates a mutual information-based blur heatmap with an edge-aware boundary heatmap to locate adversarial regions, and finally leverages structure-guided image inpainting to restore the image. Our experiments on multiple datasets and existing state-of-the-art defense methods demonstrate that APEX can effectively defend against various types of adversarial patches (*e.g.*, non-naturalistic, naturalistic, and infrared images). In addition, APEX shows superior capability in patch localization, maintains high robustness against varying environments (*e.g.*, lighting conditions) and extreme cases, and also demonstrates high performance in protecting various models in physical-world scenarios.

1 Introduction

Adversarial patch attacks have become one of the most ubiquitous and urgent threats to state-of-the-art computer vision systems that have been widely deployed in real-world platforms, such as critical surveillance cameras [18], autonomous driving [60], and medical diagnosis [54]. Specifically, these attacks are launched by embedding adversarial perturbations into small and purposely located patches, allowing the adversary to directly disrupt or mislead model predictions in both digital and physical environments [5]. In general, adversarial patches usually aim to hide the target object (*a.k.a.*, hiding attacks [9,53,62,76]) from model detection or result in the target object being misclassified as other objects (*a.k.a.*, altering attacks [10,23,47]). Hence, the practical execution pipeline of these patch attacks validates their threats in evading various vision-based detection systems in safety-critical domains.

While previous studies have made substantial efforts in mitigating the impact of adversarial patches by designing certifiably object detectors [68–70] to identify, localize and purify these patches [24,35,39,52,66], there are several limitations that hinder the effectiveness of existing approaches. In particular, some defense methods (*e.g.*, [68,69]) can only be effective in defending against hiding attacks while failing against patch-based altering attacks. Moreover, these studies show limited performance in localizing adversarial patches, leading to unsatisfactory performance in purifying patches and restoring the original image. Meanwhile, other studies (*i.e.*, [35, 52]) expose vulnerabilities that can be exploited by attackers to bypass the defenses through adversarial patches with irregular shapes and entropy distributions. Furthermore, several existing patch defenses (*e.g.*, [24,35,52]) demonstrate constrained settings in specific patch sizes and coherence with the background, *i.e.*, resulting in their failure in detecting tiny patches and patches that share similar color or texture patterns with the image background. In addition, current practices for improving defense robustness typically rely on models trained from newly generated adversarial samples, which show huge computational cost and impracticality in real-world scenarios. For example, defenses such as [35, 52, 66] could achieve enhanced robustness if retrained with sufficient samples of patches with irregular shapes. However, the model reliance and training costs of these defense methods prevent their quick iterations to defend against emerging adversarial patches in the physical world.

To advanced state-of-the-art defenses, we propose APEX (Adversarial Patch EXterminator), a zero-short, patch-agnostic defense framework against adversarial patch attacks. Unlike prior studies [11, 68, 69], APEX requires neither retraining models nor prior knowledge of potentially new adversarial patches but instead leverages intrinsic image features, such as entropy, texture, and gradients, to identify, localize, and purify adversarial patches. In particular, APEX is

(a) Illustration of three challenging adversarial patch cases

(b) Illustration of the defense results by APEX

Figure 1: Illustration of three challenging adversarial patch cases that have been successfully mitigated by APEX.

designed based on two inherent characteristics: *i)* adversarial patches inevitably disrupt the inherent consistency of images by introducing anomalies in statistical characteristics that can be captured through the analysis of visual attributes, *e.g.*, variations in color distribution, or texture discontinuities, even if they appear to be naturalistic; and *ii)* the limited robustness of existing defenses is largely due to their failure to deal with signal dilution, where anomalous features of patches (*e.g.*, irregular shapes and high background coherence) could be overwhelmed by global statistics, making these patches difficult to be identified and located. Therefore, by addressing these challenges, APEX could achieve a promising robustness in effectively defending against adversarial patch attacks without relying on costly model retraining.

Specifically, APEX consists of three key modules. First, it utilizes the bounding-box extraction module to employ low-threshold detection to preliminarily identify regions that potentially contain adversarial patches, which enables APEX to mitigate background interference and narrow the analysis scope to concentrate patch signals in key regions. Specifically, it prevents signal dilutions and enhance APEX's ability in detecting tiny, irregular, and background-coherent patches. Second, the patch localization module of APEX can generate blur heatmap from mutual information and local entropy, and boundary heatmap by extracting edge structural information from multi-scale gradient anomaly detection. These two heatmaps are then fused to produce a dual heatmap, which is subsequently integrated with the segmentation model to accurately locate the adversarial patch mask regions. Finally, as shown in Figure 1, APEX deploys an the image inpainting module is designed as leverage a plug-and-play structure-guided model to purify patches and restore the image, achieving both high detection performance and promising visual quality that is close to the original image.

To evaluate APEX's effectiveness, we conduct experiments using three public datasets [8, 12, 32], eight types of adversarial patch attacks (*i.e.*, non-naturalistic [21, 22, 53], naturalistic [20, 31, 51], and infrared [76, 77] patches) and five object detectors [3, 15, 45, 55, 56]. The results demonstrate that our method outperforms other studies [24, 35, 52] by achieving high recall with low false positives in patch localization and high precision in restoring images, and maintaining high detection rates across all three aforementioned challenging cases. In addition, we validate APEX's real-world applicability by testing on datasets captured in different environments that contain printed adversarial patches, where it successfully decreases the attack success rate, demonstrating its effectiveness in physical-world scenarios. Finally, our extensive evaluations on special cases (*e.g.*, extreme light conditions, abstract paintings, and rendered patches) demonstrates the resilience and generalization of APEX in defending against adversarial patches in complicated physical worlds.

Our contributions can be summarized as follows:

- We propose a zero-shot and patch-agnostic defense framework APEX that requires no prior knowledge and model retraining to counter adversarial patch attacks. It not only accurately localize various types of patches (*e.g.*, non-naturalistic, naturalistic, and infrared) but also reconstruct the image with high visual quality.
- The proposed APEX effectively overcomes limitations in prior defense methods and achieves high detection rates for adversarial patches with signal dilution and being difficult to be detected, such as tiny, irregularly shaped, or highly background-coherent patches.
- We conduct comprehensive experiments across multiple datasets and detectors to evaluate APEX, demonstrating its competitive performance in both patch localization and image inpainting, along with its applicability and resilience in varying conditions in real-world scenarios.

2 Background

2.1 Object Detection

Object detection is a fundamental computer vision task that involves localizing and classifying objects within a given image. Typically, there are two dominant object detection frameworks that exploit deep learning models: Faster R-CNN [15] and YOLO [3, 43, 45, 55, 56, 59]. Despite the two frameworks showing differences in architectures and computation pipelines, they ultimately predict bounding box candidate coordinates (x, y, w, h) with associated confidence scores $s \in [0, 1]$. Specifically, for an input image $I \in \mathbb{R}^{H \times W \times 3}$, the detector outputs a set of tuples $\{(b_i, s_i, c_i)\}_{i=1}^{N}$, where $b_i = (x_i, y_i, w_i, h_i)$ denotes the bounding box coordinates (*i.e.*, center position, width, and height) normalized to $[0, 1]$. Meanwhile, $s_i \in [0, 1]$ represents the detection confidence score, and $c_i \in \{1, ..., K\}$ indicates the predicted class label among K categories. The bounding box prediction is typically optimized using Intersection-over-Union (IoU) metrics or

its differentiable variants: $IoU(b_{pred}, b_{gt}) = \frac{b_{pred} \cap b_{gt}}{b_{pred} \cup b_{gt}}$. In practice, only predictions exceeding a confidence threshold τ (*i.e.*, empirically $\tau \in [0.5, 0.7]$) are retained as valid detections. This filtering process can be formalized as: $\mathcal{D}_{final} = \{(b_i, s_i, c_i) \mid s_i \geq \tau\}_{i=1}^{N}$, where $\mathcal{D}_{final}$ represents the final output after non-maximum suppression (NMS). The threshold τ controls the trade-off between the rates of precision and recall in the object detection system.

2.2 Adversarial Patch Attacks

Adversarial patch attacks represent a specific class of adversarial attacks targeting deep learning models, such as object detection models deployed in critical surveillance systems [48]. The core principles of adversarial patch attacks involves adding localized, highly visible digital perturbations (*i.e.*, patches) to the captured images. Unlike traditional adversarial attacks that apply subtle and global perturbations [17, 75], adversarial patch superimposes an optimized, limited-area disruptive pattern at arbitrary locations within an image in different scenarios. Moreover, these attacks remain effective even when the patch is physically printed as a sticker and placed on target objects in real-world environments [26, 36, 49]. The causality results from the abnormal sensitivity of deep learning models to localized strong interference patterns and the physical transferability inherent in the patch design. Typically, adversarial patch attacks aim to mislead the target detection models into failing to recognize objects covered by the patch (*a.k.a.*, hiding attacks [9, 53, 62, 76, 77]), or to induce them to misrecognize objects with incorrect classification decisions (*a.k.a.*, altering attacks [10, 23, 47]).

3 Threat Model

3.1 Attack Model

Adversary's Capabilities. The adversary aims to evade detection or mislead classification of the object detector by adding well-crafted localized and visible perturbations to the input image. Specifically, we assume that attackers have white-box access to the object detector, along with the code and data of our defense framework while fully understanding all algorithms and deployment details. Under this assumption, attackers are capable of launching defense-aware attacks by crafting adaptive adversarial patches customized to circumvent the proposed defense mechanism.

Attack Formulation. In the context of adversarial patch attacks for object detection, we consider the input image $\mathbf{x} \in \mathbb{R}^{W \times H \times C}$, where $W \times H$ denotes the spatial dimensions, and C denotes the number of channels. An adversarial patch is represented by a binary tensor, referred to as the patch mask $\mathbf{m} \in \{0,1\}^{W \times H}$, which identifies the affected region of the patch. Specifically, elements inside the patch mask are set to 0, while elements outside this region are set to 1. We define $\mathcal{M}$ as the set of all possible patch mask regions that the adversary can exploit, and the constraint set of the adversarial patch attack is represented as:

$$\mathcal{A}_{\mathcal{M}}(\mathbf{x}) = \left\{ \mathbf{x}' = \mathbf{x} \cdot \mathbf{m} + \mathbf{x}'' \cdot (1 - \mathbf{m}) \mid \mathbf{x} \in \mathcal{X}, \mathbf{m} \in \mathcal{M} \right\} \quad (1)$$

In this formulation, $\mathbf{x}'' \in [0,1]^{W \times H \times C}$ represents the content of the patch that the adversary can manipulate. These patches can have various shapes, such as squares, rectangles, circles, or even irregular forms (*e.g.*, "flower shapes"), and can have different sizes within reasonable constraints. In addition, following the consistent settings in previous adversarial patch attacks [20, 21, 51, 53, 76, 77], we assume that the patch is placed near or partially on the target object, and the patch should not be able to completely obscure the object.

3.2 Defense Model

Defender's Capabilities. The defender has no prior knowledge of the adversarial patches in the input image to achieve patch-agnostic detection, including the number of patches, the characteristics of the patches (*e.g.*, shapes, sizes, positions), and the targets of the patches (*e.g.*, hiding attacks or altering attacks). Next, the defender can detect all adversarial patches, localize their positions and generate masks, and then utilize a pre-trained inpainting model to reconstruct the original image.

Defense Formulation. In the defense framework, the primary goal is to identify and mitigate the effects of adversarial patches. Given an input image $\mathbf{x} \in \mathbb{R}^{W \times H \times C}$, our objective is to accurately localize all adversarial patches from a set of potential patch regions, denoted as $\mathcal{M}$, to generate a mask $\mathbf{m}'$ corresponding to the detected patch locations.

Specifically, the defense framework first attempts to identify the subset $\mathcal{M}' \subset \mathcal{M}$ that represents the regions in the image likely to contain adversarial patches, which can be formulated as follows:

$$\mathcal{M}' = \left\{ \mathbf{m}' \in \mathcal{M} \mid f(\mathbf{x}, \mathbf{m}') > \tau \right\} \quad (2)$$

where $f(\mathbf{x}, \mathbf{m}')$ is the scoring function to evaluate the probability that a region is adversarial, and τ is a pre-defined threshold.

Once $\mathcal{M}'$ is determined, masks corresponding to these regions are generated. These masks are then used to guide a pre-trained image inpainting model aimed at filling in the adversarial patch regions to restore the original image:

$$\mathbf{x}_{inpainted} = g(\mathbf{x}, \mathcal{M}') \quad (3)$$

Here, $g(\mathbf{x}, \mathcal{M}')$ represents the inpainting process that seeks to transform the image into a state that approximates the appearance of the original image from both a model detection perspective and a human visual inspection.

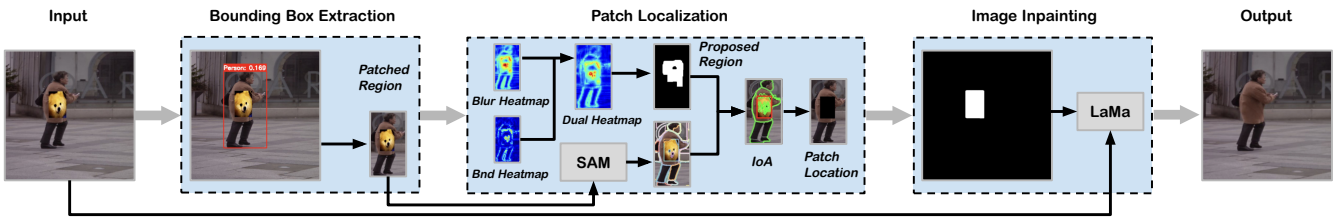


Figure 2: Overview of APEX

4 Design

4.1 Overview

As shown in Figure 2, the APEX framework consists of three steps to protect images affected by adversarial patch attacks. ❶ ***Bounding-box Extraction*** (§ 4.2): First, it extracts bounding boxes for potential objects within the patch-attached image to localize areas that are affected by adversarial patches. ❷ ***Patch Localization*** (§ 4.3): Second, APEX conducts image heat-mapping and feature analysis on each of these bounding boxes individually to accurately determine all positions of the adversarial patches, instead of analyzing the entire image. ❸ ***Image Inpainting*** (§ 4.4): Finally, APEX uses the identified patch locations to generate masks and applies these masks along with a pre-trained restoration model to effectively mitigate adversarial patches and reconstruct the integrity of the original image.

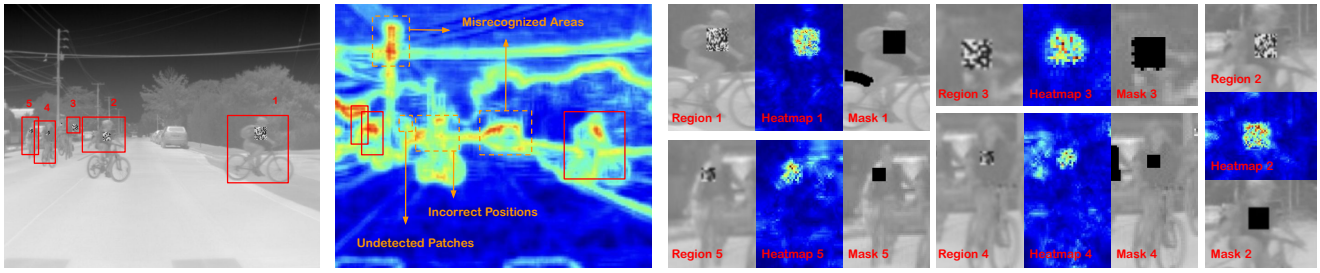
4.2 Bounding-box Extraction

Common Features of Adversarial Patches. To identify patches without any prior knowledge, APEX can only analyze the inherent features of the input image, including entropy, texture, edge characteristics, and heatmap distributions that exhibit significantly altered pixel diversity [6, 52], texture pattern details [21], shapes and boundaries [35], and visual attributes [24, 35, 39, 52] under adversarial patches, respectively. In addition, adversarial patches can induce substantial changes in gradient heatmaps because deep learning-based models usually assign higher weights and attention to patch regions, which affects the color distribution of hotspots in the generated heatmap [24]. As a result, these changes resulted from adversarial patches disrupt the images' visual and statistical characteristics, which can be utilized as common and important features for patch detection.

Dilemma of Existing Patch Detections. However, the effectiveness of existing image feature-based detection methods may be significantly compromised if any of the following three conditions occur: *i)* the adversarial patch occupies an extremely small proportion of the original image, *ii)* the patch boundaries are irregular, or *iii)* the patch exhibits high coherence with the background. This is due to signal dilution [16], where anomalous features are obscured by background pixels during global computation. For instance, when adversarial patches occupy an extremely small proportion of an image, their impact on overall visual features becomes negligible, increasing the difficulty to detect them. If the patch edges exhibit irregular shapes, they may blend with the complex edges of the image background, which can further reduce the precision of edge feature-based detection. Moreover, when the patch demonstrates high coherence with the background, which means its texture, color, or shape may closely resemble the background, and the information about texture differences become indistinct. In addition, the varying backgrounds can obscure the patch's influence on gradient heatmaps, which affects the identification of the patch region. Hence, these special scenarios substantially undermine the effectiveness of existing patch detection methods that rely solely on image feature analysis, as the patches only induce subtle or ambiguous visual feature changes.

Our Bounding-box Extraction Solution. To address the above limitations, we leverage small object detection [28, 37, 58] by effectively dividing the input image into smaller sub-images to enhance detection performance [30, 42, 72]. That is because applying image tiling methods can potentially increase the relative size of adversarial patches in divided sub-images, which can amplify their edge and text features while reducing interference from irrelevant background. However, we cannot directly tile images because it may inadvertently fragment patch features and consume extra computational resources by analyzing irrelevant areas, as most sub-images would likely contain no patches.

Therefore, we design a *bounding-box extraction* module to improve the tiling strategy for adversarial patch localization inspired by [11] which utilizes low-threshold bounding boxes with strategically placed defensive patches. Specifically, we propose to extract low-threshold bounding boxes for protected object categories and use them as tiled batches for subsequent patch localization. It is reasonable because *i)* existing adversarial patch attacks [20, 21, 51, 53, 77] normally reduce the confidence score of target bounding boxes below the typical threshold τ (*e.g.*, 0.5) to evade model detection, whereas they cannot reduce the confidence score to an extremely low-level threshold τ_{low} (*e.g.*, 0.1 or lower), and *ii)* adversarial patches are mostly placed inside or near the detection boxes of target objects (*e.g.*, people). In practice, in the input images, we first extract all extremely low-threshold bounding boxes of pro-



(a) Original adversarial example and corresponding heatmap.

(b) Images, heatmaps and patch masks of extracted bounding boxes.

Figure 3: Illustration before and after applying the bounding box extraction module. (a) demonstrates that prior to bounding box extraction, the generated heatmap was significantly affected by background interference, failing to accurately locate regions with high patch localization probability while completely ignoring tiny patches. (b) illustrates that all patches are identifiable from the heatmap because the bounding box extraction mitigate background interference and signal dilution effect.

tected objects as potential patch regions. Then we filter and merge overlapping bounding boxes, prioritizing larger boxes to ensure patch integrity after cropping, followed by selecting boxes with higher confidence scores. Subsequently, the positional information of the filtered bounding boxes in the original image is preserved to facilitate image reconstruction, and the cropped images are then fed into the patch localization module for analysis. Figure 3 illustrates that bounding box extraction significantly improves heatmap accuracy by reducing background interference and enabling precise localization of all adversarial patches, including those tiny patches, which were previously ignored or inaccurately detected.

4.3 Patch Localization

Localization Workflow. We design a two-stage module in APEX to localize adversarial patches in input images. *i) Heatmap Generation:* Initially, we extract two types of statistical anomaly information from the input image to generate heatmaps as references to determine the probability of patches, including the Gaussian blur-based heatmap [14] that depicts anomaly areas remaining prominent even after blurring, and the boundary heatmap derived from multi-scale gradient second-order variations, which highlights abrupt edge structure changes indicative of anomalies. *ii) Patch Mask Generation:* Then, the generated heatmaps are fused to create a dual heatmap and processed through thresholding and morphological operations to produce a mask representing the suspicious region. Subsequently, we apply the *Segment Anything Model* (SAM) [27] to generate candidate segments throughout the image, and candidates with sufficient spatial overlap with the dual heatmap are directly identified as patch regions, which can provide precise segmentation boundaries for the detected patch areas.

Heatmap Generation. In practice, the high-frequency contrast of many natural textures is weakened in images captured in real-world scenarios under low-frequency or blurred conditions. On the contrary, artificial patches can maintain relatively abrupt color blocks, edges, or contrast anomalies even after being blurred. Hence, in APEX, we use a blurred image $\tilde{I}$ as input to run a unified heatmap generator to obtain a reference heatmap of regions that remain noticeable under a blurred perspective. First, we perform a blur operation as

$$\tilde{I} = GaussianBlur(I,(k,k),\sigma) \tag{4}$$

where I represents the original image, σ is the standard deviation of the kernel, and the kernel size k is automatically corrected to be an odd number to ensure the legality and stability of the Gaussian kernel.

In the blurred image $\tilde{I}$, the low-frequency structural information of the scene is preserved while the high-frequency details of the natural textures are weakened. Under these conditions, the statistical dependencies between local areas and their neighborhoods in natural images usually remain stable. To characterize the degree of statistical dependence between two random variables, we use *Mutual Information* (MI) to measure not only linear correlations, but also variations in non-linear statistical relationships [29, 46, 57]. When the captured images contain adversarial patches, the pixel distribution of the patched area often differs significantly from the surrounding environment, which introduce entirely new color or texture distributions that can disrupt the stable joint distribution in adjacent blocks. As a consequence, the mutual information between the patch and its surroundings decreases drastically. By calculating the mutual information between the local blocks and their neighboring blocks in the blurred image, APEX can detect the decreases of statistical dependency in the patched image and then identify the localization of potential adversarial patch areas.

Specifically, MI computation is performed on two vectorized grayscale patches $P,Q \in \{0,\ldots,255\}^N$ and $N = w^2$. A co-occurrence matrix C is constructed as

$$C[u,v] = \#\{t \mid P[t] = u, Q[t] = v\}. \tag{5}$$

The joint probability distribution is given by

$$p(u,v) = \frac{C[u,v]}{N}, \quad p(u) = \sum_v p(u,v), \quad p(v) = \sum_u p(u,v). \tag{6}$$

The mutual information is then defined as

$$MI(P,Q) = \sum_{u=0}^{255}\sum_{v=0}^{255} p(u,v)\log\frac{p(u,v)}{p(u)p(v)}. \tag{7}$$

In the implementation, the co-occurrence matrix C is efficiently constructed in batches using sparse tensors. In practice, MI is computed for each pair of patches with a fixed relative displacement, the bidirectional results are merged, and the final pixel-wise MI map E is obtained by averaging the number of valid neighbors. In particular, the discriminative power of MI varies with the local information content. For instance, in low-entropy regions (*e.g.*, flat backgrounds), the limited intensity variation can lead to low MI values even in the absence of anomalies. This can cause false positives in simple regions and undetected cases in complex regions, as the perceptual threshold should ideally be proportional to the background stimulus strength. To address this issue, we compute the local entropy $H_{local}(x,y)$ on the original grayscale image using a 9×9 window and 16 histogram bins:

$$H_{local}(x,y) = -\sum_{k=1}^{16} p_k(x,y)\log_2 p_k(x,y), \tag{8}$$

where $p_k(x,y)$ is the normalized histogram count in bin k for the window centered at (x,y). Next, a min-max normalization is then applied:

$$\tilde{H}_{local}(x,y) = \frac{H_{local}(x,y) - H_{\min}}{H_{\max} - H_{\min} + 10^{-8}} \in [0,1]. \tag{9}$$

By using low-entropy regions as sensitive detectors and high-entropy regions as enhanced detectors, we achieve spatially adaptive anomaly detection sensitivity. Finally, the MI map $E(x,y)$ is weighted by the normalized entropy to produce the heatmap with $\beta = 0.5$ to achieve a balanced performance in our implementations:

$$Heatmap(x,y) = E(x,y)\left[1+\beta\tilde{H}_{local}(x,y)\right]. \tag{10}$$

To achieve adaptive adjustment of detection sensitivity, we can estimate the background complexity from local entropy, which solves the fundamental problem of uneven performance of fixed thresholds in heterogeneous texture environments.

Although the blur heatmap can capture abrupt changes in texture statistics and color distribution within patches, it fundamentally relies more on statistical anomalies within regions. In particular, if the adversarial patches exhibit similar colors or textures in the background but with irregular shapes or sharp boundaries, the response from such methods can be affected. Therefore, we introduce a boundary heatmap, which specifically characterizes structural anomalies along boundaries by analyzing second-order variations in gradient direction and magnitude to detect abrupt changes in geometric patterns and edge features This complementary perspective addresses the limitations of blur heatmap in detecting edge-dominant patches, leading to a synergistic enhancement to improve the detection accuracy for diverse adversarial patches.

Specifically, the boundary heatmap is constructed using a multi-scale gradient and second-order variation analysis. The input image I is first downsampled at multiple scales $s \in \{1,2,4\}$, and for each scale $I^{(s)}$, the horizontal gradient and vertical gradient are computed using the Sobel operator $\partial_x I^{(s)}$ and $\partial_y I^{(s)}$. From these, the gradient magnitude and gradient direction are derived as:

$$G^{(s)} = \sqrt{\left(\partial_x I^{(s)}\right)^2 + \left(\partial_y I^{(s)}\right)^2}, \tag{11}$$

$$\Theta^{(s)} = \arctan 2\left(\partial_y I^{(s)}, \partial_x I^{(s)}\right). \tag{12}$$

The Laplacian operator Δ is then applied to both $\Theta^{(s)}$ and $G^{(s)}$ to measure second-order variations, yielding the direction variation map $A_\theta^{(s)} = \left|\Delta\Theta^{(s)}\right|$ and the magnitude variation map $A_g^{(s)} = \left|\Delta G^{(s)}\right|$. These two maps are fused with hyperparameters α_θ and α_g: $B^{(s)} = \alpha_\theta A_\theta^{(s)} + \alpha_g A_g^{(s)}$. Finally, all $B^{(s)}$ are upsampled to the original resolution and normalized as:

$$H_{bnd} = Norm\left(\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} Up\left(B^{(s)}\right)\right), \tag{13}$$

which highlights edges exhibiting directional discontinuities or abrupt magnitude changes across scales.

Patch Mask Generation. After generating two heatmaps, we combine them to develop a dual heatmap, and then apply morphological processing and binarization to guide patch region selection. Specifically, we adopt linear fusion to integrate the abnormal signals reflected in the two heatmaps, which provides stable and reliable performance while mitigating redundant overheads that could impact detection performance. Next, we exploit SAM [27] or any other segmentation-capable model to segment the image and obtain the mask for each region. Each mask is then matched with the fused heatmap, and all masks whose intersection area exceeds the threshold θ_{cov} are categorized as the patch masks. Specifically, we justify that the threshold-based heatmap generation aligns with settings in prior adversarial patch defenses [24, 52] in the physical world. Because it can effectively identify spatial activations induced by patches in the dual heatmap and obtain naturally separated IoU distributions across varying physical environments. Note that this mechanism's effectiveness stems from dual heatmaps based on image feature information, whereas SAM [27] does not have the ability to identify patches, which is merely used to assist the heatmap in segmenting reasonable patch regions.

Specifically, given the blur heatmap H_{blur} and boundary heatmap H_{bnd}, a dual-heatmap is obtained via a weighted linear combination $H_{dual} = \alpha_{blur}H_{blur} + \alpha_{bnd}H_{bnd}$, where α_{blur} and α_{bnd} are tunable hyperparameters controlling the relative contributions of each source. Next, a percentile-based threshold $\tau = Percentile(H_{dual}, p)$ is applied to produce a pre-

(a) Input (b) Original mask $\mathcal{M}^{'}$ (c) Expanded mask $\mathcal{M}_e^{'}$

Figure 4: Inpainting results with different mask. (a) Input patches image; (b) Original mask $\mathcal{M}^{'}$; (c) Expanded mask $\mathcal{M}_e^{'}$ (1% dilation). The three images in (b) and (c) separately show the binary patch mask, image with overlaying patch mask, and the reconstructed image after inpainting.

liminary binary mask:

$$B(x) = \begin{cases} 1, & \text{if } H_{dual}(x) > \tau \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

which is further refined via morphological connected-component filtering [24] to remove small and isolated regions while preserving the detected location of the patch.

Subsequently, the SAM [27] generates a set of candidate masks $\{\mathcal{M}_k \in \{0,1\}^{H\times W}\}$. For each candidate, a coverage ratio can be calculated as:

$$r_k = \frac{\sum_x (\mathcal{M}_k(x) \wedge B(x))}{\sum_x \mathcal{M}_k(x) + \varepsilon}. \tag{15}$$

r_k is actually a variant of *Intersection over Area* (IoA), where the denominator is the area of the candidate mask $\mathcal{M}_k$, which measures the proportion of the candidate supported by the preliminary binary mask B. Meanwhile, candidates are retained if $r_k > \theta_{cov}$, and further filtered with the following method:

$$\tilde{r}_k = \frac{\sum_x (\mathcal{M}_k(x) \wedge R(x))}{\sum_x \mathcal{M}_k(x) + \varepsilon} \tag{16}$$

where $\tilde{r}_k$ quantifies the overlap between the k-th candidate mask and the set of already accepted masks R, so that candidates with excessive overlap can be discarded. Therefore, the final patch mask set is selected and denoted as $\mathcal{M}^{'} = R$.

4.4 Image Inpainting

There are three types of image inpainting methods to complete missing or occluded areas in images: *i)* traditional interpolation and propagation methods, such as pixel propagation based on the Navier-Stokes partial differential equations [2] and PatchMatch [1] to infer missing content through adjacent pixel matching or similar block matching, *ii)* deep learning-based generative models, including GAN-based [34, 61] and diffusion-based [7, 38] approaches, which utilize large-scale data to learn content and texture generation capabilities for semantic-level, high quality reconstructions, and *iii)* multi-scale and structure-guided methods like EdgeConnect [40] and LaMa [50], which guide the generation of missing content by extracting structural edges and frequency domain features from the scene, yielding better results in terms of structural consistency and global coherence.

When we have obtained masks of the missing patch, the inpainting task becomes more constrained, with the main challenge focusing on the seamless integration of structure and texture, rather than generating open-domain content from scratch. In this case, utilizing GAN-based or diffusion-based deep generative models may increase unnecessary semantic hallucination risk, resulting in additional inference overhead, and training complexity. That is, the generation process of these models requires multiple iterations that cause higher inference latency and resource consumption. In contrast, multi-scale and structure-based models like LaMa [50], which exploit large receptive field convolution and frequency-aware mechanisms, can more directly and efficiently leverage existing context and mask information for restoration while ensuring structural continuity and edge alignment with fast inference suitable for engineering deployment.

Therefore, we ultimately selected a plug-and-play, structure-guided LaMa [50] image restoration method. In practice, we directly deploy the pre-trained image inpainting module after the patch localization module. For the input image I, the localization module first outputs the corresponding patch set $\mathcal{M}^{'}$, which is then fed into this structure-guided inpainting module for prediction. To guarantee the image inpainting model can effectively utilize contextual information over a wider area and generate content that seamlessly integrates with the original image, we enlarge the original patch's mask region $\mathcal{M}^{'}$ to $\mathcal{M}_e^{'}$. It is reasonable to apply the mask expansion module to cover both the patch region and the boundaries, which enables the inpainting model to produce smooth and coherent restorations. As shown in Figure 4, the moderately expanded mask demonstrates improved coherence in restored regions, which generates better visual completion with LaMa [50] inpainting. Specifically, widening the repair region $\mathcal{M}_e^{'}$ alleviates the issues caused by a lack of context in narrower areas, ensuring that the final output image I^* maintains consistency with the style and structure of the original image.

5 Evaluation

5.1 Experiment Setup

Datasets and Target Object Detectors. In our experiments, we construct adversarial examples using three public datasets for digital-domain evaluations and one private dataset that we collected for physical domain evaluations. Specifically, the digital-domain datasets include two RGB-based datasets (*e.g.*, *MS COCO* [32] and *INRIA_Person* [8]) and one infrared image dataset (*e.g.*, *FLIR_ADAS_v1_3* [12]) to evaluate `APEX`'s effectiveness in NIR-based camera surveillance, and we perform additional filtering on these datasets to select

Table 1: Preliminary results to determine the low threshold τ_{low}. Ground Truth (GT):The annotated number of people in the dataset, True Positives (TP): The number of correctly detected people. Recall (%): The detection success rate for human objects.

Dataset	GT	$\tau_{low} = 0.025$		$\tau_{low} = 0.05$		$\tau_{low} = 0.075$		$\tau_{low} = 0.1$		$\tau_{low} = 0.125$		$\tau_{low} = 0.15$	
		TP	Recall	TP	Recall	TP	Recall	TP	Recall	TP	Recall	TP	Recall
INRIA_Person [8]	1837	1817	98.91	1810	98.53	1807	98.36	1779	**96.84**	1724	93.85	1619	89.10
MS COCO [32]	8939	8366	93.59	8049	90.04	7553	84.49	7230	**80.88**	6794	75.51	6075	67.96
FLIR [12]	13094	13011	99.37	12905	98.56	12168	92.93	11643	**88.92**	11075	84.58	10264	78.39

images containing target subjects (*e.g.*, people) along with their corresponding labels. Furthermore, we collect the physical-domain dataset by generating printable adversarial patches and recording multiple handheld videos of these patches using a DJI Pocket 3 camera at 60 fps in various resolutions, and then extracting 20 video frames per second. While we investigate the effectiveness of infrared adversarial patches in the digital domain, we exclude infrared adversarial patches from the physical dataset due to the additional hardware requirements for implementing thermal infrared patches in the physical environment. In practice, for target object detection models, we use Faster R-CNN [15], YOLOv3 [45], YOLOv4 [3], YOLOv5s [55] and YOLOv8n [56] with the officially provided pre-trained weights, and we fine-tuned these models on *FLIR* dataset for infrared image evaluations.

Tested Adversarial Patches. To evaluate APEX's effectiveness against adversarial patches with different characteristics (*e.g.*, color spaces, styles, and shapes), we evaluated three categories of adversarial patches proposed in eight previous studies, including non-naturalistic patches with complicated texture patterns without natural semantics (*e.g.*, T-SEA [22], AdvPatch [53], and TC-EGA [21]), naturalistic patches (*e.g.*, GNAP [20], DM-NAP [31], and LAP [51]), and infrared adversarial patches (*e.g.*, Bulb [77] and QR Code [76]). Note that because these studies define human subjects as the attack objectives, we follow the research line by setting human objects as the default defense objectives in our evaluations.

Comparisons with Other Defense Methods. We choose five state-of-the-art adversarial patch defense methods for qualitative and quantitative comparisons, including *i) Local Gradient Smoothing* (LGS) [39], which is a local gradient-based smoothing method that leverages gradient features for noise suppression; *ii) Segment and Complete* (SAC) [35], which exploits shape-prior learning to segment adversarial patches; *iii) Jedi* [52], which identifies patch regions using entropy-anomaly-aware detection and then masks these regions with black blocks for image inpainting; *iv) ObjectSeeker* [69], which utilizes dynamic masking statistics to provide certified defense; and *v) PAD* [24], which adopts semantic-spatial dual-heatmap for adversarial patch localization.

Parameter Setting. In practice, to select a reasonable low threshold τ_{low} mentioned in § 4.2, we conduct preliminary tests [11] to determine the optimal threshold to avoid additional processing time when $\tau_{low} < 0.025$ and overlooked small patches when $\tau_{low} > 0.15$. Since most patches in the evalutaed studies are trained on YOLO-based detectors, we adopt the recall rates of YOLOv3's low-threshold detection as the reference. As shown in Table 1, empirical results demonstrate the recall rates of APEX exceed 80% across all evaluated datasets when $\tau_{low} = 0.1$. Therefore, we set $\tau_{low} = 0.1$ for the cropping of the bounding box with considerations of the trade-off between recall and efficiency. Additionally, we apply a hierarchical hyperparameter optimization strategy to identify optimal configurations. We consider four parameters affect detection performance: *i)* $\tau = Percentile(H_{dual}, p)$, the anomaly threshold percentile; *ii)* θ_{cov}, the IoU threshold between SAM masks and dual heatmaps; *iii)* α_{blur} and $\alpha_{bnd} = 1 - \alpha_{blur}$, the fusion ratio between blur and boundary heatmaps; and *iv)* blur kernel size k, controlling Gaussian smoothing. Hence, we conduct a grid search over 625 combinations ($5 \times 5 \times 5 \times 5$) shows that the optimal configuration achieves an F1 score of 0.885 and a recall of 96.73%, with $\tau = 85$, $\theta_{cov} = 0.7$, $\alpha_{blur} = 0.5$, and $k = 15$, which justify the parameter settings we adopted in APEX. More empirical results about parameter sensitivity analysis are discussed in Appendix A.

Implement Details. In practice, we use the *SAM ViT-Large* [27] model for image segmentation and the *Big LaMa* [50] model for image inpainting, enabling the refinement to reconstruct high-quality inpainting outputs. Note that we discarded bounding boxes smaller than 16×16 in the design and implementation because such tiny boxes cause inefficiency with unnecessary latency. For the hardware platforms, we conducted our main evaluations on a single NVIDIA A6000 GPU, which provides 48GB VRAM. In the efficiency analysis, we further extend our experiments to configurations with up to eight NVIDIA A6000 GPUs to evaluate the scalability and potential efficiency enhancement.

5.2 Effectiveness

Overall Defense Performance. In object detection tasks, the mean Average Precision (mAP) serves as a widely adopted evaluation metric, which provides a comprehensive measurement of a model's performance in both localizing and recognizing objects of specific categories. Specifically, mAP is calculated by first determining the Average Precision (AP) for each individual class and then calculating the mean value across all classes, making it a effective evaluation metric to demonstrate the effectiveness of defense mechanisms.

As shown in Table 2, our defense, APEX, achieves state-of-the-art robustness against different adversarial patch attacks across multiple object detectors. Compared to existing de-

Table 2: mAP (%) of object detection models under different adversarial attacks with various defense methods. The **red** numbers denote the highest performance, and Non-NAPs: Non-naturalistic patches, NAPs: Naturalistic patches, IRPs: Infrared patches.

Detector	Defense Method	Non-NAPs			NAPs			IRPs	
		T-SEA	AdvPatch	TC-EGA	GNAP	DM-NAP	LAP	Bulb	QR Code
Faster R-CNN	w/o defense	32.17	43.82	48.92	28.19	25.16	18.94	35.71	19.91
	LGS	45.96	60.87	76.82	54.61	57.89	43.72	63.91	53.43
	SAC	75.69	84.15	**81.94**	49.57	60.32	48.89	64.79	56.71
	Jedi	56.29	73.81	62.84	61.29	72.21	56.27	47.87	39.79
	ObjectSeeker	49.59	66.37	52.41	53.28	62.53	39.47	45.78	32.86
	PAD	78.31	85.49	80.21	60.93	68.74	57.49	51.72	43.29
	APEX (Ours)	**79.84**	**88.46**	80.68	**71.35**	**77.24**	**65.42**	**68.32**	**63.11**
YOLOv3	w/o defense	24.29	38.51	41.37	22.52	23.29	15.67	31.44	18.43
	LGS	40.22	48.92	60.78	41.64	47.69	39.82	52.71	43.23
	SAC	71.43	76.28	78.26	45.63	58.29	47.75	59.48	47.89
	Jedi	47.46	74.92	60.29	54.21	70.34	51.17	45.32	38.86
	ObjectSeeker	45.66	71.72	55.32	46.78	60.46	32.48	41.86	29.97
	PAD	81.75	88.23	85.29	65.38	74.42	60.23	53.41	48.97
	APEX (Ours)	**82.28**	**90.23**	**86.62**	**78.62**	**81.27**	**76.84**	**72.43**	**61.52**
YOLOv4	w/o defense	30.12	41.39	45.33	26.22	23.47	16.94	32.39	23.57
	LGS	49.72	65.54	70.38	50.38	51.54	44.39	61.32	48.19
	SAC	68.12	82.23	81.19	53.52	65.45	45.09	57.13	51.24
	Jedi	56.95	71.17	57.28	50.46	67.32	48.61	42.05	35.92
	ObjectSeeker	48.12	69.25	52.55	50.62	57.44	34.97	39.77	32.18
	PAD	**78.08**	84.47	81.02	68.65	70.70	54.22	50.74	44.72
	APEX (Ours)	76.57	**86.02**	**83.15**	**73.21**	**74.29**	**69.53**	**65.79**	**58.46**
YOLOv5	w/o defense	26.47	40.42	43.59	20.41	21.55	14.81	29.85	24.89
	LGS	48.91	62.29	68.54	54.34	55.73	45.18	55.34	49.26
	SAC	72.91	80.47	76.20	54.53	63.34	47.11	60.13	53.55
	Jedi	53.86	69.33	65.11	58.54	65.37	54.96	49.85	32.43
	ObjectSeeker	42.11	72.58	58.09	48.44	62.87	33.50	45.62	34.25
	PAD	76.39	82.94	79.37	62.92	**77.85**	52.64	48.07	40.01
	APEX (Ours)	**77.19**	**86.42**	**82.29**	**75.47**	75.93	**72.23**	**64.31**	**58.97**
YOLOv8	w/o defense	28.12	39.12	45.52	27.37	29.32	22.52	38.85	29.43
	LGS	46.38	67.81	72.94	52.34	60.66	46.28	57.89	52.31
	SAC	75.04	77.42	75.22	47.61	59.32	52.84	62.49	52.38
	Jedi	51.26	72.58	60.48	62.34	63.31	53.98	41.79	30.21
	ObjectSeeker	50.23	70.31	60.85	49.12	56.23	32.01	43.96	29.80
	PAD	80.12	83.17	82.34	62.76	71.44	58.42	51.27	42.86
	APEX (Ours)	**80.63**	**87.02**	**84.89**	**72.19**	**73.65**	**74.36**	**66.64**	**61.37**

Table 3: Patch localization recall (%).

Patch Type	Attack Method	SAC	Jedi	PAD	APEX (Ours)
Non-NAPs	T-SEA	28.47	39.54	60.39	**78.63**
	AdvPatch	37.78	48.18	68.94	**82.42**
	TC-EGA	26.54	37.99	65.37	**74.55**
NAPs	GNAP	9.57	8.45	48.24	**71.31**
	DM-NAP	6.32	4.91	47.73	**67.84**
	LAP	1.42	5.68	39.98	**62.33**
IRPs	Bulb	19.54	27.44	50.56	**70.65**
	QR Code	15.79	19.28	45.37	**66.38**

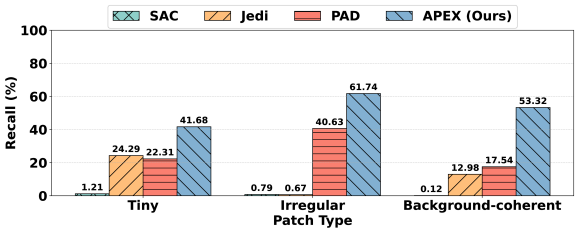


Figure 5: Patch localization recall of three challenging cases, including patches occupy a tiny area of the image, has an irregular shape, or exhibit high coherence with the background.

fense approaches such as LGS, SAC, Jedi, ObjectSeeker, and PAD, our method demonstrates superior performance against various adversarial patches. For example, APEX improves the mAP from 47.89% to 61.52% in defending against QR code-based patch attacks on the YOLOv3 detection model. In particular, APEX not only shows promising performance in single object detector (*e.g.*, YOLOv4, YOLOv5, and YOLOv8), but also demonstrates high robustness in two-stage object detectors such as Faster R-CNN. Therefore, these empirical results validate the effectiveness and generalizability of our defense method in countering various adversarial patch threats.

Patch Localization Performance. To measure the effectiveness in detection, it is significant to examine whether a defense can accurately identify the spatial location of the adversarial patches and then trigger the following purification steps. Specifically, we utilize recall rates for localization performance evaluations, which represents the fraction of predicted patch masks whose IoU with ground-truth patch exceeds the threshold 0.5. Table 3 reports the patch localization recall for different types of adversarial patches and attack methods under four defenses. Note that we compare APEX with SAC, Jedi, and PAD because only these three defenses generate explicit patch masks. Our method consistently achieves the highest recall in all cases, demonstrating strong and reliable localization ability across different patch types. For example, under the AdvPatch attack, our method achieves 82.42% recall, compared with 37.78% for SAC, 48.18% for Jedi, and 68.94% for PAD. In naturalistic patches generated by LAP, our method reaches 62.33% recall, while SAC and Jedi drop dramatically to 1.42% and 5.68%, which demonstrates the severe limitations of previous defenses on naturalistic patches. By contrast, our proposed APEX mitigates these limitations and maintains

robust localization capability across both naturalistic and non-naturalistic patches, as well as infrared patches.

Furthermore, we design supplementary experiments to evaluate APEX's detection recall under three challenging scenarios: *i)* the adversarial patch occupies a tiny area of the image, *ii)* the adversarial patch has an irregular shape, and *iii)* the adversarial patch demonstrates high coherence with the background. Specifically, we list the details of construction criteria for the testing sets of the above three challenging cases in Appendix E. The experimental results in Figure 5 demonstrate that APEX substantially outperforms other methods in detecting and localizing tiny, irregular, and background-coherent patches while methods such as SAC completely fail in these cases.

False Positive Analysis. In patch detection tasks, the false positive rate measures the occurrence of erroneous alerts generated by the detection algorithm in non-patch regions. If masking is also applied to non-patch regions, this may further impact the object detection model's performance. In our experimental setup, we define a false positive for an image containing multiple patches as occurring when the model's misclassified pixel blocks outside any genuine patch region exceed a predefined threshold relative to the entire image area. Specifically, we deploy three threshold levels of non-patch pixels, *i.e.*, 1%, 3%, and 5%, based on the strictness of the criteria. As shown in Figure 6, our method shows an increased false positive rate at the 1% threshold, and it degrades drastically when we deploy loose thresholds (*e.g.*, 3%–5%). This is because the strict detection criteria we adopted in patch localization, which shows a balanced trade-off between acceptable false positive rates and competitive recall rates compared with prior methods (*e.g.*, SAC and Jedi).

Furthermore, as illustrated in Figure 7, SAC and Jedi show substantial deviations in patch localizations in non-naturalistic patches (Non-NAPs), while SAC presents large-scale missed detections and Jedi produces numerous false positives. In contrast, only PAD and our proposed APEX defenses achieve accurate patch localizations, whereas PAD still generates notable false positives at the bottom of the image frame. For naturalistic patch (NAP) and infrared patch (IRP) attacks, both SAC and Jedi completely fail in patch localization, and PAD cannot detect some NAP cases while generates extensive false positives in IRP cases. In comparison, our method achieves precise detection and localization even for NAPs with irregular shapes and extremely small IRPs, which shows only minimal false positives. These results demonstrate that our proposed patch localization module effectively balances the recall and false positive rates to realize the best overall performance. In addition, we also utilize image quality metrics to quantitatively assess the quality of images restored by APEX after patch localization, and to compare the results with those of other approaches, which shows that APEX significantly outperforms other methods in terms of inpainting results, and more details are provided in Appendix F.

Table 4: ASRs (%) of adversarial patch attacks in physical environment under different defense methods.

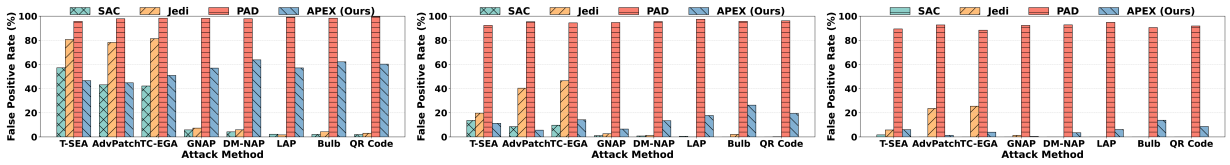
Detector	Distance	w/o defense	LGS	SAC	Jedi	PAD	APEX (Ours)
Faster R-CNN	1m	23.67	15.27	23.67	2.89	11.18	**2.34**
	3m	19.29	11.39	19.29	**0.94**	8.33	0.96
YOLOv3	1m	29.71	16.68	29.71	4.71	6.54	**3.74**
	3m	23.38	15.43	23.38	2.08	11.22	**1.86**
YOLOv4	1m	27.62	17.77	27.62	3.56	15.38	**2.89**
	3m	22.29	16.21	22.29	1.42	10.77	**1.15**
YOLOv5	1m	27.18	19.67	27.18	3.31	16.09	**2.67**
	3m	21.11	16.34	21.11	1.32	9.37	**1.18**
YOLOv8	1m	25.38	18.59	25.38	2.33	13.76	**2.25**
	3m	20.24	15.22	20.24	1.87	8.97	**1.03**

5.3 Effectiveness in Physical Environment

As described in § 5.1, we conduct physical-domain evaluations using a dataset collected by ourselves. In particular, we select the most representative AdvPatch [53] to generate the printable adversarial patch with a standardized size of 28cm×28cm. In the data collection, we capture images at two different distances (*i.e.*, 1m and 3m) to obtain scenarios where the adversarial patch occupies large or small proportion of the image frame. Considering the differences in physical testing environments, the initial attack success rates (ASRs) of AdvPatch show <30% performance across all evaluated object detectors. In our physical domain evaluations, we focus on evaluating defense effectiveness by showing the reduction in ASRs from this baseline.

In Table 4, we present the ASRs after applying APEX to other comparative defense methods, such as LGS, SAC, Jedi, and PAD. Note that we exclude ObjectSeeker in this evaluation because it applies masking directly to the image to defend against adversarial examples without detecting and localizing the adversarial patches. As shown in Table 4, our defense framework outperforms other methods in most tested cases by achieving significant reductions in ASRs. Furthermore, we also observe that APEX presents promising accuracy in patch localization and reconstructs more natural and harmonious images than Jedi. In addition, as shown in Figure 8, we find that PAD underperforms compared to Jedi in physical-domain experiments due to its high false positive rates resulting from the occlusion of parts of the human body.

In addition, there are other factors that can impact APEX's performance in physical-domain evaluations, including environmental lighting conditions and the clothing worn by human subjects. Thus, apart from the previously evaluated settings in the hallway (200 lux), we extend our physical experiments to three more lighting conditions in different scenarios: laboratory (500 lux), balcony (1,000 lux), and outdoor (10,000 lux) environments. Specifically, the empirical results of APEX (*i.e.*, AdvPatch [53] on the YOLOv3 detector at a distance of 3m) show that the ASRs are reduced to 1.86% in the hallway, 1.97% in the laboratory, 3.98% on the balcony, and 3.21% in outdoor environments, which indicates that APEX is effec-



(a) Non-patch pixels $> 1\%$ (b) Non-patch pixels $> 3\%$ (c) Non-patch pixels $> 5\%$

Figure 6: False positive rates under different defense methods across various threshold settings (1%, 3%, 5%).

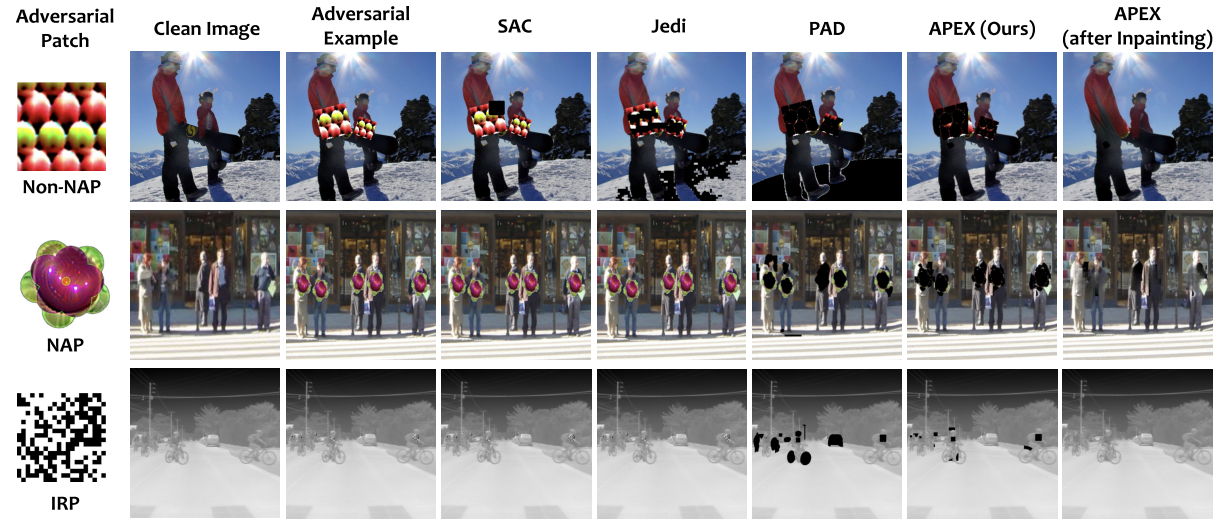


Figure 7: Illustration of defense visualization results against different patch types. SAC and Jedi **fail** to detect both NAP and IRP, while PAD produces false positives for Non-NAP and IRP and misses some NAP detections.

tive across diverse lighting conditions. Additionally, we conduct experiments to explore the potential bias in the physical-domain caused by the clothing worn by human subjects, such as wearing colorful and textured clothing (*e.g.*, black, striped, and checkered) compared with the plain white shirt adopted in most defense settings. As shown in Figure 11, different clothing presents limited impact on APEX performance, leading to the ASRs of 3.95%, 6.28%, and 6.97%, respectively.

5.4 Effectiveness in Abstract Paintings

Unlike real-world images, adversarial patches usually present visual patterns similar to abstract paintings. To evaluate APEX's generalization limits under conditions with unintended deployment scenarios (*e.g.*, patches on abstract paintings), we select the *PeopleArt* [64] dataset that contains 1,529 abstract paintings in different styles (*e.g.*, Cubo-Futurism, Divisionism, Impressionism, and Mechanistic Cubism) and generate adversarial patches on them (*e.g.*, AdvPatch [53]). Figure 9 shows examples of localizing adversarial patches in different abstract paintings, and our empirical results indicate that APEX reduces the ASRs from 71.52% to 17.65% while maintaining acceptable patch localization performance (*i.e.*, a recall of 78.41% with a false positive rate of 27.52%).

In particular, APEX achieves accurate localization in divisionism paintings (*e.g.*, Figure 9b) but presents increasing false positives in mechanistic cubism paintings (*e.g.*, Figure 9d), especially in the facial regions of the human subject. Specifically, there are two reasons to explain the increased false positives and performance degradation in abstract paintings: *i)* the visual patterns and irregular strokes in abstract paintings show similarities to the textures of most adversarial patches, which inevitably increase recognition difficulty; and *ii)* human objects depicted in abstract paintings cannot be detected by most modern object detection models that are trained on real-world human images, which reduces performance in bounding box extractions. As a consequence, these explain that the proposed APEX cannot achieve the anticipated high effectiveness under such extreme visual conditions.

5.5 Effectiveness in Adaptive Attacks

As described in § 3.1, attackers can launch adaptive attacks with white-box knowledge of APEX by modifying existing patch generations. Specifically, based on the third-party framework proposed in [11], we design three adaptive attacks by exploiting AdvPatch [53] with the original loss function $\mathcal{L}_{\text{AdvPatch}}$ and three modified loss functions: $\mathcal{L}_{\text{Adaptive1}}$, $\mathcal{L}_{\text{Adaptive2}}$ and $\mathcal{L}_{\text{Adaptive3}}$. In practice, these modified losses are designed to interfere with the bounding-box extraction and heatmap generation of APEX as the adaptive attack can *i)* impact the bounding-box extraction to prevent APEX from ac-

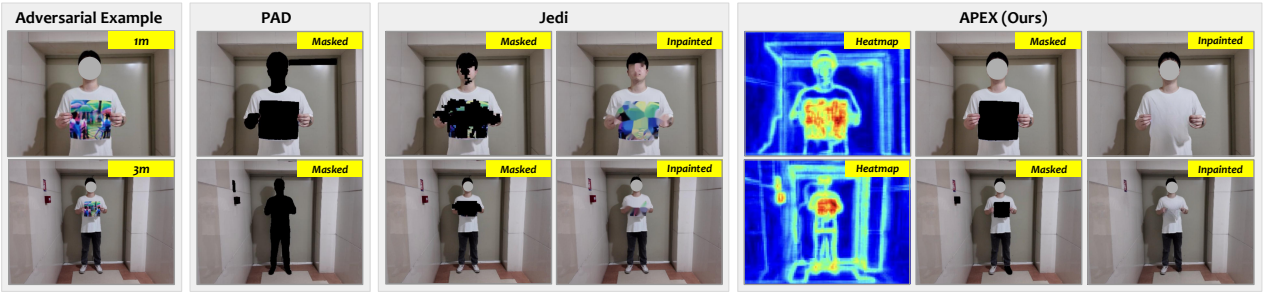
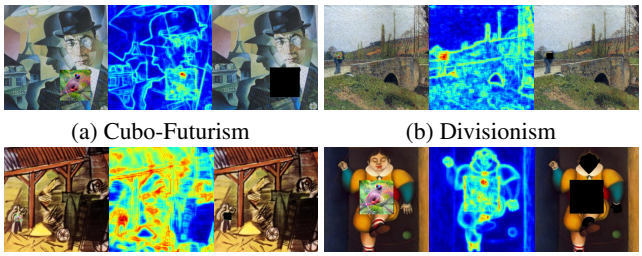


Figure 8: Illustration of defense visualization results against patch in physical environment.



(a) Cubo-Futurism (b) Divisionism

(c) Impressionism (d) Mechanistic Cubism

Figure 9: Illustration of APEX defense results under four different abstract painting styles.

curately identifying potential patch regions; *ii)* utilize mutual information regularization to align the statistical dependencies between the patch and background regions to reduce its saliency on the blur heatmap; and *iii)* apply variation regularization to affect the smoothness by generating irregular and discontinuous edges in the boundary heatmap. More design details and implementations are listed in Appendix C.

In practice, we optimize for 500 epochs with the four loss functions mentioned above and produce adversarial patches shown in Figure 10a. Although the adaptive attacks decrease the proportion of highlighted areas in the heatmap, Figure 10b shows that our proposed APEX can still detect the locations of these adversarial patches. As shown in Figure 13, when facing three types of adaptive attacks, APEX can still restore the model's AP to a level slightly lower than that under non-adaptive attacks, demonstrating its effectiveness in resisting the targeted design of adaptive attacks. It shows partial robustness because loss-function constraints on patch-generated features can at most attenuate the statistical anomalies introduced when patches are embedded into new images, rather than removing them entirely. Compared to the original optimization method, the three adaptive attacks require more epochs to converge. In the optimization, we find that these four loss functions require 18, 25, 40, and 50 seconds per epoch, respectively. Hence, it could consume additional computational overhead in the training and hyperparameters' optimization processes to obtain more effective adaptive adversarial patches.

5.6 Efficiency

As shown in Table 5, our proposed method exhibits different efficiencies in different execution modes. The full execution version (APEX ❸) demonstrates longer processing times than LGS, SAC, and PAD in certain scenarios, particularly with YOLOv4 and YOLOv5. However, when only executing the patch localization module (APEX ❶), the computational efficiency of APEX outperforms Jedi, ObjectSeeker, and PAD. Specifically, the bounding-box extraction, patch detection, and LaMa [50] inpainting stages account for approximately 6.7%, 62.9%, and 30.4% of the total runtime, respectively, and consume 1034MB, 5156MB, and 2413MB of VRAM.

Additionally, Table 5 shows that the average inference time of APEX on a single NVIDIA A6000 GPU is 6.94–9.21 seconds per image. In practice, we can increase the real-time efficiency in APEX by image-level parallel processing because the inference of different images is independent. For example, we extend our experiments to more GPUs (*i.e.*, APEX on YOLOv3 Inria_person) by parallelizing the pipeline across multiple GPUs, which reduces the inference time to 5.09 seconds, 2.58 seconds, and 1.21 seconds on 2, 4, and 8 A6000 GPUs, respectively. This enhancement shows that APEX scales efficiently across multiple GPUs without modifying the detection algorithm, which enables practical deployment across different scenarios. Apart from scaling up GPU resources, we can also realize an additional 5%–10% efficiency improvement by disabling the LaMa [50] refinement, which slightly sacrifices the naturalness of reconstructed images.

5.7 Ablation Study

We conduct an ablation study to evaluate the impact of each module in APEX. Specifically, we evaluate ablation settings in which the bounding box extraction, blur heatmap, and boundary heatmap modules are disabled individually, and compare the corresponding performances with the full pipeline. Table 6 shows the empirical results of our ablation study, which show that APEX's performance decreases in all patch types when disabling different modules. In particular, when the bounding-box extraction module is disabled, we observe performance degradation in both patch localization and detection,

Table 5: Efficiency (Seconds per image). OS: ObjectSeeker. APEX ❶: Patch localization only. APEX ❷: Bounding box extraction and patch localization. APEX ❸: Full pipeline.

Detector	Dataset	LGS	SAC	Jedi	OS	PAD	APEX ❶	APEX ❷	APEX ❸
Faster R-CNN	Inria_Person	0.05	0.16	2.53	3.99	4.99	2.43	4.53	7.47
	MS COCO	0.03	0.11	1.98	3.87	4.36	1.69	4.05	7.28
	FLIR	0.06	0.22	2.79	4.01	5.31	2.68	4.67	7.72
YOLOv3	Inria_Person	0.05	0.19	2.85	4.23	5.76	2.44	4.17	7.39
	MS COCO	0.04	0.13	2.17	4.12	4.78	1.79	3.78	7.08
	FLIR	0.06	0.25	3.35	4.53	6.03	3.01	4.29	7.64
YOLOv4	Inria_Person	0.05	0.17	2.81	4.27	5.97	2.57	5.63	8.89
	MS COCO	0.04	0.11	2.29	4.07	5.21	1.98	5.11	8.31
	FLIR	0.07	0.22	2.94	4.49	6.14	2.71	5.82	9.07
YOLOv5	Inria_Person	0.07	0.23	3.11	4.19	5.94	2.84	5.77	8.98
	MS COCO	0.05	0.17	2.57	4.06	5.32	2.21	5.25	8.42
	FLIR	0.09	0.28	3.32	4.53	6.01	2.93	5.98	9.21
YOLOv8	Inria_Person	0.09	0.14	2.54	1.92	4.57	2.14	3.99	7.31
	MS COCO	0.08	0.09	2.06	1.67	4.21	1.62	3.62	6.94
	FLIR	0.11	0.16	2.86	2.01	4.93	2.53	4.01	7.39

with F1 score dropping by 2.3%–10.4% and mAP decreasing by 5.9%–6.6% across different types of adversarial patches. Furthermore, we notice that the recall rates of patch localization decreases by approximately 6.3%–20.5% when removing the blur heatmap, and the increase of false positive rates by 16.4%–21.4% when removing the boundary heatmap across different types of patches, In addition, we also evaluate the mask expansion module by comparing the inpainting results. The empirical results show that the quality of restored images from LaMa [50] decreases without mask expansion, *i.e.*, PSNR drops from 24.67 to 23.19 and SSIM decreases from 0.9624 to 0.9184, and the images' mAP in object detection decreases by 1.39%, 1.95%, and 0.98% for Non-NAPs, NAPs, and IRPs patches, respectively.

6 Discussion and Limitations

APEX's Performance on Rendered Patches. A prior study [19] has discussed that adversarial patches introduce visual differences that mismatch the surroundings, making them easy to be recognized. Thus, they propose a rendering strategy to mitigate this issue by applying the relighting transform to the patched image. Therefore, in Figure 15, we deploy our defense framework on real patched images and rendered patched images using the exact same method proposed in [19], and the results indicate that APEX can still localize these rendered patches without intensive visual differences. It is reasonable that adversarial patches inevitably cause perturbations in the pixel-level features of images to attack the target deep learning models, which have been adopted in many recent state-of-the-art defense solutions [24, 52].

Limitations. Despite APEX presenting high performance in both digital and physical domains, we still find some limitations in the current design. First, as shown in Figure 16, APEX presents generalization limits and cannot handle the following four failure cases: *i)* textures and color patterns in abstract paintings can obfuscate the patch localization module (Figure 16a); *ii)* blurred infrared images with heuristic settings (*e.g.*, kernel $k > 25$) can reduce patch anomalies and degrade detection performance (Figure 16b); *iii)* extreme low-light conditions make patches difficult to detect, even for human observers (Figure 16c); and *iv)* the patch's shape is close to the target object in the detection models, which leads to incorrect segmentation and misidentification (Figure 16d).

Table 6: Ablation study results in different patches.

Patch Type	Disabled Module	Recall	FP	F1 Score	mAP
Non-NAPs	Bounding box	75.66%	7.92%	0.825	82.59%
	Blur heatmap	61.90%	1.31%	0.714	71.36%
	Boundary heatmap	93.72%	23.97%	0.752	80.21%
	No module disabled	82.42%	5.63%	0.871	88.46%
NAPs	Bounding box	62.74%	15.44%	0.721	73.41%
	Blur heatmap	53.64%	5.97%	0.652	67.85%
	Boundary heatmap	76.22%	32.81%	0.703	72.13%
	No module disabled	67.84%	11.45%	0.805	77.24%
IRPs	Bounding box	62.55%	18.29%	0.686	59.38%
	Blur heatmap	60.12%	12.59%	0.655	58.74%
	Boundary heatmap	68.23%	35.89%	0.579	56.92%
	No module disabled	66.38%	19.46%	0.702	63.11%

Second, we provide heuristic settings in APEX's design and implementation, including thresholds, fusion rules in heatmap generation, and mask expansions. Although we have justified the selection of modules and parameters in § 4 and § 5.1, and the results of ablation study (§ 5.7) demonstrates its effectiveness, the scalability of APEX can be impacted in other complicated physical-world environments and requires fine-tuning of the parameters to achieve more adaptive defenses. Last, it is significant to enhance the real-time efficiency of APEX, especially when enabling the refinement functions. All mentioned limitations will be addressed in our future research.

7 Related Work

Adversarial Patch Attacks against Object Detection. Existing physical adversarial patch attacks can be categorized into hiding attacks [9, 21, 53, 62] and altering attacks [23, 47] based on their objectives. For example, AdvPatch [53] and TC-EGA [21] achieve hiding attacks by employing optimized printable patches to conceal humans in object detection, and [47] launches an altering attack via a targeted label-switch patch attack that can deceive object detectors by causing misclassifications. Furthermore, the patch patterns can be classified into non-naturalistic patches [22, 67, 71] that appear as localized noise and naturalistic patches [9, 20, 23, 31, 51] with semantic meaning, such as dogs [20, 23] and flowers [51], representing natural objects that are difficult to detect. Additionally, the attacks can be effective on both RGB and infrared images [41, 62, 63, 73, 74, 76] (*e.g.*, interference light sources [41, 77] or thermal insulation materials [62, 63, 76]).

Defenses against Adversarial Patch Attacks. There are many research efforts to defend against adversarial patch attacks, such as [13, 24, 25, 33, 35, 39, 52, 66, 68–70], whereas these existing defense methods present several limitations. For instance, certifiable object detection frameworks [68, 69] can improve robustness, but are restricted to specific attack

types (*e.g.*, hiding attacks). The performance of studies such as [35, 52, 66] depends on prior knowledge of patches (*e.g.*, shape or entropy) or requires datasets containing patches for training, presenting low generalizability. Furthermore, despite diffusion-based patch purification methods [13, 25] shows promising performance, they also exhibit practical limitations due to the inherent complexity of diffusion models, resulting in significant computational costs and low efficiency in real-world scenarios. A recent study [11] shows the effectiveness of exploiting customized adversarial patches to defend against detected adversarial patches, at the cost of degrading the original image's visual quality with additional noise. In addition, other solutions [13, 24, 35, 52] face a common challenge of poor robustness when handling abnormal cases. Hence, we overcome these limitations and propose a zero-shot and patch-agonistic defense solution.

8 Conclusion

In this paper, we propose APEX, a zero-shot and patch-agnostic defense framework to identify and localize adversarial patches and reconstruct original images. Our framework addresses limitations in existing adversarial defense methods in special patch cases, including tiny, irregular, and background-coherent patches. Its lightweight and training-free design effectively enhances the efficiency and robustness of defending against adversarial patches in real-world scenarios. The extensive experiments demonstrate that APEX outperforms existing defense methods and achieves high defensive performance in both digital and physical environments while presenting effectiveness against adaptive attacks. In addition, APEX also shows resilience in protecting models against adversarial patch attacks in extreme scenarios such as varying light conditions.

Acknowledgments

This work was supported by the Hong Kong Research Grants Council (RGC) under Grants ECS (CityU 21201420), CRF (8730087), 11219524, 11219025, R6021-20F, R1012-21, RFS2122-1S04, C1029-22G, C6015-23G, and CRS_HKUST601/24, and by the Hong Kong Innovation and Technology Commission (ITC) under Scheme MHP/135/23. Any opinions, findings, and conclusions in this paper are those of the authors and are not necessarily those of the supported organizations. Qingchuan Zhao is the corresponding author.

Ethical Considerations

We develop APEX as a defense framework to mitigate the security risks posed by adversarial patches in vision-based systems. To this end, our study involves generating adversarial patches and evaluating them in both digital and physical environments. Recognizing the potential misuse of adversarial patches, we consider the ethical implications of our work and implement mitigation strategies throughout the research process.

Stakeholders. Our study affects two key stakeholders, including end users and access control system developers. End users may face safety risks when vision-based systems (*e.g.*, surveillance platforms) are compromised by adversarial patches. Additionally, system developers may experience system failures, catastrophic damage, and loss of user trust if adversarial patch attacks are not defended in their model-based detection systems. Therefore, by detecting adversarial patches, restoring images, and improving the reliability of vision-based systems in real-world deployments, APEX is intended to support these stakeholders and to protect their digital privacy and physical safety in critical access control systems.

Ethics in Physical Experiments. As our physical-world experiments involve human participants (*i.e.*, three volunteers with two males and one female, two for wearing patches and one for recording videos), we obtained IRB approval from our institution to implement strict procedures to ensure informed consent and protect the privacy of all participants. Specifically, all participants are informed about the experimental details, such as wearing shirts (*e.g.*, different colors and textures) attached with adversarial patches. Then, all participants need to sign the consent form to allow the usage of their data (*e.g.*, videos) collected in different conditions (*e.g.*, light conditions) for physical experiment evaluations.

Mitigating Potential Harm. To mitigate potential misuse of adversarial patches generated in the experiments, we ensure that all adversarial patches generated for evaluation are used solely for testing purposes and are not shared, released, or reused in any other contexts. Furthermore, all physical adversarial patches are printed and evaluated only in experimental settings in laboratory and are removed and destroyed after data collection. Next, data collection from volunteer participants were conducted with the supervision of IRB committee. In addition, the captured videos from all participants were stored on an encrypted server to prevent any form of privacy leakage pertaining to the volunteers.

Open Science

To align with USENIX Security's Open Science policy, we make all relevant source code and supporting scripts publicly available at the recommended third-party platform Zenodo: https://doi.org/10.5281/zenodo.17915537. Meanwhile, we provide the repository with a detailed, step-by-step README.md about user instructions. Furthermore, the datasets we use, including *MS COCO* [32], *INRIA_Person* [8], *FLIR_ADAS_v1_3* [12], *PeopleArt* [64] and *Diverse-Weather Dataset* [65] are publicly available and can be accessed and downloaded from their official websites. Overall, we totally support the policy and make our artifact publicly available to facilitate reproducibility and foster further research.

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 2009.

[2] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of CVPR*, 2001.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[4] Folkmar Bornemann and Tom März. Fast image inpainting based on coherence transport. *Journal of Mathematical Imaging and Vision*, 2007.

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

[6] Niklas Bunzel, Ashim Siwakoti, and Gerrit Klause. Adversarial patch detection and mitigation by detecting high entropy regions. In *Proceedings of DSN Workshop*, 2023.

[7] Ciprian Corneanu, Raghudeep Gadde, and Aleix M Martinez. Latentpaint: Image inpainting in latent space with diffusion models. In *Proceedings of WACV*, 2024.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of CVPR*, 2005.

[9] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Proceedings of CVPR*, 2020.

[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of CVPR*, 2018.

[11] Jianan Feng, Jiachun Li, Changqing Miao, Jianjun Huang, Wei You, Wenchang Shi, and Bin Liang. Fight fire with fire: Combating adversarial patch attacks using pattern-randomized defensive patches. In *Proceedings of IEEE S&P*, 2025.

[12] FLIR. Free flir thermal dataset for algorithm training. [EB/OL]. https://www.flir.com/oem/adas/adas-dataset-form/ Accessed Jun. 12, 2025.

[13] Jia Fu, Xiao Zhang, Sepideh Pashami, Fatemeh Rahimian, and Anders Holst. Diffpad: Denoising diffusion-based adversarial patch decontamination. In *Proceedings of WACV*, 2025.

[14] Estevão S Gedraite and Murielle Hadad. Investigation on the effect of a gaussian blur in image filtering and segmentation. In *Proceedings of ELMAR*, 2011.

[15] Ross Girshick. Fast R-CNN. In *Proceedings of ICCV*, 2015.

[16] Mahdi Golizadeh, Nassibeh Golizadeh, Mohammad Ali Keyvanrad, and Hossein Shirazi. Architectural insights into knowledge distillation for object detection: A comprehensive review. *arXiv preprint arXiv:2508.03317*, 2025.

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

[18] Amira Guesmi, Muhammad Abdullah Hanif, and Muhammad Shafique. Advrain: Adversarial raindrops to attack camera-based smart vision systems. *Information*, 2023.

[19] Nabeel Hingun, Chawin Sitawarin, Jerry Li, and David Wagner. Reap: A large-scale realistic adversarial patch benchmark, 2023.

[20] Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, Jun-Cheng Chen, Kai-Lung Hua, and Wen-Huang Cheng. Naturalistic physical adversarial patch for object detectors. In *Proceedings of ICCV*, 2021.

[21] Zhanhao Hu, Siyuan Huang, Xiaopei Zhu, Fuchun Sun, Bo Zhang, and Xiaolin Hu. Adversarial texture for fooling person detectors in the physical world. In *Proceedings of CVPR*, 2022.

[22] Hao Huang, Ziyan Chen, Huanran Chen, Yongtao Wang, and Kevin Zhang. T-sea: Transfer-based self-ensemble attack on object detection. In *Proceedings of CVPR*, 2023.

[23] Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, Alan L Yuille, Changqing Zou, and Ning Liu. Universal physical camouflage attacks on object detectors. In *Proceedings of CVPR*, 2020.

[24] Lihua Jing, Rui Wang, Wenqi Ren, Xin Dong, and Cong Zou. Pad: Patch-agnostic defense against adversarial patch attacks. In *Proceedings of CVPR*, 2024.

[25] Caixin Kang, Yinpeng Dong, Zhengyi Wang, Shouwei Ruan, Yubo Chen, Hang Su, and Xingxing Wei. Diffender: Diffusion-based adversarial defense against patch attacks. In *Proceedings of ECCV*, 2024.

[26] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. In *Proceedings of ICML*, 2018.

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of ICCV*, 2023.

[28] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019.

[29] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 2004.

[30] Weijian Liang and Yaoru Sun. ELCNN: A deep neural network for small object defect detection of magnetic tile. *IEEE Transactions on Instrumentation and Measurement*, 2022.

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

[32] 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 *Proceedings of ECCV*, 2014.

[33] Zijin Lin, Yue Zhao, Kai Chen, and Jinwen He. I don't know you, but i can catch you: Real-time defense against diverse adversarial patches for object detectors. In *Proceedings of CCS*, 2024.

[34] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. Pd-gan: Probabilistic diverse gan for image inpainting. In *Proceedings of CVPR*, 2021.

[35] Jiang Liu, Alexander Levine, Chun Pong Lau, Rama Chellappa, and Soheil Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *Proceedings of CVPR*, 2022.

[36] 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.

[37] Yang Liu, Peng Sun, Nickolas Wergeles, and Yi Shang. A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, 2021.

[38] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of CVPR*, 2022.

[39] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *Proceedings of WACV*, 2019.

[40] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *Proceedings of ICCV Workshops*, 2019.

[41] Muyao Niu, Zhuoxiao Li, Yifan Zhan, Huy H Nguyen, Isao Echizen, and Yinqiang Zheng. Physics-based adversarial attack on near-infrared human detector for nighttime surveillance camera systems. In *Proceedings of ACM MM*, 2023.

[42] F Ozge Unel, Burak O Ozkalayci, and Cevahir Cigla. The power of tiling for small object detection. In *Proceedings of CVPR Workshops*, 2019.

[43] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of CVPR*, 2016.

[44] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of CVPR*, 2017.

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

[46] Daniel B Russakoff, Carlo Tomasi, Torsten Rohlfing, and Calvin R Maurer Jr. Image similarity using mutual information of regions. In *Proceedings of ECCV*, 2004.

[47] Avishag Shapira, Ron Bitton, Dan Avraham, Alon Zolfi, Yuval Elovici, and Asaf Shabtai. Attacking object detector using a universal targeted label-switch patch. *arXiv preprint arXiv:2211.08859*, 2022.

[48] Abhijith Sharma, Yijun Bian, Phil Munz, and Apurva Narayan. Adversarial patch attacks and defences in vision-based tasks: A survey. *arXiv preprint arXiv:2206.08304*, 2022.

[49] 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 *Proceedings of USENIX WOOT*, 2018.

[50] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust

large mask inpainting with fourier convolutions. In *Proceedings of WACV*, 2022.

[51] 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 ACM MM*, 2021.

[52] 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 CVPR*, 2023.

[53] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of CVPR Workshops*, 2019.

[54] Min-Jen Tsai, Ping-Yi Lin, and Ming-En Lee. Adversarial attacks on medical image classification. *Cancers*, 2023.

[55] ultralytics. Ultralytics yolov5, 2022. https://github.com/ultralytics/yolov5.

[56] ultralytics. Ultralytics yolov8, 2023. https://docs.ultralytics.com.

[57] Paul Viola and William M Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 1997.

[58] Dwi Wahyudi, Indah Soesanti, and Hanung Adi Nugroho. Toward detection of small objects using deep learning methods: A review. In *Proceedings of ICITEE*, 2022.

[59] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of CVPR*, 2023.

[60] Jian Wang, Fan Li, and Lijun He. A unified framework for adversarial patch attacks against visual 3d object detection in autonomous driving. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[61] Wentao Wang, Li Niu, Jianfu Zhang, Xue Yang, and Liqing Zhang. Dual-path image inpainting with auxiliary gan inversion. In *Proceedings of CVPR*, 2022.

[62] Hui Wei, Zhixiang Wang, Xuemei Jia, Yinqiang Zheng, Hao Tang, Shin'ichi Satoh, and Zheng Wang. Hotcold block: Fooling thermal infrared detectors with a novel wearable design. In *Proceedings of AAAI*, 2023.

[63] Xingxing Wei, Jie Yu, and Yao Huang. Physically adversarial infrared patches with learnable shapes and locations. In *Proceedings of CVPR*, 2023.

[64] Nicholas Westlake, Hongping Cai, and Peter Hall. Detecting people in artwork with cnns. In *Proceedings of ECCV*, 2016.

[65] Aming Wu and Cheng Deng. Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In *Proceedings of CVPR*, 2022.

[66] Siyang Wu, Jiakai Wang, Jiejie Zhao, Yazhe Wang, and Xianglong Liu. Napguard: Towards detecting naturalistic adversarial patches. In *Proceedings of CVPR*, 2024.

[67] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *Proceedings of ECCV*, 2020.

[68] Chong Xiang and Prateek Mittal. Detectorguard: Provably securing object detectors against localized patch hiding attacks, 2021.

[69] Chong Xiang, Alexander Valtchanov, Saeed Mahloujifar, and Prateek Mittal. Objectseeker: Certifiably robust object detection against patch hiding attacks via patch-agnostic masking. In *Proceedings of IEEE S&P*, 2023.

[70] Chong Xiang, Tong Wu, Sihui Dai, Jonathan Petit, Suman Jana, and Prateek Mittal. PatchCURE: Improving certifiable robustness, model utility, and computation efficiency of adversarial patch defenses. In *Proceedings of USENIX Security*, 2024.

[71] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *Proceedings of ECCV*, 2020.

[72] Zheng Yang, Xu Wang, Jiahang Wu, Yi Zhao, Qiang Ma, Xin Miao, Li Zhang, and Zimu Zhou. Edgeduet: Tiling small object detection for edge assisted autonomous mobile vision. *IEEE/ACM Transactions on Networking*, 2022.

[73] Shuai Yuan, Xingshuo Han, Hongwei Li, Guowen Xu, Wenbo Jiang, Tao Ni, Qingchuan Zhao, and Yuguang Fang. The fluorescent veil: A stealthy and effective physical adversarial patch against traffic sign recognition. In *Proceedings of NeurIPS*, 2025.

[74] Shuai Yuan, Hongwei Li, Rui Zhang, Hangcheng Cao, Wenbo Jiang, Tao Ni, Wenshu Fan, Qingchuan Zhao, and Guowen Xu. Omni-angle assault: An invisible and powerful physical adversarial attack on face recognition. In *Proceedings of ICML*, 2025.

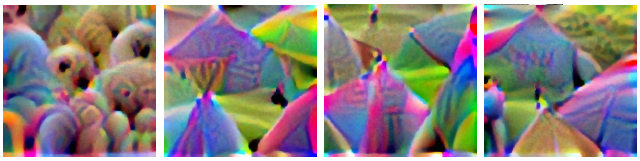
[75] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of CCS*, 2019.

[76] Xiaopei Zhu, Zhanhao Hu, Siyuan Huang, Jianmin Li, and Xiaolin Hu. Infrared invisible clothing: Hiding from infrared detectors at multiple angles in real world. In *Proceedings of CVPR*, 2022.

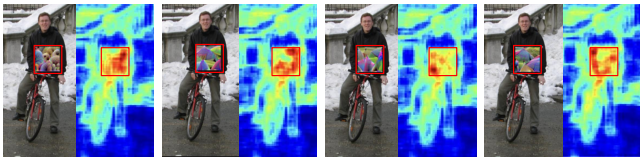
[77] Xiaopei Zhu, Xiao Li, Jianmin Li, Zheyao Wang, and Xiaolin Hu. Fooling thermal infrared pedestrian detectors in real world using small bulbs. In *Proceedings of AAAI*, 2021.

A Parameter Sensitivity Analysis

In our evaluations, we have several observations when analyzing the sensitivity of APEX to reasonable parameter variations: *i)* using the hyperparameter selection experiments (*e.g.*, τ, θ_{cov}, $\alpha_{\text{blur}}/\alpha_{\text{bnd}}$, k), we observe that parameter variations only show apparent impact on false positive rates, while the recall rates remain stable. For instance, the F1 score increases from 0.617 to 0.826 while the recall varies only from 92.47% to 96.73%, showing less than 5% changes, which indicates that APEX is not highly sensitive to reasonable hyperparameter variations. *ii)* We find that the threshold percentile τ has the most significant impact on F1 score, with the F1 score increasing as the threshold becomes higher, primarily because a higher threshold reduces false positives. In contrast, the blur kernel size k presents the minimal influence with less than approximately 1% variations in the F1 score. *iii)* We observe that APEX shows better performance on infrared images if $\alpha_{\text{bnd}} > \alpha_{\text{blur}}$, whereas it performs better in RGB images when $\alpha_{\text{bnd}} < \alpha_{\text{blur}}$. Meanwhile, the defense framework maintains a consistently high recall rate when we set α_{blur} and α_{bnd} in a reasonable range (*e.g.*, 0.3–0.7).



(a) Patches optimized by different loss functions



(b) Patched images with corresponding heatmaps

Figure 10: Comparison of adaptive patches optimized by four different loss functions ($\mathcal{L}_{\text{AdvPatch}}$, $\mathcal{L}_{\text{Adaptive1}}$, $\mathcal{L}_{\text{Adaptive2}}$, $\mathcal{L}_{\text{Adaptive3}}$ from left to right).

B Effectiveness in Different Weathers

In addition, we further evaluate the robustness of APEX's patch localization performance under different weather conditions by attaching patches to objects such as pedestrians in the *Diverse-Weather Dataset* [65], which contains 63,636 images collected from five different weather scenes: daytime-sunny, night-sunny, dusk-rainy, night-rainy, and daytime-foggy. In particular, the evaluation results show that APEX outperforms other patch defenses by achieving an average recall rate of 71.93%, whereas PAD, Jedi, and SAC only achieves 62.19%, 43.88%, and 39.21% recall rates, respectively

C Design of Adaptive Attacks

We only modify the loss function design while keeping all other components consistent with the original framework, thus conducting training on YOLOv2 [44].

The overall adversarial patch loss function is defined as:

$$\mathcal{L}_{\text{AdvPatch}} = \alpha\mathcal{L}_{\text{tv}} + \beta\mathcal{L}_{\text{nps}} + \mathcal{L}_{\text{obj}} \tag{17}$$

where the component terms are:

$$\mathcal{L}_{\text{obj}} = -\log p(y_{\text{target}} \mid x+\delta) \tag{18}$$

$$\mathcal{L}_{\text{tv}} = \sum_{i,j} \sqrt{(\delta_{i+1,j} - \delta_{i,j})^2 + (\delta_{i,j+1} - \delta_{i,j})^2} \tag{19}$$

$$\mathcal{L}_{\text{nps}} = \sum_{i,j} \min(c_{\text{printable}} - \delta_{i,j}, 0) \tag{20}$$

The adversarial patch loss combines three components: the objective loss $\mathcal{L}_{\text{obj}}$ maximizes the target class probability for the patched image, the total variation regularization $\mathcal{L}_{\text{tv}}$ enforces spatial smoothness by computing pixel-wise differences between neighbors, and the non-printability score $\mathcal{L}_{\text{nps}}$ constrains colors to the printable range $c_{\text{printable}}$.

Based on our knowledge of the APEX architecture, we propose designing adaptive attack patches from three perspectives as discussed in § 5.5. For case i), we introduce a *bounding box attack loss* ($\mathcal{L}_{bb}$) defined as:

$$\mathcal{L}_{bb} = \frac{1}{B}\sum_{i=1}^{B} \text{Count}\left(\{b \in \text{Boxes}_i : \text{conf}(b) > \tau_{conf}\}\right) \tag{21}$$

where: B = batch size, Boxes_i = all detected bounding boxes in the i-th image, $\text{conf}(b)$ = confidence score of box b,τ_{conf} = confidence threshold. The objective is to minimize $\mathcal{L}_{bb}$ during optimization, thus reducing the number of valid bounding boxes. The total adversarial loss combines the standard adversarial patch loss with our bounding box suppression term:

$$\mathcal{L}_{\text{Adaptive1}} = \mathcal{L}_{\text{AdvPatch}} + \lambda_1 L_{bb} \tag{22}$$

For case ii), we introduce a *mutual information loss* (L_{mi}) to enforce statistical indistinguishability between the patch

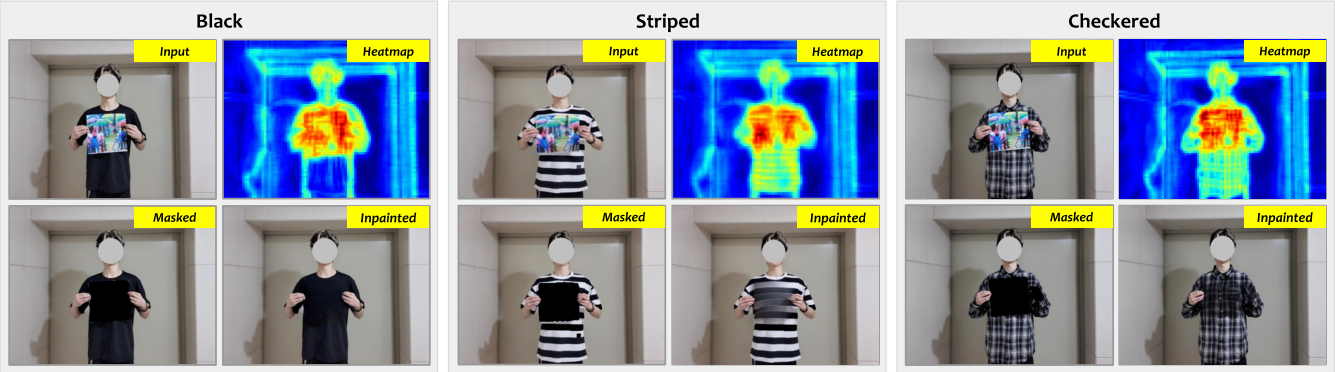
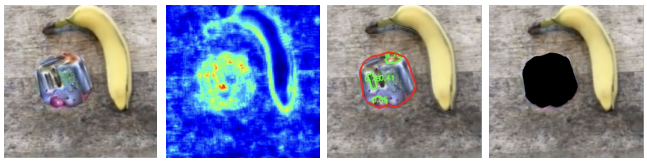


Figure 11: Illustration of defense visualization results in physical environments with different clothing textures.



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

Figure 12: Localization and mitigation of adversarial patches in an image classification task. (a) Adversarial example containing a LaVAN [26] patch. (b) Heatmap. (c) Localization result. (d) Defended image.

and background:

$$L_{mi} = \frac{1}{B}\sum_{i=1}^{B}\sum_{c=1}^{3}\left[(\mu_{patch}^{(i,c)} - \mu_{bg}^{(i,c)})^2 + (\sigma_{patch}^{(i,c)} - \sigma_{bg}^{(i,c)})^2\right] \quad (23)$$

where: $\mu_{patch}^{(i,c)}$ denotes the mean of patch region in channel c of the i-th image, $\mu_{bg}^{(i,c)}$ represents the mean of background annulus in channel c, while $\sigma_{patch}^{(i,c)}$ and $\sigma_{bg}^{(i,c)}$ are the corresponding standard deviations. All statistical measures are computed on Gaussian-blurred images which is the same as the defense setting.

The composite adversarial objective combines standard patch optimization with statistical concealment:

$$\mathcal{L}_{\text{Adaptive2}} = \mathcal{L}_{\text{AdvPatch}} + \lambda_2 L_{mi} \quad (24)$$

The optimization aims to minimize L_{mi}, achieving statistical similarity between patch and background regions ($\mu_{patch} \approx \mu_{bg}$), matching distribution characteristics ($\sigma_{patch} \approx \sigma_{bg}$), and effectively evading blur heatmap detection.

Moreover, by incorporating both $\mathcal{L}_{bb}$ and $\mathcal{L}_{mi}$ into the original loss function, we obtain:

$$\mathcal{L}_{\text{Adaptive3}} = \mathcal{L}_{\text{AdvPatch}} + \lambda_1 L_{bb} + \lambda_2 L_{mi} \quad (25)$$

This leads to the third adaptive attack strategy, which simultaneously suppresses the number of bounding boxes and reduces the statistical distinguishability between the patch and the background.

D Extension Defense to Other Models

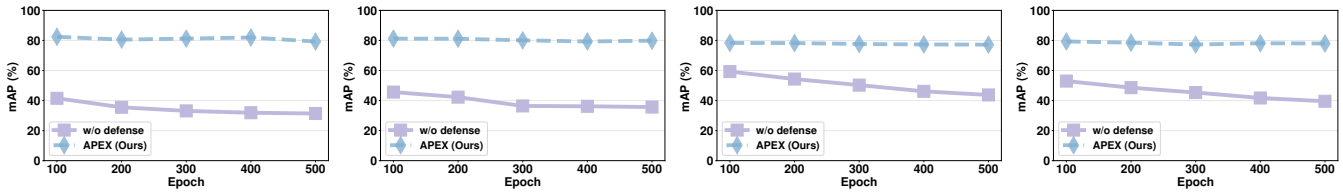
The defense framework, APEX, can be extended to defend against adversarial patches that change the statistical characteristics of images in other models across varying deep learning tasks (*e.g.*, image classification). For instance, Figure 12 shows APEX's defensive effectiveness in detecting and localizing adversarial patches in image classification models (*e.g.*, LaVAN [26]). In addition, the proposed bounding-box extraction module can be deployed in other tasks to prevent signal dilution resulting from background interference and pinpoint the analysis scope, such as partitioning class activation maps in image classification models and localizing facial areas in face detection and recognition systems.

E Challenging Cases Selection

For scenario *i)* where the patch occupies a small area, we follow the MS COCO dataset's definition that objects with pixel areas smaller than 32 $\times$ 32 are considered small targets. Accordingly, we select images containing patches smaller than 32 $\times$ 32 pixels as our small-patch test set. *ii)* Regarding irregularly shaped patches, we utilize three distinct floral-patterned patches generated by the LAP [51] attack method as our irregular-shape testing set, since LAP naturally produces patches with various shapes. For scenario *iii)* involving background-coherent patches, while visual inspection remains the primary evaluation method due to the lack of quantitative metrics, we simplify the selection criteria by choosing patches and backgrounds with high entropy ($>$ 7.0) and edge density ($>$ 0.18).

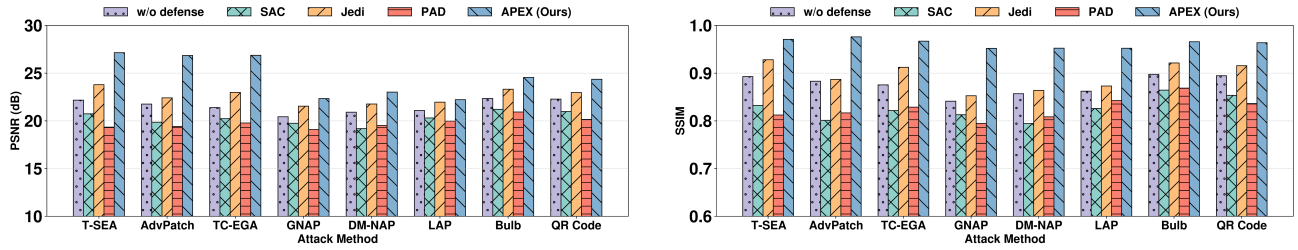
F Defense Image Quality

To defend against adversarial patch attacks and reconstruct images, we need to accurately detect patches while maintain the quality of image restoration because distortions or information loss in restored images could adversely affect



(a) $\mathcal{L}_{\text{AdvPatch}}$ (b) $\mathcal{L}_{\text{Adaptive1}}$ (c) $\mathcal{L}_{\text{Adaptive2}}$ (d) $\mathcal{L}_{\text{Adaptive3}}$

Figure 13: The defense effectiveness of APEX under adaptive attacks varies with training epochs, which incur an additional computational overhead of 25-50 seconds per epoch, yet only cause limited degradation in APEX's defensive performance.



(a) PSNR (b) SSIM

Figure 14: Image quality evaluation (PSNR and SSIM) of defense methods against adversarial patch attacks.

(a) Real patch (b) Rendered patch

Figure 15: Illustration of patch localization results for real-world physical patches and rendered patches in [19].

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

Figure 16: Potential failure cases. (a) Abstract painting. (b) Blurry infrared image. (c) Extremely low-light scenarios. (d) Human-shape patch with partial human body.

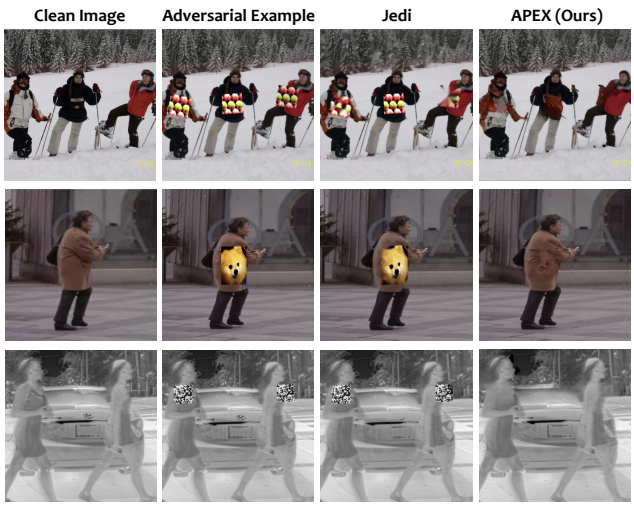


Figure 17: Illustration of adversarial patch purification results against different patch types.

the accuracy and usability of downstream tasks. Hence, we use quality metrics include *Peak Signal-to-Noise Ratio* (PSNR) and *Structural Similarity Index* (SSIM). Specifically, PSNR measures pixel-level differences between restored and original images, where higher values indicate greater similarity and better image clarity. SSIM comprehensively evaluates image similarity across three dimensions: luminance, contrast, and structure, where values close to 1 indicate better structural consistency and good visual quality.

As shown in Figure 14, our method achieves the highest PSNR and SSIM scores in all attack scenarios, demonstrating its superior capability in preserving both image clarity and structural consistency after adversarial patch purification, which outperforms existing defense methods. Additionally, SAC and PAD employ black mask purification, thus we omit their visualization. In contrast, Jedi utilizes a coherence transport-based inpainting method [4], so we present the inpainted results of both Jedi and our method in Figure 17. It can be observed that Jedi still exhibits significant deficiencies in patch localization, and the restored parts seems unnatural. In contrast, APEX achieves precise patch localization and utilizes LaMa to reconstruct images similar to original images.