This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3603988

No Trespassing: Ground-view Adversarial Patches for Privacy-aware Management in COTS Robot Vacuum Cleaner

Shuai Yuan, Guowen Xu (Corresponding Author), *Senior Member, IEEE*, Hongwei Li, *Fellow, IEEE*, Rui Zhang, Hangcheng Cao, Xinyuan Qian, Tao Ni, Qingchuan Zhao, Yuguang Fang, *Fellow, IEEE*

***Abstract*—Robot vacuum cleaners (RVCs) with autonomous navigation and decision-making capabilities have become an integral part of modern homes. During their operations, these devices may inadvertently enter privacy-sensitive areas, leading to potential privacy breaches. However, existing defense methods risk exposing the location of private areas, require root privileges, or are designed for infrared sensors that are ineffective for camera-based RVCs. To overcome these limitations, we propose a novel solution, a ground-view adversarial patch named GPatch, preventing RVCs from entering privacy-sensitive areas. Users only need to place GPatch at the entrance of restricted areas to prevent an RVC's unauthorized access, while also providing a warning to unauthorized individuals. We evaluate GPatch in real-world environments with an average success rate of 87.27%, and experimental results demonstrate its effectiveness, robustness, and transferability, making it a practical, user-friendly, and reliable solution for safeguarding privacy in home environments.**

⚠ Disclaimer: This paper contains images that some readers may find offensive, disturbing, and/or distressing. Reader discretion is advised.

***Index Terms*—Ground-view adversarial patch, Robot vacuum cleaner, Privacy-sensitive areas.**

I. Introduction

In recent years, demand for commercial off-the-shelf (COTS) robot vacuum cleaners (RVCs) has surged, driving the global RVC market to $4.5 billion in 2023, with a projected compound annual growth rate of 6.5% through 2032 [1]. Recent trends in the RVCs market [2], [3] is the integration of cameras to enhance artificial intelligence (AI) navigation and obstacle avoidance. According to IDC's reports [4], [5], Roborock [6] has led the global RVC market share for two consecutive quarters, with all of its 2024 products featuring camera integration and AI models. However, as shown in Figure 1, a camera-based RVC may unintentionally invade privacy-sensitive spaces, capturing intimate moments such as a young woman in a lavender T-shirt sitting on a toilet with her shorts pulled down to mid-thigh, or a couple engaging in a kiss [7], [8]. It may also monitor personal activities [9], such as a man drinking coffee [10].

(a) A woman sitting on toilet (b) A couple kissing (c) A man drinking coffee

(d) RVC can only move in non-sensitive areas

Fig. 1: Illustration of preventing an RVC from entering a privacy-sensitive area. (a-c) Private pictures were taken by an RVC where we obscure faces with black circles. (d) GPatch can be recognized as obstacles (e.g., shoes) by the detector, thus preventing the RVC from entering the privacy-sensitive area.

Existing privacy-preserving works can be categorized into post-processing [11]–[14] or prevention-at-source [15]–[18] approaches. Post-processing approaches take effect only after data capture, meaning sensitive information may still be temporarily stored within the system, increasing the risk of unintended privacy exposure. An ideal RVC should be capable of identifying and avoiding entry into privacy-sensitive spaces, thereby preventing privacy breaches at the source. These prevention-at-source solutions can be classified into digital-based [19], [20] and physical-based [21], [22] approaches

Shuai Yuan, Guowen Xu, Hongwei Li, Rui Zhang and Xinyuan Qian are with the school of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.(e-mail: mk2456mk@gmail.com; guowen.xu@uestc.edu.cn; hongweili@uestc.edu.cn; ruizhangsec@163.com; xinyuanqian@outlook.com)

Hangcheng Cao, Tao Ni, Qingchuan Zhao and Yuguang Fang are with the Department of Computer Science, City University of Hong Kong, China. (e-mail: hangccao@cityu.edu.hk; taoni2@cityu.edu.hk; qizhao@cityu.edu.hk; my.fang@cityu.edu.hk)

depending on the implementation method. Firstly, digital-based methods usually set restricted areas [23], [24] via a mobile app. However, an attacker could potentially hack the RVC to learn the location of sensitive areas, resulting in privacy leakage. In addition, digital-based methods require root privileges, which is not user-friendly for the elderly and children, and other household members may be unaware of these designated private areas. Secondly, physical-based methods typically employ actual objects to create virtual walls, such as magnetic strips [25], [26] and infrared emitters [22], [27]. Unfortunately, this method is designed for infrared sensors, making it invalid with camera-based RVCs. Therefore, there is an urgent need to prevent camera-based RVC intrusion into privacy-sensitive areas.

Inspired by the development of vaccines [28], the concept of "prevention through attacking" emerges. Vaccines function by mimicking a virus to stimulate the immune system for preventive purposes. Similarly, we observe that AI models are inherently vulnerable to physical adversarial attacks [29]–[33], causing models to misidentify or incorrectly detect targets. Thus, we leverage this vulnerability by using adversarial examples to create virtual spatial obstacles (as illustrated in Figure 1), to effectively prevent RVCs from entering privacy-sensitive areas. Although adversarial attack mechanisms have been extensively studied, their customized deployment in RVCs still faces several unique challenges. First, most adversarial patches fail to incorporate semantic information, rendering them effective only in misleading machine recognition while remaining incomprehensible to humans, who are unable to identify restricted privacy-sensitive areas. Secondly, all existing adversarial patches are generated from a frontal perspective. Consequently, these adversarial examples are ineffective for RVCs to capture image information from low-angle perspectives. Finally, an RVC may approach an adversarial patch from various angles and adapt its cleaning strategy according to different obstacles, posing a significant challenge to the robustness and effectiveness of adversarial patches in real-world applications.

To address the aforementioned challenges, we propose a ground-view adversarial patch, referred to as GPatch[1], to prevent RVCs from entering privacy-sensitive areas. As illustrated in Figure 1, GPatch incorporates a clearly visible warning message, "STOP," to enable human users to easily identify restricted areas. We can see that an RVC interprets the GPatch [2] placed at an entry point as a pair of shoes, effectively blocking its access to sensitive areas and thereby safeguarding user privacy. In the implementation process, we first introduce a warning embedding method that overlays a visible warning message onto the images captured by an RVC. Second, we develop a comprehensive optimization framework that jointly considers factors such as patch location, bounding box alignment, classification probabilities, and patch color to optimize adversarial perturbations. Additionally, we apply perspective transformation techniques to adapt the adversarial patch from a ground view to a frontal view, ensuring compatibility with an RVC's visual perception. Finally, to enhance the robustness of GPatch in real-world scenarios, we analyze the unique target classes prevalent in indoor environments and employ a combination of advanced transformation algorithms.

Our GPatch is not a digital-based method, and an attacker hacking into RVCs cannot access the location of privacy-sensitive areas. Users simply need to place GPatch at the entrance of areas where access restrictions are required, and it will work on both RVCs and individuals without root privileges. Thus, GPatch addresses the limitations of current solutions. We evaluate the performance of GPatch in the physical world across various models, with an average prevention success rate of 68% for one-stage detectors and 83% for two-stage detectors, and compare it against two baseline methods. Additionally, through ablation experiments, we assess the impact of different parameters on GPatch's performance. Furthermore, we examine its robustness under diverse physical environmental factors, such as ambient light and flooring materials. Finally, we test the transferability of GPatch across different models.

Our contributions are summarized as follows.

- We are the first to introduce adversarial patches to prevent RVCs from entering privacy-sensitive areas.
- We design GPatch, a ground-view adversarial patch, that incorporates semantic representation features for enhanced effectiveness and robustness in real-world environments.
- We conduct extensive evaluations in the physical world to assess GPatch's effectiveness, robustness, and transferability.

II. Background and Related Works

A. Robot Vacuum Cleaner

Functionality and components. As integral devices in smart home ecosystems, robotic vacuum cleaners (RVCs) are equipped with a variety of functionalities that enhance their efficiency and adaptability, including autonomous navigation, obstacle detection, cleaning, and mapping. These capabilities are powered by a combination of core components, including advanced sensor systems, path planning modules, power systems, and control systems, which work cohesively to ensure optimal performance in dynamic environments [34]–[36]. At the core of these capabilities, path-planning modules leverage algorithms such as SLAM (Simultaneous Localization and Mapping) to generate efficient cleaning routes while avoiding obstacles and maximizing coverage. Power systems provide sustained operation, often incorporating features like automatic docking and recharging to improve user convenience. Meanwhile, control systems integrate these components to enable precise task execution, seamless communication with other smart home devices, and adaptability to user preferences. Among these components, sensor systems are particularly critical, as they underpin key processes such as environmental perception and obstacle detection. By utilizing technologies like infrared sensors, LiDAR, cameras, and ultrasonic sensors, these systems gather spatial data that supports real-time adaptation to changes in the surroundings, ensuring smooth and autonomous operation [37], [38].

[1]Our codes of GPatch can be found at https://anonymous.4open.science/r/seucre_robot-14B7/.

[2]Compared to commonly used obstacles in daily life, such as chairs or doors, which are employed to prevent unauthorized robot access, the proposed patch design effectively achieves blocking function without causing any inconvenience to authorized individuals entering the designated areas. This ensures both practicality and convenience.

TABLE I: Comparative analysis with related privacy-aware management in RVCs. "W/o Privacy" means the RVC does not capture or store any privacy-sensitive information, "Location" ensures that attackers cannot determine restricted zone positions even if the RVC is compromised, "W/o Root" indicates the method requires no root access, "Camera" shows it is compatible with camera-only RVCs, and "Semantics" means the restricted zone contains semantic information that is easy for family members to recognize.

Type	Method	Domain	Vector	W/o Privacy	Location	W/o Root	Camera	Semantics
Post-processing	[13], [14]	Digital	Masking		✓	✓	✓	
Prevention-at-source	[23], [24]	Digital	Keep-out zone	✓			✓	
	[25], [26]	Physical	Magnetic strip	✓	✓	✓		
	[22], [27]	Physical	Infrared emitter	✓	✓	✓		
	GPatch	Physical	Adversarial patch	✓	✓	✓	✓	✓

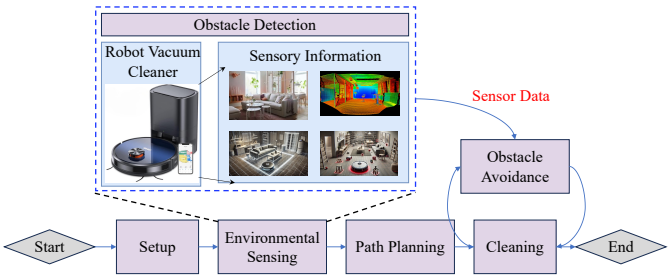


Fig. 2: The workflow of robot vacuum cleaners.

The operational workflow of an RVC involves several sequential stages: setup, environmental sensing, path planning, obstacle avoidance, and task execution, as illustrated in Figure 2. During the environmental sensing phase, RVCs collect spatial data using various sensors, including cameras, LiDAR, and 3D structured light, to define the working areas. This data is then processed by path planning modules to generate an optimal cleaning route. Obstacle detection is a critical step in this workflow, as it directly impacts navigation safety and cleaning efficiency. By classifying and responding to obstacles in real time, RVCs can maintain seamless operation and adapt dynamically to their surroundings during cleaning tasks. In our study, we focus on using cameras as the primary sensor for obstacle detection. The rationale behind this choice is discussed in detail in supplemental materials.

Challenges in sensitive area protection. Despite their advances, RVCs face challenges in protecting sensitive areas. Existing methods for setting no-go zones include mobile app configurations, virtual walls, and magnetic strips. Mobile app setups require root-level permissions, which are cumbersome and not user-unfriendly. Virtual walls use infrared signals to create boundaries [39], but these often fail with camera-based robots and add maintenance overhead. Magnetic stripes [40] require RVCs to be equipped with cliff sensors and are sensitive to ambient lighting. These methods lack user-friendly deployment and robustness, necessitating a more efficient solution for sensitive area protection [41].

B. Object Detection

RVCs commonly rely on object detection to support obstacle avoidance and optimize cleaning strategies. Object detection enables the RVC to identify and locate objects of interest, such as furniture, wires, and small household items, based on input from onboard cameras or RGB-D sensors. Modern object detection models can generally be categorized into two types: one-stage detectors and two-stage detectors. One-stage detectors, e.g., SSD [42], YOLOv3 [43], and YOLOv7 [44], perform both localization and classification in a single stage, which makes them computationally efficient and suitable for real-time applications in RVCs. Two-stage detectors, in contrast, divide the detection process into two steps. They first generate candidate object regions through a region proposal network, and then perform classification within each proposed region. This approach typically achieves higher accuracy but requires more computational resources. Representative models include Faster R-CNN [45], Mask R-CNN [46], and Cascade R-CNN [47]. The output of both one-stage and two-stage detectors usually consists of a set of bounding boxes, labels, and corresponding confidence scores. A bounding box is typically represented as (x, y, h, w), where (x, y) denotes the coordinates of the top-left corner of the box, and h and w indicate its height and width, respectively.

C. Privacy-aware Management in RVCs

Existing privacy-aware management methods for RVCs can be broadly categorized into two types: post-processing and prevention-at-source. We compare the strengths and limitations of these approaches in Table I. Post-processing methods typically apply various masking techniques to obscure privacy-sensitive content in the images or videos. Hu et al. [13] proposed adversarial makeup transfer generative adversarial networks to synthesize adversarial face images with makeup transferred from reference images. Davis et al. [14] proposed a time-domain mixed signal circuit architecture facilitating in-situ privacy with hardware-secured data reversibility. However, these methods only take effect after data collection, meaning that sensitive information may still be temporarily stored on the device, increasing the risk of privacy exposure.

To prevent privacy leakage at the source, some RVC manufacturers have developed various products for privacy management. These solutions can be further divided into digital and physical approaches. Digital methods usually rely on mobile apps to define restricted zones [19], [20], [23], [24]. Although these approaches offer flexibility, they often require root access, which is not user-friendly for the elderly or children. Moreover, an attacker could potentially hack the RVC to learn the location of sensitive areas, resulting in privacy leakage. Physical methods include magnetic strips and infrared emitters. Magnetic strips [25], [26] are attached to the floor to guide the

RVC around restricted areas, while infrared emitters [22], [27] create virtual walls by projecting invisible signals that block the RVC's movement. Since these approaches are hardware-based and independent of software, they naturally offer stronger resistance to cyberattacks. However, they are not compatible with camera-only RVCs, which limits their applicability to various RVCs. In addition, existing methods typically lack semantic information in the physical world, making it difficult for elderly or young family members to understand which areas are restricted.

D. Adversarial Attacks in Robots

Existing detection and classification systems in robots are vulnerable to carefully crafted adversarial examples. Research on adversarial examples for robots has primarily focused on two domains: the digital domain and the physical world. Initially, most studies concentrated on the digital domain, where adversarial examples were crafted by introducing perturbations to the information in a robot's perception system [48], [49]. However, these approaches often require direct access to robotic systems to inject perturbations or noise. Due to the closed nature of robotic systems and restrictions on hardware access, such intrusions are both technically complex and resource-intensive. In contrast, recent research has shifted its focus to physical adversarial attacks, which do not rely on system infiltration and instead target the robot's perception in the physical world. This shift aligns closely with the focus of our study.

Physical adversarial attacks. Unlike digital adversarial examples, which require system infiltration [48], [49], physical adversarial examples manipulate the environment to mislead a robot's perception. Physical adversarial attacks have been extensively studied in areas such as traffic sign recognition, where carefully crafted perturbations on road signs can cause misclassification, potentially leading to dangerous behavior in autonomous driving systems [50]–[52]. These attacks are designed to remain effective under physical constraints such as varying lighting conditions, viewing angles, and distances. These attack strategies are equally applicable to RVCs, especially given their ground-level camera perspectives.

Motivated by privacy protection and ease of use, physical adversarial patches allow users to define "soft boundaries" for RVCs without additional hardware [53], [54]. By visually interfering with RVCs' perception systems, these patches prevent robots from entering restricted areas while minimizing user intervention. Designing robust adversarial patches for RVCs poses challenges. Low-angle, ground-level views necessitate effective perspective transformations [30], [55], and varying lighting or floor textures demand adaptable designs [56], [57]. Solutions such as perspective transformation [58] and robustness-enhancing techniques [53], [59] have been proposed to improve recognition under diverse conditions. Furthermore, integrating perturbations ensures a robot's sensors consistently misinterpret specific regions as obstacles [60], [61]. Extensive tests demonstrate that well-designed adversarial patches effectively interfere with object detection models [62], [63]. However, future designs for RVCs must balance effectiveness, robustness, and user practicality to ensure seamless integration into home environments [64], [65].

III. Threat Model

We assume that RVCs are capable of autonomous navigation within indoor environments. In this scenario, an attacker aims to exploit RVCs to gather private information, while the defender's goal is to address the root cause of privacy leakage and prevent RVCs from accessing sensitive areas.

For an **attacker**, we define the following capabilities:

- *Remote access to sensors of RVCs.* Attacks can remotely exploit vulnerabilities in RVCs to gain access to sensors like cameras and LiDAR. For example, the Bluetooth connector [9] allows access to the Ecovacs X2 from over 100 meters away, and the PIN code system [66] has been found to be faulty. These security vulnerabilities could explain why attackers can utilize RVCs to spy on victims [7]–[9], [67].
- *Behavior concealment.* Attackers can hide their malicious actions by remotely disabling the warning sound that alerts users when the camera is accessed [9], [68], ensuring their activities remain unnoticed by victims.
- *Operational limitations.* Current rooting methods [69]–[72] usually require disassembling RVCs, which is impractical. Therefore, we assume that attackers cannot physically access the device for hardware modifications or obtain root privileges. As a result, they cannot completely bypass user control or access unauthorized areas.

For a **defender**, we assume the following capabilities:

- *White-box access to detection models.* Similar to most adversarial attacks [48], [49], [73]–[77], we assume that a defender has white-box level access to the detection models used in RVCs, including its structure and weights to the degree that the defender can both compute outputs and gradients. Moreover, even in black-box settings [78]–[80] where model details are unknown, a defender can still leverage the transferability of adversarial examples to craft perturbations using shadow models. As we demonstrate in Section V-F, these transferable adversarial patches can effectively prevent RVCs from entering privacy-sensitive areas, thereby making the method feasible under both white-box and black-box assumptions.
- *No direct access to an RVC.* A defender has no direct access to the physical or digital environment of the target RVC. She cannot modify the camera or obstacle detection system and does not have root access. Moreover, the target RVC operates as a closed system with no access for a defender to perform any operations inside it.
- *Physical deployment.* A defender can deploy or remove markings on the ground to affect RVCs, but she cannot physically restrict people from accessing these areas. Since there may be no physical separation between regions, the markings must be non-intrusive and live with human activity. However, as these patches cannot obstruct human movement, embedding semantic information becomes particularly crucial. This allows the markings to be easily recognized by household members, including elderly and children, who may otherwise be unaware of sensitive areas. By incorporating clear semantic cues, such as warnings or instructions, the system ensures

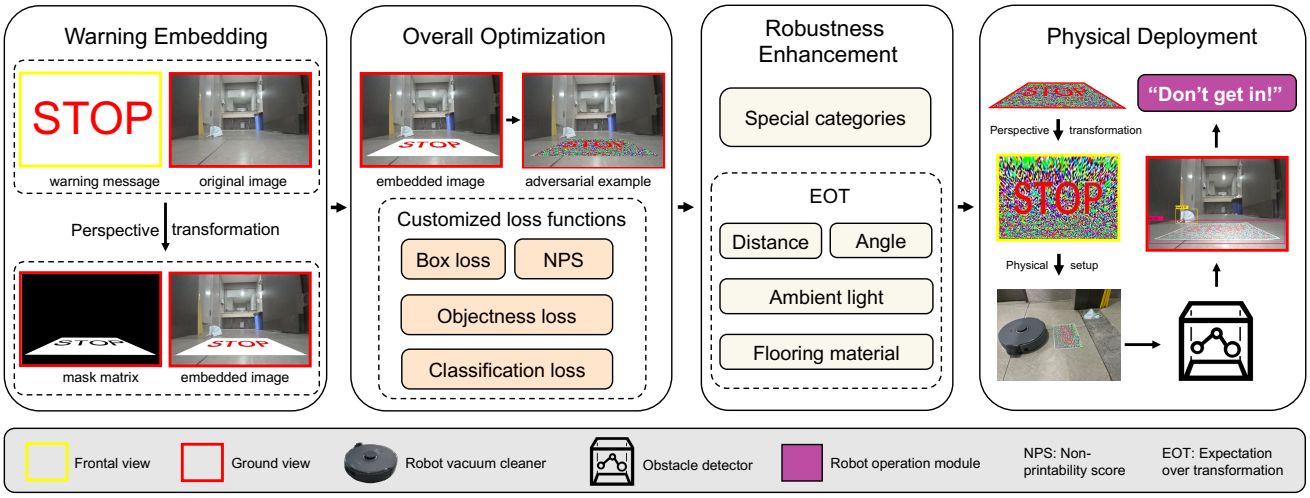


Fig. 3: The workflow of our GPatch. We first design the warning embedding method to embed semantic information into the ground-view image and generate a mask matrix. This enables us to produce adversarial perturbations in a specified region without obstructing the warning information. We then propose custom loss functions to optimize perturbations and enhance the robustness of patches from multiple perspectives. By deploying this approach in the physical world, our GPatch can create virtual obstacles for obstacle detectors, preventing an RVC from entering privacy-sensitive areas.

that sensitive zones are understood by everyone in the household, effectively mitigating the risk of accidental human intrusion while guiding robot behavior.

IV. Methodology

A. Overview

In this paper, we place adversarial patches at regional boundaries, which an RVC recognizes as obstacles, preventing it from entering restricted privacy-sensitive areas. Unlike adversarial examples generated from a frontal perspective, an RVC's camera operates close to the ground. Generating effective adversarial patches for RVCs requires addressing the following challenges:

Challenge 1: How to add semantic information to adversarial patches?

Challenge 2: How to generate adversarial examples that are effective in the ground view of RVCs?

Challenge 3: How to improve the robustness of adversarial patches in the physical world, e.g., distance, ambient light, and flooring material?

To tackle the above challenges, we propose a ground-view adversarial patch, named GPatch, to implement in the physical world. As shown in Figure 3, we first design the **Warning Embedding** method, which overlays a warning message onto the original image captured by an RVC and constructs a mask matrix to prevent the warning message from being obscured by adversarial perturbations. This method effectively addresses Challenge 1. Second, we propose the **Overall Optimization** techniques with customized box, objectness, classification, and color loss functions to optimize the adversarial perturbations. These adversarial patches work with perspective transformations to effectively address Challenge 2. Third, our **Robustness Enhancement** module considers special obstacle classes and integrates expectation over transformation to boost patch robustness across diverse environments, addressing Challenge 3. Finally, the **Physical Deployment** module applies GPatch in the physical world and effectively causes models to recognize GPatch as an obstacle, thereby preventing robots from entering privacy-sensitive areas. The following subsections provide a detailed explanation of each step.

TABLE II: Notations used in our paper.

Notation	Definition
x	Input image
x'	Embedded image
x'_{adv}	Adversarial example for image x
D	DNN-based detector
θ	Parameters of detector D
b	Bounding box of object
y	Label of object
pr	Probability of object
M_A	Mask matrix for area A
p_F	Point on the frontal plane F
p_G	Point on the ground plane G
p_C	Point on the camera coordinates
(u, v, w)	Homogeneous coordinate system
M_H	Homography matrix

B. Problem Formulation

Given an input image $x \in \mathbb{R}^{H \times W}$ contains N objects with each class label $y_i \in [1, 2, \cdots, k]$, a DNN-based detector D: $\mathbb{R}^{H \times W} \rightarrow \mathbb{R}^d$ is trained to derive outputs bounding boxes, class labels, and their confidence scores. The general formulation for the detector's output can be expressed as::

$$D(x) = \{(b_i, s_i, y_i, pr_i)\}_{i=1}^{N} \quad (1)$$

where b_i represents the bounding box of the i-th object, s_i is the confidence score of b_i, y_i means the label of the i-th object, and pr_i is the probability of the i-th object. The goal of our proposed method is to add perturbations to the area

A through the mask matrix M_A to produce an adversarial example $x_{adv} = x + \delta \cdot M_A$, which causes the model D to recognize virtual obstacles:

$$D(x_{adv}) = \{(b'_i, s'_i, y'_i, pr'_i)\}_{i=1}^{N_{adv}} \tag{2}$$

where N_{adv} is the number of detected objects, which is typically larger than N as the patch may cause the detector to identify forged obstacles. Table II lists the key notations used throughout this paper.

C. Warning Embedding

Current adversarial examples usually lack semantic information, and directly incorporating such information into images tends to override adversarial perturbations. In addition, the camera of an RVC is typically mounted at a low-angle position near the ground, resulting in a significant difference between the captured image and a frontal view image. To address this, we propose a warning embedding method that embeds the warning message into the low-angle image while preventing it from being overwritten by adversarial perturbations. Figure 4 illustrates the workflow of our warning embedding method. Specifically, we first add a warning message, such as "STOP", at the center of the blank rectangle to get the original frontal patch $p_F \in \mathbb{R}^{H \times W}$, where H and W are the height and width of the original image x. Note that the original image x refers to a picture taken when the RVC approaches a sensitive area and contains the dividing line between normal and sensitive areas. Depending on the position of the warning message A, we construct the corresponding mask matrix M_A to avoid adding perturbations to the area A as follows:

$$M_A(i,j) = \begin{cases} 0, \text{if}(i,j) \in A \\ 1, \text{otherwise} \end{cases} \tag{3}$$

Next, we need to convert the initial patch p_F into a ground view patch p_G and place it at the boundary of the two areas in the original image x. To achieve the above goal, we design a random perspective transformation method in a specified region. Perspective transformation is a geometric operation that uses a matrix to convert an image from one viewpoint to another. For example, as shown in Figure 4, a patch appears as a regular rectangle when viewed from the front, but may become a trapezoid when captured from a ground-level perspective. Perspective transformation simulates or corrects such distortions caused by changes in the viewing angle. Specifically, we first select four vertices in the original patch as initial points:

$$a_F = (0,0), b_F = (W,0), c_F = (0,H), d_F = (W,H) \tag{4}$$

Second, as shown in Figure 4, we randomly label 4 areas near the boundary of regions in the original image x, and 1 vertex is randomly generated for each region, i.e., a_G, b_G, c_G, d_G. The quadrilateral area formed by these 4 points is denoted as A. To transform the frontal view image into a ground-view image, we use the perspective transformation [81] based on the randomly generated points on the ground plane and the four vertices on the frontal plane. As shown in Figure 5, there are three coordinate systems in total, which are the camera coordinate system, frontal coordinate system, and ground coordinate

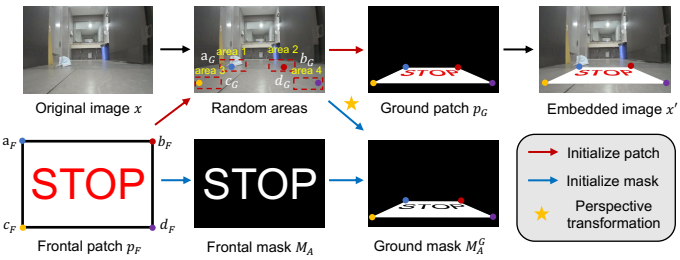


Fig. 4: Illustration of warning embedding method. We begin by generating the warning message on a blank rectangle. Then a random vertex is generated in each of the four designated areas. We use the perspective transform to obtain the ground view image with embedded warning information and the mask matrix for the optimization phase.

system. First, we convert the point $p_F = (X_F, Y_F, Z_F)$ on the frontal plane to camera coordinates $p_C = (X_C, Y_C, Z_C)$:

$$\begin{aligned} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} &= R \begin{bmatrix} X_F \\ Y_F \\ Z_F \end{bmatrix} + T = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_F \\ Y_F \\ Z_F \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_F \\ Y_F \\ Z_F \\ 1 \end{bmatrix} \end{aligned} \tag{5}$$

where R represents the angle of rotation of the $X-Y-Z$ axis, and T is the displacement of the camera's spatial center coordinates from the front spatial center of gravity coordinates. Next, we project the camera coordinates to the ground coordinates:

$$\begin{bmatrix} U_C \\ V_C \end{bmatrix} = \begin{bmatrix} \frac{f}{Z_C} & 0 \\ 0 & \frac{f}{Z_C} \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \end{bmatrix} \tag{6}$$

where f represents the distance from the camera to a plane parallel to the frontal plane extending from the point p_G, i.e., the focal length. As shown in Figure 5, Equation 6 holds because the ground plane G can take advantage of the focal length f and then using similar triangles principle. We use the focal length f that is perpendicular to the frontal plane F, which gives:

$$\frac{f}{Z_C} = \frac{U_C}{X_C}; \quad \frac{f}{Z_C} = \frac{V_C}{Y_C} \tag{7}$$

which are equivalent to Equation 6. However, the coordinates of the point $p_G = (U_G, V_G)$ in the ground plane have to be converted according to the angle ϕ with the plane V. In addition, in the ground plane, the origin is often not the center point, but the lower left corner of the image, so the coordinates (U_C, V_C) with the center of the image as the origin need to be corrected. The coordinates of point p_G in the ground plane (U_G, V_G) are:

$$\begin{bmatrix} U_G \\ V_G \end{bmatrix} = \begin{bmatrix} \frac{U_C}{sin\phi} \\ \frac{V_C}{sin\phi} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} \frac{f}{Z_C sin\phi} & 0 \\ 0 & \frac{f}{Z_C sin\phi} \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \tag{8}$$

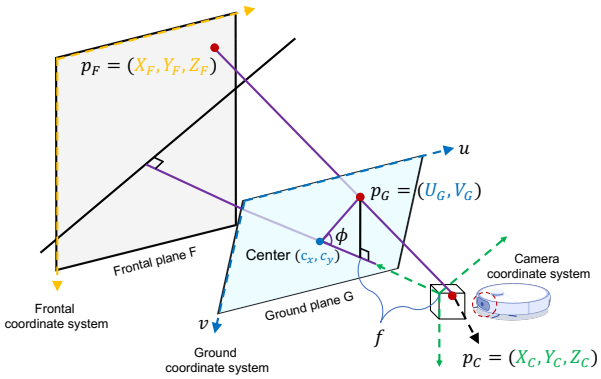


Fig. 5: Diagram of perspective transformations across three coordinate systems.

where (c_x, c_y) is the coordinates of the center of the ground plane. Currently, (U, V) is in units of length (cm), but in images, it is usually calculated in pixels. We introduce two new parameters, ψ and ω, in pixel/cm, meaning the number of pixels per length unit:

$$\begin{bmatrix} U_G \\ V_G \end{bmatrix} = \begin{bmatrix} \frac{f}{Z_C sin\phi}\psi & 0 \\ 0 & \frac{f}{Z_C sin\phi}\omega \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} \frac{f_x}{Z_C sin\phi} X_C + c_x \\ \frac{f_y}{Z_C sin\phi} Y_C + c_y \end{bmatrix} \tag{9}$$

where (f_x, f_y) is the focal length expressed in pixel units.

Our next step is to re-represent the coordinates (U_G, V_G) in terms of the homogeneous coordinate system, which is a simpler representation of space than the Cartesian coordinate system. In contrast, homogeneous coordinates use 3 coordinates (u, v, w) to represent points in space:

$$(u, v, w) \rightarrow (x, y) : \begin{cases} x = \frac{u}{w} \\ y = \frac{v}{w} \end{cases} \tag{10}$$

Note that when $w = 1$, the homogeneous coordinates map nicely to Cartesian coordinates. Then Equation 9 can be expressed in the homogeneous coordinate system as:

$$\begin{aligned} \begin{bmatrix} U_G \\ V_G \\ W_G \end{bmatrix} &= \begin{bmatrix} \frac{f_x}{Z_C sin\phi} & 0 & c_x \\ 0 & \frac{f_y}{Z_C sin\phi} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \\ &= \begin{bmatrix} \frac{f_x}{Z_C sin\phi} & 0 & c_x \\ 0 & \frac{f_y}{Z_C sin\phi} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_F \\ Y_F \\ Z_F \\ 1 \end{bmatrix} \end{aligned} \tag{11}$$

Note that we assume that the image is not distorted after the transformation. However, not all parameters are usually known in the real world, so we can approximate the camera matrix M_C as follows:

$$\begin{aligned} \begin{bmatrix} U_G \\ V_G \\ W_G \end{bmatrix} &= M_C \begin{bmatrix} X_F \\ Y_F \\ Z_F \\ 1 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \begin{bmatrix} X_F \\ Y_F \\ Z_F \\ 1 \end{bmatrix} \\ &= \frac{1}{c_{34}} \cdot M_c \begin{bmatrix} X_F \\ Y_F \\ Z_F \\ 1 \end{bmatrix} = \begin{bmatrix} c'_{11} & c'_{12} & c'_{13} & c'_{14} \\ c'_{21} & c'_{22} & c'_{23} & c'_{24} \\ c'_{31} & c'_{32} & c'_{33} & 1 \end{bmatrix} \begin{bmatrix} X_F \\ Y_F \\ Z_F \\ 1 \end{bmatrix} \end{aligned} \tag{12}$$

where M_C is a 3×4 matrix and c_{34} is usually set to 1. This is because congruent coordinates usually require the matrix to be normalized, and the scale of the coordinate points is negligible. By choosing a flexible coordinate system, we can simplify the camera matrix. Specifically, we locate the X and Y axis in the frontal plane and the Z axis pointing out of the plane. Therefore, the coordinates of all points in the plane on the Z-axis are 0, and the point p_F is $(X_F, Y_F, 0)$. Then the point in the ground plane is:

$$\begin{aligned} \begin{bmatrix} U_G \\ V_G \\ W_G \end{bmatrix} &= \begin{bmatrix} c'_{11} & c'_{12} & \cancel{c'_{13}} & c'_{14} \\ c'_{21} & c'_{22} & \cancel{c'_{23}} & c'_{24} \\ c'_{31} & c'_{32} & \cancel{c'_{33}} & 1 \end{bmatrix} \begin{bmatrix} X_F \\ Y_F \\ \cancel{0} \\ 1 \end{bmatrix} \\ &= M_H \begin{bmatrix} X_F \\ Y_F \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} X_F \\ Y_F \\ 1 \end{bmatrix} \end{aligned} \tag{13}$$

where M_H is the homography matrix with 8 unknown elements. Since the frontal plane F and the ground plane G have corresponding coordinates, all elements of M_H can be approximated by choosing 4 pairs of corresponding points. In general, the points outside the ground-view patch p_G in the output image are filled with 0 pixels. We overlap the initialized ground-view patch p_G with the original image x to get the embedded image x'. Finally, we perform a perspective transformation [81] on the mask matrix M_m using the same homography matrix M_H to obtain the ground-view matrix M_m^G. Note that within the assigned area of the ground view, M_m^G is 0 on the warning message and 1 otherwise, while outside the assigned area it is filled with zeros.

D. Overall Optimization

After obtaining the embedded image x', we need to make the model recognize the ground-view patch p_G as a forged object. However, current adversarial attack methods usually focus only on the classification score during the optimization process, making it challenging to convincingly mimic a large object using a small patch. To address the above issue, we design an overall optimization method with four customized loss functions, which optimize adversarial perturbations to fake large objects:

$$\begin{aligned} \mathcal{L}_{total}(x, b, y; \theta) &= \lambda_1 \mathcal{L}_{box}(x, b; \theta) + \lambda_2 \mathcal{L}_{obj}(x, b; \theta) \\ &+ \lambda_3 \mathcal{L}_{cla}(x, y; \theta) + \lambda_4 \mathcal{L}_{nps}(c_p, c_{print}); \sum_{i=1}^{4} \lambda_i = 1 \end{aligned} \tag{14}$$

where b is the ground truth bounding box, y is the ground truth label, θ is the parameters of the model, and λ_1, λ_2, λ_3, λ_4 are adjustable weights. Each loss in $\mathcal{L}_{total}$ is described in detail next.

Specifically, the box loss $\mathcal{L}_{box}$ measures the difference between the bounding box and the real box based on the Complete Intersection over Union (CIoU) loss [82], and can induce the model to locate a position larger than the contour of the ground-view patch:

$$\mathcal{L}_{box}(x, b; \theta) = \mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\hat{b}, b)}{c^2} + \mu v \quad (15)$$

where IoU is a metric that evaluates the overlap between the predicted and true bounding box, $\rho(\cdot)$ represents the Euclidean distance between the center coordinates of the predicted box $\hat{b}$ and the ground truth box b, c is the diagonal length of the smallest enclosing box covering the two boxes, μ is a positive trade-off parameter, and v measures the consistency of aspect ratio. By setting the ground truth box b to be larger than the actual contour of the patch, we can deceive the model into recognizing the patch as a larger object. Furthermore, the size of b affects the effectiveness of GPatch, which we evaluate in Section V-D.

The objectness loss $\mathcal{L}_{obj}$ is to make the model accurately determine whether each bounding box contains an object. We make the model incorrectly believe that region A contains an object without affecting the decision of other areas:

$$\mathcal{L}_{obj}(x, b; \theta) = \mathcal{L}_{In_A}(x, b; \theta) + \mathcal{L}_{Out_A}(x, b; \theta) \quad (16)$$

when the model outputs a bounding box $\hat{b}$ containing the area A, the objectness loss computes $\mathcal{L}_{In_A}$:

$$\mathcal{L}_{In_A}(x, b; \theta) = -\sum_{i=1}^{I} log(pr_i) \quad (17)$$

where b is the rectangular box containing area A, I is the number of bounding boxes, and $pr_i \in [0, 1]$ is the model's probability for the i-th bounding box. When the model outputs a bounding box b excluding area A, the objectness loss computes $\mathcal{L}_{Out_A}$:

$$\mathcal{L}_{Out_A}(x, b; \theta) = -\sum_{i=1}^{I} [S_i log(pr_i) + (1 - S_i) log(1 - pr_i)] \quad (18)$$

where b represents the location of the real object, S_i indicates whether there is a real object in the i-th prediction box ($S_i = 1$) or not ($S_i = 0$), and pr_i represents the confidence score of the model for the i-th bounding box.

For the prediction results on area A, we design classification loss $\mathcal{L}_{cla}$, which misleads the model to identify the patches into different classes:

$$\mathcal{L}_{cla}(x, y; \theta) = -\sum_{i=1, i \neq y}^{n} log(pr_i) \quad (19)$$

where n is the number of labels, and pr_i represents the probability that the model predicts an object to class i.

Finally, we include the non-printability score (NPS) [83], which represents how closely digital colors match colors printed by a standard printer:

$$\mathcal{L}_{NPS} = \sum_{c_p \in p} min_{c_{print} \in P} |c_p - c_{print}| \quad (20)$$

where c_p represents a color in patch p and c_{print} is a color in the set of printable colors P. This loss penalizes colors that deviate significantly from the printable color set.

According to Section IV-B, given a deep learning model D and the initial image x', the generation of an adversarial sample x'_{adv} with the specified region A can be formulated as:

$$\|x'_{adv} - x'\|_k < \epsilon \quad s.t.\ N_A(D(x'_{adv})) > N_A(D(x')) = 0 \quad (21)$$

where $\|\cdot\|_k$ denotes the distance between two images, $N_A(\cdot)$ indicates the number of detected objects in region A. In this paper, we choose the widely used L_∞ [77], [84], [85] as the difference metric. The ϵ limits the maximum perturbation that can be added. By computing the gradient of $\mathcal{L}_{total}$ with respect to the input, we can add adversarial perturbations to region A with mask M_m^G along the direction of the gradient:

$$x'_{adv} = \mathcal{P}(x' + \alpha \cdot sign(\nabla_{x'} \mathcal{L}_{total}(x', b; \theta)) \cdot M_m^G) \quad (22)$$

where $\mathcal{P}$ projects the perturbed example to an ϵ-radius ball to ensure the perceptual similarity, and α represents the step size in PGD [49]. As shown in Figure 3, the adversarial example in overall optimization adds perturbations only in region A and avoids warning messages.

E. Robustness Enhancement

To improve the robustness of our GPatch in real-world environments, we apply expectation over transformation (EOT) and modify the classification loss function. First, to make our GPatch effective in the physical world, we develop our customized expectation over transformation (EOT) [86] to consider different environmental conditions. Specifically, different input transformations need to be considered during the optimization process to increase the robustness of these examples. These transform distributions are specifically designed for RVCs, including distance, angle, ambient light, and flooring material, as detailed in supplemental materials.

Second, for the prediction results on region A, we modify classification loss $\mathcal{L}_{cla}$ in Equation 19, which misleads the model to identify the patches into some special classes:

$$\mathcal{L}_{cla}(x, y; \theta) = -\sum_{i=1}^{E} log(pr_e) \quad (23)$$

where E is the number of special categories, and pr_e represents the probability that the model predicts an object to class e. The reason for selecting specific categories is that the model's classification is closely related to the bounding box size. We observe that the model exhibits a smaller box loss when classifying larger objects, indicating that the model's output bounding box is closer to our intended large box. As a result, we prioritize larger obstacles, such as shoes, over smaller ones, like batteries. Additionally, certain RVCs maintain a greater distance when cleaning around specific obstacles, such as pet feces, so we designate these obstacles as special categories to prevent the robot from approaching our patches closely. Details can be found in supplemental materials.

F. Physical Deployment

After optimizing GPatch, we deploy it in the physical world. Similar to the approach in Section IV-C, we convert GPatch from the ground view to the frontal view using the perspective transformation. Note that the same homography matrix M_H is not used here. Although the corresponding points remain unchanged, the transformation order differs, requiring recalculation of the homography matrix. Once GPatch for the frontal view is printed, we place it at the entrance of the privacy-sensitive area. When the RVC approaches the entrance, it recognizes GPatch as an obstacle, and the model predicts a bounding box larger than GPatch's actual contour. The obstacle avoidance algorithm determines that the RVC cannot bypass GPatch, causing the RVC's operation module to prevent it from entering the privacy-sensitive area, thus achieving our design goal.

V. Physical Evaluation

In this section, we evaluate and compare the performance of our scheme with other approaches using a real robot vacuum cleaner in the physical world. In addition, we implement ablation experiments to explore the impact of different variables and test the transferability of our designed patches to attack black-box models.

A. Experimental Setup

We present the experimental setup in the physical world.

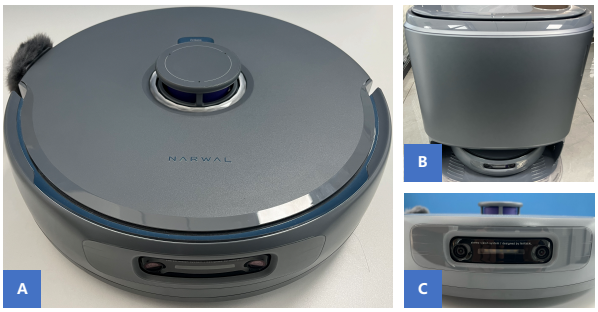


Fig. 6: Experimental setup in the real-world environments. (A) The base station of Narwal robot vacuum cleaner. (B) The robot vacuum cleaner called Narwal Freo Z Ultra. (C) Narwal's built-in dual cameras.

Robot vacuum cleaner. To evaluate the effectiveness of our scheme in the physical world, we use the Narwal Freo Z Ultra, as shown in Figure 6. It is the first RVC to use dual RGB cameras and dual AI chips [87]. Specifically, the Narwal Freo Z Ultra is equipped with dual cameras with 136° wide field of view (FoV), and features 1280×720 photo capabilities. We resize the input images to $(640, 640, 3)$ and then transfer them to a computer equipped with an NVIDIA RTX A6000 GPU to simulate obstacle recognition using different models.

Datasets. To ensure reproducibility in our experiments, we have selected two publicly available datasets focused on RVCs. The robot-cleaner dataset [88] contains 3,396 images featuring 14 common indoor items, including clothes, masks, and toys. The gbgs dataset [89] consists of 2,543 images that can be categorized into a total of 12 distinct categories.

Models. We evaluate six different detection models, categorized into one-stage and two-stage detectors, to assess their performance on the obstacle detection task. A one-stage model detects and classifies objects in one step, making it faster. A two-stage model first generates region proposals and then classifies them, offering higher accuracy but being slower. For one-stage detectors, we employ SSD [42], RetinaNet [90], Yolov5, and Yolov7 [44]. For two-stage detectors, we select Faster R-CNN [91] and Cascade R-CNN [47]. Each model was trained on the datasets until convergence, using the Adam optimizer with a learning rate of 0.001. See supplemental materials for a detailed description of these models.

Metrics. We define the prevention success rate (PSR) as the success rate in preventing the RVC from entering the privacy-sensitive area, i.e., causing the robot to recognize the adversarial patch as an obstacle in area A.

$$PSR = \frac{1}{N}\sum_{i=1}^{N} I_{f(x_i)[box]>0.5 \& f(x_i)[y]\in E}(x) \tag{24}$$

where N is the total number of images, I is the indicator, f represents the detector, box is the confidence score of the predicted bounding box, y is prediction label, and E is the set of special categories in Equation 19. The indicator $I(x)$ equals 1 if the predicted bounding box confidence is greater than 0.5 and the predicted label belongs to the set E, otherwise 0. A higher PSR indicates that the adversarial patch is more effective at creating the illusion of obstacles. This also means that the robot is able to recognize these forged obstacles at the boundary, preventing it from entering sensitive areas.

Baselines. Since we are the first to introduce adversarial patches to prevent RVCs from entering privacy-sensitive areas, we consider two naive methods as baselines:

- *PrintAdv*: Generate adversarial examples using PGD on a specified region of the ground-view image x with our four customized loss functions, and directly print the generated patch for deployment in the physical world.
- *ClassAdv*: Generate adversarial examples using PGD on a specified region of the ground-view image x with only classification loss, apply a perspective transformation to the generated patch, and then print it for deployment in the physical world.

B. Overall Performance

To demonstrate the effectiveness of GPatch, we deploy it in the physical world. Before testing PSR, we train each model on different datasets. Specifically, we randomly select 2,717 images from the robot-cleaner dataset and 2,034 images from the gbgs dataset as the training sets. Additionally, the perturbation is set to 0.6 $\delta/pixel$, and the bounding box decision threshold is set to 0.5. Additionally, we set the PGD perturbation to 0.6 δ/pixel, with a step size of 0.01 and 200 iterations to ensure convergence. The bounding box decision threshold of detection models is set to 0.5. For each model, we generated 110 different GPatch with $210mm \times 297mm$. Each GPatch is placed at the entrance of a sensitive area. We utilize the binocular camera of the Narwal Freo Z Ultra to take three images from the leaf ($-30°$), front ($0°$), and right

TABLE III: The PSR of GPatch across different models.

Model		SSD	RetinaNet	Yolov5	Yolov7	Faster R-CNN	Cascade R-CNN
Dataset	robot-cleaner	86.06%	82.12%	78.18%	92.12%	100%	85.15%
	gbgs	79.09%	76.06%	70.00%	83.94%	100%	80.30%

(30°) sides. As a result, each model is tested with 330 images, which are specifically generated for that model, meaning the images for each model testing are unique.

The experimental results in Table III show the PSR of GPatch on models trained with different datasets. From the table, it is evident that the PSR on the robot-cleaner dataset is slightly higher than on the gbgs dataset. Specifically, the average PSR on the robot-cleaner dataset is 87.27%, while on the gbgs dataset, it is 81.57%. This may be because the robot-cleaner dataset contains a wider variety of categories, making it easier for GPatch to generate perturbations that the model recognizes as a real object. Additionally, the experimental results indicate that GPatch performs better on YOLOv7 than on YOLOv5. This could be because YOLOv7, while having stronger detection capabilities, also becomes more sensitive to perturbations. Notably, when using Faster R-CNN, the PSR of GPatch reached the highest value of 100%. As stated in our experimental setup, SSD, RetinaNet, Yolov5, and Yolov7 are one-stage models, while Faster R-CNN and Cascade R-CNN are two-stage models. The experimental results in Table III show that GPatch achieves an average PSR of 80.95% for one-stage detectors and 91.36% for two-stage detectors. The higher PSR achieved by the two-stage detector may be attributed to the region proposals, which enhances its sensitivity to adversarial perturbations. Overall, GPatch achieves a maximum PSR of 100%, with an average PSR of 84.42% across all six models, demonstrating its effectiveness when deployed in the physical world.

C. Comparison with Baselines

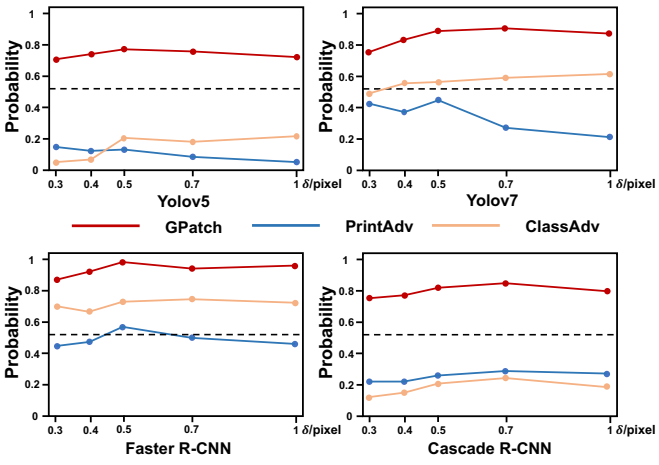
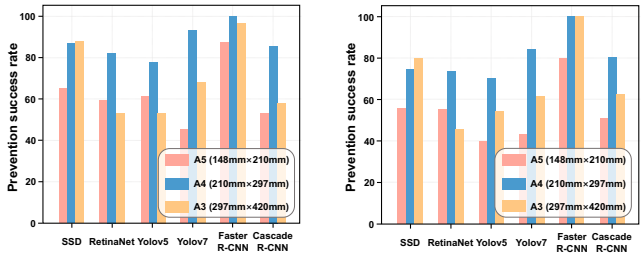


Fig. 7: Detection probabilities of different models for different patch generation schemes.

Similar to the setup in Section V-B, we place the Narwal Freo Z Ultra 20 cm away from the entrance of the sensitive area and capture images of approaching the area through its



(a) Models trained on robot-cleaner (b) Models trained on gbgs

Fig. 8: The PSR of GPatch with different patch size.

cameras. To thoroughly assess the impact of GPatch on the detectors, we consider two additional baseline methods, i.e., *PrintAdv* and *ClassAdv*. It provides a clearer understanding of the advantages of GPatch over other methods in real-world environments. As shown in Figure 7, we evaluate how these schemes affect the probability of multiple detectors under various perturbations. In the best case, GPatch causes the Faster R-CNN model to recognize an object with a probability of 99.70%. For these four models, GPatch achieves an average recognition probability of 87.86%.

The experimental results in Figure 7 show that the detection probability of GPatch usually increases with increasing perturbation, but perturbations above 0.7 usually lead to a decrease in probability. Compared to these baseline methods, GPatch significantly improves the model's score for the patches, indicating that GPatch does not severely degrade the adversarial perturbation during the perspective shift. As shown in Figure 7, the average probability of *ClassAdv* is higher than that of *PrintAdv*. This is because *PrintAdv* directly prints the adversarial examples from the ground view, which compresses the image features further, especially the vertical features. In contrast, *ClassAdv* uses a perspective transformation to adjust for the viewpoint change, improving the recognition probability. However, since *ClassAdv* only considers the classification loss when optimizing the perturbation, the probability is still lower than the threshold of 0.5, causing the model to ignore such patches, rendering it ineffective in preventing the RVC from entering sensitive areas. It is also worth noting that neither *PrintAdv* nor *ClassAdv* significantly improves the model's classification probability as the perturbation increases. This suggests that in real-world deployments, both methods severely undermine against perturbations, failing to achieve the desired results even with large perturbations.

D. Ablation Study

In this section, we examine several possible factors that affect GPatch, including the size and shape of GPatch, special

TABLE IV: The PSR of GPatch with different shapes.

Shape	Model			
	SSD	Yolov5	Faster R-CNN	Cascade R-CNN
Rectangle	85.15%	79.39%	98.79%	82.73%
Circle	70.30%	49.09%	86.36%	61.52%
Triangle	71.82%	53.94%	82.42%	57.58%

TABLE V: The PSR of GPatch with different classification loss.

Classification loss	Model			
	RetinaNet	Yolov5	Faster R-CNN	Cascade R-CNN
All classes	29.70%	29.09%	57.88%	36.06%
Random classes	23.94%	35.45%	46.67%	30.91%
Special classes	82.73%	77.58%	100%	86.06%

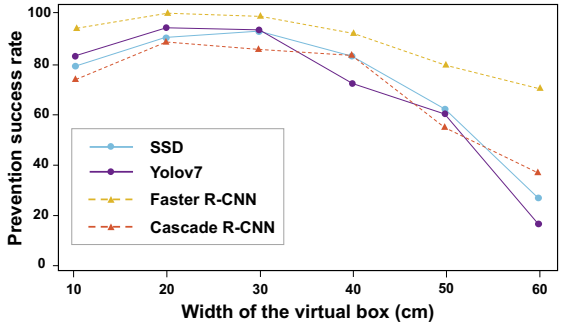


Fig. 9: The PSR of GPatch with different box widths.

categories, the size of virtual boxes, and the length of warning messages.

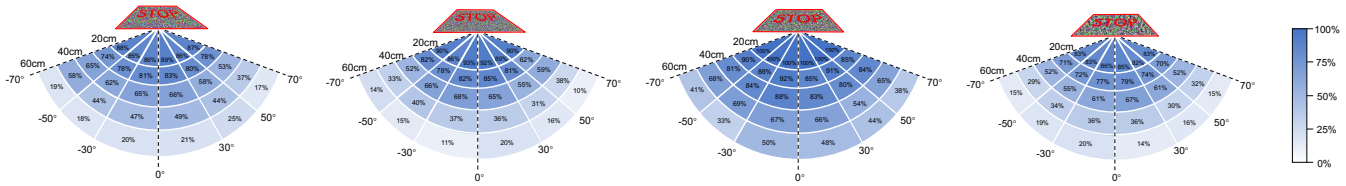
Impact of the size of GPatch. We evaluate the impact of GPatch size on PSR, conducting experiments with models trained on two datasets. Specifically, we examine three different patch sizes, i.e., A3, A4, and A5, and Figure 8 shows the detailed specifications for each size. The experimental results in Figure 8 show that the A4-sized GPatch typically achieves the highest PSR, while the A5-sized GPatch yields the lowest PSR. This is because larger sizes introduce more adversarial perturbations, which increase the probability of model misclassification. However, increasing the size does not necessarily lead to a sustained increase in PSR. Among all six models, only the A3-sized GPatch under the SSD model achieves a higher PSR than the A4-sized GPatch. Moreover, different models demonstrate varying sensitivity to patch size. For instance, on the Yolov7 model trained on robot-cleaner, the PSR difference between A3 and A5 reaches 22.34%, while on the Cascade R-CNN model, the PSR difference for the same size variation is only 4.24%. Therefore, it is recommended to select an A4-sized GPatch, as further increasing the size does not result in a significant improvement in PSR.

Impact of the shape of GPatch. We set up patches with three representative shapes, ensuring that they all have the same area. Specifically, the size of the rectangle is $20.00cm \times 30.00cm$, the circle has a radius of $13.82cm$, and the equilateral triangle has a side length of $37.22cm$. As shown in Table IV, the rectangle consistently achieves the highest PSR across all models. This may be due to the rectangle's contour aligning most closely with the bounding box. In contrast, the circle and triangle exhibit varying performance across different models, suggesting that there is no clear advantage between the circle and triangle. Notably, the circle achieves only 49.09% PSR under the Yolov5 model. Therefore, in the subsequent experiments, we primarily use the rectangle shape for patches to maximize the PSR.

Impact of the special classes. A key characteristic of the RVC is that if the model recognizes an object as a stain (e.g., soy sauce), the cleaner will sweep over it directly instead of avoiding it. Only when the model considers the object identified to be an obstacle that needs to be bypassed, does the vacuum cleaner avoid the corresponding area. Based on this behavior, we generate adversarial examples using different classification losses to verify the effectiveness of the special category set E proposed in Section V-E. An adversarial example is considered ineffective if the model's recognition result does not belong to the obstacle category requiring avoidance. As shown in Table V, "All classes" indicates that the adversarial example attempts to make the model classify it into any category. The number of categories in "Random classes" is equal to that of "Special classes", with half randomly selected from the "Special classes" and the other half drawn from the remaining categories. "Special classes" represents the specific set of categories E in dataset gbgs, i.e., bracelet, earphonewired, earpiece, necklace, ring, USB. The experimental results in Table V show that GPatch achieves the highest PSR across various detectors compared to adversarial examples generated using other classification losses. Additionally, the PSR of "All classes" is generally higher than that of "Random classes", as "All classes" are more likely to be misclassified by the model, leading to a higher probability of being predicted as belonging to the special classes. It is worth noting that the "All classes" results of the Yolov5 model are lower than those of the "Random classes". This may be because the adversarial examples generated by the Yolov5 model are more likely to be misclassified into the special classes of the "Random classes". These findings validate the effectiveness of considering special classes and highlight that, for RVCs, merely causing the model to misclassify is insufficient to achieve the desired adversarial goal.

Impact of the size of virtual boxes. RVCs typically opt to bypass obstacles, as they cannot jump and their vertical clearance is usually limited to less than 5 cm. Consequently, our focus is on the width of the adversarial patches. We place a patch with actual width of $20\,\mathrm{cm}$ at the entrance of a $100\,\mathrm{cm}$ wide area and adjust the width of the virtual bounding box. Different virtual box sizes lead to varying box losses, and we evaluate the PSR of GPatch for different box sizes. If the width of the virtual bounding box exceeds the predicted width of the bounding box by $10\,\mathrm{cm}$, the patch is considered a failure. As shown in Figure 9, the highest PSR is achieved when the width of the virtual box matches the actual width of the patch. However, when the virtual box width exceeds the actual patch width, the PSR begins to decrease. Notably, the most significant drop in PSR occurs when the virtual box width exceeds $50\,\mathrm{cm}$. Additionally, the PSR of the Faster R-CNN model is the highest of the four models. Overall, the width of the virtual box should remain within twice the actual patch width to ensure optimal PSR.

Impact of the length of warning messages. As shown in



(a) SSD (b) Yolov7 (c) Faster R-CNN (d) Cascade R-CNN

Fig. 10: The PSR of GPatch under different distances and angles.

TABLE VI: The PSR of GPatch with different character lengths for warning messages.

Character length (L)	Model				
	RetinaNet	Yolov5	Yolov7	Faster R-CNN	Cascade R-CNN
L≤ 5	83.03%	77.27%	91.82%	99.39%	85.45%
5<L≤ 10	54.24%	50.61%	66.36%	90.61%	57.88%
L>10	36.97%	29.39%	60.91%	79.09%	47.58%

TABLE VII: The PSR of GPatch under different ambient light intensities.

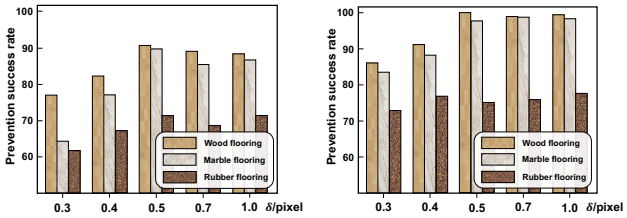
Ambient light	RetinaNet	Yolov5	Faster R-CNN	Cascade R-CNN
500 lux	65.45%	52.73%	80.30%	73.03%
1000 lux	61.21%	47.88%	76.06%	70.00%
2000 lux	70.61%	62.42%	83.94%	76.67%
3000 lux	76.97%	75.76%	88.79%	81.82%
4000 lux	80.30%	77.27%	95.15%	85.15%
5000 lux	83.03%	80.91%	100%	86.67%

Table VI, we evaluate the impact of warning message character length in GPatch on five different models. Each letter uses the same font and font size. The character lengths are divided into three intervals, with detailed examples of the warning messages for each interval provided in supplemental materials. The results in Table VI show that shorter character lengths correspond to higher PSR in GPatch. This occurs because excessive characters compress the space available for adversarial perturbations, reducing their effectiveness. Notably, the impact of character length is most pronounced on the Yolov5 model, where the PSR difference between the shortest and longest warning messages reaches 47.88%. Based on these findings, we recommend embedding warning messages with fewer than five characters in GPatch to minimize the impact of warning message length on PSR.

E. Robustness

To verify the robustness of GPatch in the physical world, we evaluate the success rate of GPatch under different environmental conditions, including ambient light, distance, angle, and ground material. Note that the experimental setup is the same as Section V-B.

Robustness to distance & angle. In real-world environments, RVCs may approach the entrances of sensitive areas from varying distances and angles. Therefore, we evaluate the robustness of GPatch for the four models at various locations. As shown in Figure 10, we categorize the distances into five intervals: [0 cm, 20 cm], [20 cm, 30 cm], [30 cm, 40 cm], [40 cm, 50 cm], and [50 cm, 60 cm]. Similarly, we divide the



(a) Yolov7 (b) Faster R-CNN

Fig. 11: The PSR of GPatch on different flooring materials.

angles into six ranges: [−70°, −50°], [−50°, −30°], [−30°, 0°], [0°, 30°], [30°, 50°], and [50°, 70°]. For each model, we generate GPatch and place it at the entrance. The Narwal Freo Z Ultra takes 220 pictures in each region, which we use for model recognition. From Figure 10, it can be seen that GPatch achieves a PSR of up to 100% within a distance of 20 cm. As the distance increases, the PSR gradually decreases, reaching a minimum of 10% at 60 cm. This decrease occurs because the vertical features of the patch compress with distance, impairing the model's feature extraction capabilities. However, we consider a PSR below 50% acceptable at such distances since the RVC remains too far from the privacy-sensitive area to pose an immediate concern. It is sufficient to ensure that the patch effectively misleads the robot when it gets closer. Angle variations have a minimal impact on PSR at the same distance, demonstrating the effectiveness of the robustness enhancement method described in Section V-E. Notably, the Yolov7 model is more sensitive to angle changes, resulting in more pronounced PSR variations across different angles. In summary, GPatch achieves an average of 88% within 20 cm and shows strong resilience to angle variations, highlighting its practicality and effectiveness in real-world environments.

Robustness to ambient light. We measure the light intensity on the surface of GPatch using a photometer and simulate various light intensities (ranging from 500 lux to 5000 lux) with a desk lamp that has adjustable brightness. During the adjustment of light intensity, we test the PSR of GPatch under four different models, as shown in Table VII. Specifically, changes in light intensity can significantly impact the PSR of the GPatch, which achieves an average PSR of 84.32% at ambient light above 3000 lux. The lowest PSR occurs at a light intensity of 1000 lux, which is due to the fact that images captured in darker environments lose certain details in the darker regions, causing some of the adversarial perturbations to be less effective. In addition, the Narwal Freo Z Ultra automatically activates the illumination in

TABLE VIII: The PSR of GPatch across different detectors.

Transferability		Target Model					
Source Model	Detector	SSD	RetinaNet	Yolov5	Yolov7	Faster R-CNN	Cascade R-CNN
	SSD	86.06%	77.88%	66.97%	80.30%	91.82%	75.45%
	RetinaNet	80.61%	82.12%	59.09%	79.39%	88.48%	68.48%
	Yolov5	56.97%	46.06%	78.18%	83.94%	76.06%	58.79%
	Yolov7	75.15%	60.91%	63.03%	92.12%	94.85%	60.00%
	Faster R-CNN	92.73%	64.24%	78.79%	90.30%	100%	79.70%
	Cascade R-CNN	71.85%	55.15%	66.67%	82.73%	83.94%	85.15%

low-light environments, causing adversarial perturbations to be more effective at 500 lux than at 1000 lux. As the light intensity increases, the PSR of GPatch gradually improves across all four models. However, once the light intensity exceeds 4000 lux, the PSR does not increase significantly anymore. It is worth noting that the average PSR on one-stage detectors listed in Table VII are generally lower than those on two-stage detectors. This is probably because lighting conditions have a greater impact on the global feature extraction performed by one-stage detectors, while two-stage detectors generate candidate regions that focus on local areas of the image. This approach mitigates the impact of lighting variations on the overall image, making it easier to recognize GPatch. In summary, the GPatch demonstrates superior robustness under different lighting conditions.

Robustness to flooring material. Since flooring materials vary from home to home, we evaluate the robustness of GPatch across different types of flooring. Specifically, we select three representative materials: wood flooring, marble flooring, and rubber flooring. It is important to note that carpeting is excluded from this study due to its highly variable patterns, which make consistent testing conditions difficult to achieve. Our objective is to assess the PSR of GPatch on these typical flooring materials. Specifically, we generate GPatch with different levels of perturbation for the Yolov7 and Faster R-CNN models, respectively, and test the GPatch on the three flooring types. The Narwal Freo Z Ultra is positioned 20 cm directly in front of the GPatch, taking 330 pictures in each case for analysis.

The experimental results, illustrated in Figure 11, show that GPatch achieves the highest PSR on wood flooring and the lowest on rubber flooring. This is likely due to the smooth nature of wood, which enhances the visibility of GPatch features under good lighting conditions, thereby improving the model's detection capabilities. In contrast, the granular texture of rubber flooring significantly impairs the feature extraction capabilities of the model, especially for one-stage detectors like Yolov7, resulting in a lower PSR. Furthermore, Figure 11 demonstrates that the GPatch can achieve higher PSR for the Faster R-CNN model than the Yolov7 model under the same perturbation intensity, highlighting the greater robustness of GPatch on two-stage detectors in handling complex textured backgrounds. Further analysis shows that the PSR on rubber flooring does not improve significantly with increased perturbation intensity, verifying that the granular surface interferes with the effectiveness of GPatch's perturbations. In comparison, GPatch performs better on wood or marble flooring than on rubber flooring, with a steady increase in PSR as perturbation intensity grows. Overall, GPatch achieves an average PSR of 90.37% on wood and 87.19% on marble flooring materials, demonstrating its strong robustness and adaptability in diverse home environments.

F. Transferability

To evaluate GPatch 's performance in black-box settings, we evaluate its transferability across different models. We performed white-box attacks on each source model, generating 110 GPatch. Similar to the setup in Section V-B, we capture a total of 330 images, which are resized and used as inputs for target models. Note that all attack parameters (e.g., perturbation size, attack distance) are kept constant throughout the experiment, with the only variable being the target model. Table VIII summarizes the PSR of GPatch across different models. The Faster R-CNN model achieves the highest PSR, both for the source and target models. When YOLOv5 is used as the source model and RetinaNet as the target model, the lowest PSR is obtained at only 46.06%, likely due to the significant architectural differences between the two models. In contrast, when YOLOv7 is used as the source model and Faster RCNN as the target model, the highest PSR is achieved at 94.85%. Furthermore, when the source model is a one-stage detector, GPatch shows a higher PSR on two-stage detectors. This is likely because the region proposal network in two-stage detectors generates candidate regions first, making them more vulnerable to perturbations. In addition, when Cascade R-CNN is used as the source model, GPatch exhibits relatively poor transferability. This may be attributed to the multi-stage filtering mechanism employed by Cascade R-CNN during region proposal generation, which results in a feature space distribution that differs from other models and thus reduces the effectiveness of adversarial perturbations. Future work may explore multi-model training to enhance the transferability across different models, leverage feature alignment to bridge architectural gaps, or employ domain adaptation techniques to reduce transfer domain shift. Overall, GPatch achieves an average PSR of 73.68% in the black box and 87.27% in the white box, indicating that GPatch is more effective in the white box settings.

VI. Discussions

Limitations. Our approach has two limitations in practical applications. First, the virtual bounding box of our patch can't be set infinitely large. Typically, the model generates a bounding box based on the contours of the object. When using a virtual bounding box to simulate a large obstacle, it interferes with the model's bounding box generation. This creates a trade-off between the size of the virtual bounding box and the model's recognition ability. While a larger virtual bounding

box can cover more area, if it becomes too large, it leads to an increased box loss, causing the model to output a bounding box that does not accurately align with the intended virtual box.

Second, our method may not perform optimally in dark environments. In such conditions, RVCs typically rely on fill lighting to improve visibility. However, fill lighting can lead to overexposure, particularly with nearby objects, which may impact the adversarial perturbations of our patches. This means that the designed adversarial patch may not interfere with the robot's recognition system as accurately as expected under fill light conditions. Nonetheless, since nighttime is generally a period when people are resting and robots are less likely to be deployed for cleaning, the impact of fill lighting is relatively limited and poses minimal constraints in practical applications.

Future work. In our future work, we plan to focus on two key directions. First, we aim to develop methods to prevent camera-laser-based RVCs from entering restricted areas while enhancing the generalizability of the approach. Second, we intend to design a privacy-preserving method for other service robots, such as delivery robots and are robots, that is both generalized and easy to implement to prevent robots from violating personal privacy by entering sensitive areas during their tasks. The privacy protection method should be efficient and adaptable to the diverse needs of various robot systems, ensuring that user privacy is maintained without interfering with the robots' normal functions.

VII. Conclusion

In this paper, we propose a novel physical ground-view adversarial patch, GPatch, to address the issue of robot vacuum cleaners entering privacy-sensitive areas. By simply placing GPatch at the entrance of a sensitive area, the robot will misrecognize it as an obstacle and thus avoid entering the area, effectively resolving the privacy leakage problem associated with robot vacuums. We customize the optimization of perturbations in the digital domain, considering perspective transformations and environmental disturbances to ensure successful deployment in the physical world. Experimental validation on several open-source models demonstrates the effectiveness, robustness, and transferability of our approach. This work not only offers a new solution for enhancing the privacy protection of robot vacuums but also provides valuable insights for the real-world application of adversarial patches.

References

[1] G. M. Insights, "Robotic vacuum cleaner market size," 2024, https://www.gminsights.com/industry-analysis/robotic-vacuum-cleaner-market.

[2] A. Bradford, "These robot vacuums have built-in cameras. here's what they can do," 2024, https://www.digitaltrends.com/home/these-robot-vacuums-have-built-in-cameras-heres-what-they-can-do/#dt-heading-even-more-to-come.

[3] PCMag, "Honest, objective, lab-tested reviews," 2024, https://www.pcmag.com/.

[4] IDC, "Worldwide quarterly smart home device tracker," 2024, https://www.idc.com/getdoc.jsp?containerId=IDC_P37480.

[5] ——, "Idc: Global sweeping robot market shipment growth continues in q3, rolling with momentum," 2024, https://www.idc.com/getdoc.jsp?containerId=prCHC52857724.

[6] Roborock, "Roborock official website," 2024, https://global.roborock.com/.

[7] E. Guo, "A roomba recorded a woman on the toilet. how did screenshots end up on facebook?" 2022, https://www.technologyreview.com/2022/12/19/1065306/roomba-irobot-robot-vacuums-artificial-intelligence-training-data-privacy/.

[8] ADGuard, "This sucks: intimate photos taken by robot vacuums were leaked online," 2023, https://adguard.com/en/blog/robot-vaccuum-photo-leak.html.

[9] A. News, "Hackers take control of robot vacuums in multiple cities, yell racial slurs." 2024, https://www.abc.net.au/news/2024-10-11/robot-vacuum-yells-racial-slurs-at-family-after-being-hacked/104445408?utm_campaign=abc_news_web&utm_content=link&utm_medium=content_shared&utm_source=abc_news_web.

[10] ——, "We hacked a robot vacuum and could watch live through its camera." 2024, https://www.abc.net.au/news/2024-10-04/robot-vacuum-hacked-photos-camera-audio/104414020.

[11] Y. Huang, Z. Song, K. Li, and S. Arora, "Instahide: Instance-hiding schemes for private distributed learning," in *in Proceedings of the ICML*, 2020.

[12] J. Deng, J. Guo, X. An, Z. Zhu, and S. Zafeiriou, "Masked face recognition challenge: The insightface track report," in *in Proceedings of the IEEE CVPR*, 2021.

[13] S. Hu, X. Liu, Y. Zhang, M. Li, L. Y. Zhang, H. Jin, and L. Wu, "Protecting facial privacy: Generating adversarial identity masks via style-robust makeup transfer," in *in Proceedings of the IEEE CVPR*, 2022.

[14] S. Davis, J. Liu, B. Cheng, M. Chang, and N. Cao, "In-situ privacy via mixed-signal perturbation and hardware-secure data reversibility," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2024.

[15] Lubluelu, "Robot vacuum boundary strips: Set up cleaning boundaries in your home," 2022, https://lubluelu.com/blogs/buying-guide/robot-vacuum-boundary-strips.

[16] Eufy, "The differences between virtual boundary, no-go zone, and no-mop zone," 2024, https://support.eufy.com/s/article/The-Differences-Between-Virtual-Boundary-No-Go-Zone-and-No-Mop-Zone.

[17] 3i Global, "How robot vacuums avoid certain areas?" 2024, https://us.3itech.com/blogs/news/how-robot-vacuums-avoid-certain-areas.

[18] Ecovacs, "How to block robot vacuum?" 2024, https://www.ecovacs.com/us/blog/how-to-block-robot-vacuum.

[19] C. Allison, "Roomba owners can now set keep out zones for their robot vacuums," 2019, https://www.the-ambient.com/news/irobot-roomba-keep-out-zones-update-1999/.

[20] Roborock, "What are keep-out zones?" 2021, https://homesupport.irobot.com/s/article/21094.

[21] NARWAL, "Magnetic strip invisible wall for t10," 2024, https://us.narwal.com/products/magnetic-stripe.

[22] iRobot, "Dual mode virtual wall barrier," 2024, https://www.irobot.com/en_US/dual-mode-virtual-wall-barrier/4636429.html.

[23] Ecovacs, "How to choose a robot vacuum with no-go zone?" 2024, https://www.ecovacs.com/us/blog/best-robot-vacuum-with-no-go-zones.

[24] Roborock, "Roborock s6 no-go zones and virtual walls tutorial," 2024, https://support.roborock.com/hc/en-us/articles/360036401271-Roborock-S6-No-Go-Zones-and-Virtual-Walls-Tutorial.

[25] Amazon, "Robot vacuum magnetic boundary strip," 2024, https://www.amazon.sg/Tapo-Magnetic-Boundary-Compatible-RVA400/dp/B0C54KMKGK?th=1.

[26] ——, "Magnetic boundary strips," 2024, https://www.amazon.com/s?k=roborock+magnetic+strips.

[27] iRobot, "Dual model virtual wall barrier," 2024, https://www.irobot.com/en_US/dual-mode-virtual-wall-barrier%2C-2-pack/4636426.html.

[28] G. Forni and A. Mantovani, "Covid-19 vaccines: where we stand and challenges ahead," *Cell Death & Differentiation*, 2021.

[29] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*, 2018.

[30] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen, "Seeing isn't believing: Towards more robust adversarial attack against real world object detectors," in *CCS*, 2019.

[31] G. Lovisotto, H. Turner, I. Sluganovic, M. Strohmeier, and I. Martinovic, "Slap: Improving physical adversarial examples with short-lived adversarial perturbations," in *USENIX Security*, 2021.

[32] D. Wang, W. Yao, T. Jiang, C. Li, and X. Chen, "Rfla: A stealthy reflected light adversarial attack in the physical world," in *ICCV*, 2023.

[33] T. Zheng, J. Hu, R. Tan, Y. Zhang, Y. He, and J. Luo, "Physical-world adversarial attack on monocular depth estimation with perspective hijacking," in *USENIX Security*, 2024.

[34] M.-C. Chiu, L.-J. Yeh, and Y. Lin, "The design and application of a robotic vacuum cleaner," *Journal of Information and Optimization Sciences*, vol. 30, no. 1, pp. 39–62, 2009.

[35] F. Vaussard, J. Fink, V. Bauwens, P. Rétornaz, D. Hamel, P. Dillenbourg, and F. Mondada, "Lessons learned from robotic vacuum cleaners entering the home ecosystem," *Robotics and Autonomous Systems*, vol. 62, no. 3, pp. 376–391, 2014.

[36] L. Jianying, W. Yaqian, Z. Yuru, and W. Aokun, "Review of patents analysis on floor robot vacuum," *Recent Patents on Engineering*, vol. 18, no. 4, pp. 135–155, 2024.

[37] T. Asafa, T. Afonja, E. Olaniyan, and H. Alade, "Development of a vacuum cleaner robot," *Alexandria engineering journal*, vol. 57, no. 4, pp. 2911–2920, 2018.

[38] S. Khanna and S. Srivastava, "An empirical evaluation framework for autonomous vacuum cleaners in industrial and commercial settings: A multi-metric approach," *Empirical Quests for Management Essences*, vol. 13, no. 2, pp. 1–21, 2023.

[39] iRobot Customer Care, "How do i use dual mode virtual wall® barrier?" 2023, accessed: 2024-11-09. [Online]. Available: https://homesupport.irobot.com/s/article/9053

[40] R. Global, "Using the magnetic tape for all roborock robot vacuums," 2019, accessed: 2024-11-09. [Online]. Available: https://www.youtube.com/watch?v=mwEHFwtfgxQ

[41] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *CVPR*, 2018.

[42] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.

[43] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, 2018.

[44] B. A. Wang C Y and L. H. Y. M, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *CVPR*, 2023.

[45] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NeurIPS*, 2015.

[46] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[47] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *CVPR*, 2018.

[48] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, 2014.

[49] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *CoRR*, 2017.

[50] A. Zolfi, M. Kravchik, Y. Elovici, and A. Shabtai, "The translucent patch: A physical and universal attack on object detectors," in *CVPR*, 2021.

[51] C. Hu and W. Shi, "Adversarial color film: Effective physical-world attack to dnns," *CoRR*, 2022.

[52] X. Wei, Y. Guo, and J. Yu, "Adversarial sticker: A stealthy attack method in the physical world," *IEEE TPAMI*, 2022.

[53] S. Thys, W. Van Ranst, and T. Goedemé, "Fooling automated surveillance cameras: adversarial patches to attack person detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019, pp. 0–0.

[54] H. Wei, H. Tang, X. Jia, Z. Wang, H. Yu, Z. Li, S. Satoh, L. Van Gool, and Z. Wang, "Physical adversarial attack meets computer vision: A decade survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[55] X. Zeng, C. Liu, Y.-S. Wang, W. Qiu, L. Xie, Y.-W. Tai, C.-K. Tang, and A. L. Yuille, "Adversarial attacks beyond the image space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4302–4311.

[56] D. Karmon, D. Zoran, and Y. Goldberg, "Lavan: Localized and visible adversarial noise," in *International conference on machine learning*. PMLR, 2018, pp. 2507–2515.

[57] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE TEC*, 2019.

[58] J. Kim, A. K. Mishra, R. Limosani, M. Scafuro, N. Cauli, J. Santos-Victor, B. Mazzolai, and F. Cavallo, "Control strategies for cleaning robots in domestic applications: A comprehensive review," *International Journal of Advanced Robotic Systems*, vol. 16, no. 4, p. 1729881419857432, 2019.

[59] C. Xiang, S. Mahloujifar, and P. Mittal, "{PatchCleanser}: Certifiably robust defense against adversarial patches for any image classifier," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2065–2082.

[60] X. Wei, Y. Huang, Y. Sun, and J. Yu, "Unified adversarial patch for visible-infrared cross-modal attacks in the physical world," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[61] J. Liu, A. Levine, C. P. Lau, R. Chellappa, and S. Feizi, "Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 973–14 982.

[62] Y.-C.-T. Hu, B.-H. Kung, D. S. Tan, J.-C. Chen, K.-L. Hua, and W.-H. Cheng, "Naturalistic physical adversarial patch for object detectors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7848–7857.

[63] G. Zhou, H. Gao, P. Chen, J. Liu, J. Dai, J. Han, and R. Li, "Information distribution based defense against physical attacks on object detection," in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2020, pp. 1–6.

[64] S. Hoory, T. Shapira, A. Shabtai, and Y. Elovici, "Dynamic adversarial patch for evading object detection models," *arXiv preprint arXiv:2010.13070*, 2020.

[65] K. Xu, Y. Xiao, Z. Zheng, K. Cai, and R. Nevatia, "Patchzero: Defending against adversarial patch attacks by detecting and zeroing the patch," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4632–4641.

[66] P. Arntz, "Robot vacuum cleaners hacked to spy on, insult owners," 2024, https://www.malwarebytes.com/blog/news/2024/10/robot-vacuum-cleaners-hacked-to-spy-on-insult-owners.

[67] S. Sami, Y. Dai, S. R. X. Tan, N. Roy, and J. Han, "Spying with your robot vacuum cleaner: eavesdropping via lidar sensors," in *SenSys*, 2020.

[68] L. Franceschi-Bicchierai, "Ecovacs home robots can be hacked to spy on their owners, researchers say," 2024, https://techcrunch.com/2024/08/09/ecovacs-home-robots-can-be-hacked-to-spy-on-their-owners-researchers-say/.

[69] D. Giese, "Not all iot devices are created equal: Reverse engineering of xiaomi's iot ecosystem," 2018, https://dontvacuum.me/talks/BeVX-2018/BeVX.html.

[70] ——, "Privacy leaks in smart devices: Extracting data from used smart home devices," 2019, https://dontvacuum.me/talks/DEFCON27-IoT-Village/DEFCON27-IoT-Village_Dennis-Giese-Privacy-leaks-in-smart-devices.html.

[71] ——, "Robots with lasers and cameras (but no security): Liberating your vacuum from the cloud," 2021, https://dontvacuum.me/talks/DEFCON29/DEFCON29-Robots_with_lasers_and_cameras.html.

[72] ——, "Reverse engineering and hacking ecovacs robots - the bad and the really bad," 2024, https://dontvacuum.me/talks/HITCON2024/HITCON-CMT-2024_Ecovacs.html.

[73] M. Zhou, X. Gao, J. Wu, K. Liu, H. Sun, and L. Li, "Investigating white-box attacks for on-device models," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–12.

[74] S. Agnihotri, S. Jung, and M. Keuper, "Cospgd: a unified white-box adversarial attack for pixel-wise prediction tasks," *Proceedings of the ICML*, 2024.

[75] H. Liu, Z. Ge, Z. Zhou, F. Shang, Y. Liu, and L. Jiao, "Gradient correction for white-box adversarial attacks," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[76] C. Zhang, P. Benz, A. Karjauv, J. W. Cho, K. Zhang, and I. S. Kweon, "Investigating top-k white-box and transferable black-box attack," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 15 085–15 094.

[77] Z. Yuan, J. Zhang, Y. Jia, C. Tan, T. Xue, and S. Shan, "Meta gradient adversarial attack," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7748–7757.

[78] D. Ding, M. Zhang, F. Feng, Y. Huang, E. Jiang, and M. Yang, "Black-box adversarial attack on time series classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7358–7368.

[79] F. Yin, Y. Zhang, B. Wu, Y. Feng, J. Zhang, Y. Fan, and Y. Yang, "Generalizable black-box adversarial attack with meta learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 46, no. 3, pp. 1804–1818, 2023.

[80] M. Zheng, X. Yan, Z. Zhu, H. Chen, and B. Wu, "Blackboxbench: A comprehensive benchmark of black-box adversarial attacks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[81] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[82] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. R. D.-I. Loss, "Faster and better learning for bounding box regression., 2020, 34," *DOI: https://doi.org/10.1609/aaai. v34i07*.

[83] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 acm sigsac conference on computer and communications security*, 2016.

[84] J. Zhang, Y. Huang, W. Wu, and M. R. Lyu, "Transferable adversarial attacks on vision transformers with token gradient regularization," in *CVPR*, 2023.

[85] X. Kong and Z. Ge, "Adversarial attacks on regression systems via gradient optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
[86] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *ICML*, 2018.
[87] Narwal, "The narwal freo z ultra robot vacuum," 2024, https://us.narwal.com/pages/freo-z-ultra-robot-vacuum-and-mop.
[88] L. Moon, "Robot cleaner computer vision project," 2023, https://universe.roboflow.com/leo-moon-ekhvq/robot-cleaner.
[89] tst, "gbgs computer vision project," 2023, https://universe.roboflow.com/tst-ewk7h/gbgs.
[90] T.-Y. Ross and G. Dollár, "Focal loss for dense object detection," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
[91] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the NIPS*, 2015.

Shuai Yuan received a B.S. degree in information security from the University of Science and Technology Beijing, and he is currently working toward a Ph.D. degree in cyber security at the University of Electronic Science and Technology of China. His research interests include system security, data security, and privacy-preserving machine learning.

Guowen Xu (IEEE Senior Member) is a Full Professor at the University of Electronic Science and Technology of China (UESTC). His research focuses on cybersecurity, AI security and privacy, and applied cryptography. He received his Ph.D. in Cyberspace Security from UESTC. Before joining UESTC, he has held positions as a Research Fellow at Nanyang Technological University, Singapore, and a Senior Research Fellow at City University of Hong Kong. Prof. Xu has published over 100 papers in top-tier IEEE journals and conferences, including IEEE Transactions on Dependable and Secure Computing (TDSC), IEEE Transactions on Information Forensics and Security (TIFS), IEEE INFOCOM, IEEE S&P, ICML, and NeurIPS. Hiswork has been recognized with multiple awards, including the IEEE BigDataSecurity Best Paper Award (2023), IEEE ICPADS Best Paper Award (2020), Wu Wenjun First Prize of Artificial Intelligence Science and Technology Progress (2021), IEEE Early Career Speaker, IEEE Computer Society (2025), and Computing's Top 30 Early Career Professionals, IEEE Computer Society (2025). Prof. Xu has extensive editorial experience, currently serving as an Associate Editor for IEEE TDSC, IEEE TIFS, IEEE/ACM Transactions on Audio, Speech, and Language Processing, IEEE Transactions on Circuits and Systems for Video Technology, and IEEE Transactions on Network and Service Management. He is also a Lead Guest Editor for ACM Transactions on Autonomous and Adaptive Systems (TAAS). Beyond his editorial roles, he has been an Area Chair or Senior Program Committee Member for prestigious international conferences, including ICML, ICLR, KDD, AAAI, NeurIPS, and CSCW.

Hongwei Li (Fellow, IEEE) is currently the Associate Dean at School of Computer Science and Engineering, University of Electronic Science and Technology of China. He received the Ph.D. degree from University of Electronic Science and Technology of China in June 2008. He worked as a Postdoctoral Fellow at the University of Waterloo from October 2011 to October 2012. His research interests include network security and applied cryptography. Dr. Li has published more than 100 technical papers. He is the sole author of a book, Enabling Secure and Privacy Preserving Communications in Smart Grids (Springer, 2014). Dr. Li serves as the Associate Editors of IEEE Internet of Things Journal, and Peer-to-Peer Networking and Applications, the Guest Editors of IEEE Network, IEEE Internet of Things Journal and IEEE Transactions on Vehicular Technology. He also served the Technical Symposium Co-chairs of IEEE ICC 2022, ACM TURC 2019, IEEE ICCC 2016, IEEE GLOBECOM 2015 and IEEE BigDataService 2015, and Technical Program Committees for many international conferences, such as IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, IEEE WCNC, IEEE SmartGridComm, BODYNETS and IEEE DASC. He won Best Paper Awards from IEEE ICPADS 2020 and IEEE HEALTHCOM 2015. Dr. Li currently serves as the Vice Chair(conference) of IEEE ComSoc CIS-TC. He is the Fellow of IEEE and the Distinguished Lecturer of IEEE Vehicular Technology Society.

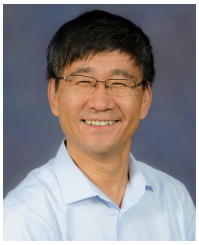
Rui Zhang is currently working toward his master's degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include the intersection of AI and Machine Learning with Security and Privacy, such as machine learning security and secure multi-party computation.

Hangcheng Cao (Member, IEEE) is currently a postdoctoral fellow in the Department of Computer Science, City University of Hong Kong, China. He obtained the Ph.D. degree in the College of Computer Science and Electronic Engineer, Hunan University, China, in 2023. He studied as a joint PhD student in the School of Computer Science and Engineering, Nanyang Technological University, Singapore, in 2022. He has published papers in IEEE S&P, ACM Ubicomp/IMWUT, IEEE INFOCOM, IEEE ICDCS, IEEE TMC, ACM ToSN, IEEE Communications Magazine, Information Fusion, IEEE IoT-J, IEEE JSAS, etc. His research interests lie in the area of IoT security.

Xinyuan Qian is currently a Postdoc at the University of Electronic Science and Technology of China (UESTC) under the supervision of Prof. Hongwei Li. He received his Ph.D. degree from UESTC in June 2025. His research interests include IBE, ABE, FE, FHE, applied cryptography, and privacy-preserving machine learning.

Tao Ni (Member, IEEE) is a postdoctoral researcher at the Department of Computer Science, City University of Hong Kong, where he obtained his PhD. Before that, he received his Master's degree from the Australian National University in 2020 and his Bachelor's degree from Shanghai Jiao Tong University in 2018. His research interests include cyber-physical system security, side channels, and low-power wireless sensing. His works received the Cybersecurity Best Practical Paper Award 2024, and he was also named an ACM MobiSys Rising Star.

Qingchuan Zhao (Member, IEEE) is an Assistant Professor in the Department of Computer Science at the City University of Hong Kong. Prior to joining the department in 2021, he completed his Ph.D. at the Ohio State University in the same year, following his M.S. degree from the University of Florida in 2015, and a B.E. degree from South China University of Technology in 2013. His research focuses on the security and privacy practices in the Android appified ecosystem. He employs both static and dynamic data flow analysis on mobile apps and delves into hardware side channels to uncover a variety of vulnerabilities, including privacy leakage, privilege escalation, and vulnerable access controls. His works received the ACM SIGSOFT Distinguished Paper Award of ICSE'24, and have been granted bug bounties from industry-leading companies and have garnered significant media attention.



Yuguang Fang (Fellow, IEEE) received MS from Qufu Normal University, China, PhD from Case Western Reserve University, USA, and another PhD from Boston University, USA, in 1987, 1994, and 1997, respectively. He joined the Department of Electrical and Computer Engineering at University of Florida in 2000 as an assistant professor, then was promoted to associate professor, full professor, and distinguished professor, in 2003, 2005, and 2019, respectively. Since 2022, he has been a Global STEM Scholar and Chair Professor with Department of Computer Science, City University of Hong Kong. He is the Director of Hong Kong JC STEM Lab of Smart City funded by The Hong Kong Jockey Club Charities Trust. He is a fellow of ACM, IEEE, and AAAS. He received many awards including US NSF CAREER Award, US ONR Young Investigator Award, the 2018 IEEE Vehicular Technology Outstanding Service Award, and IEEE Communications Society awards (AHSN Technical Achievement Award, CISTC Technical Recognition Award, and WTC Recognition Award). He was Editor-in-Chief of IEEE Transactions on Vehicular Technology and IEEE Wireless Communications.