One Patch to Rule Them All: Transforming Static Patches into Dynamic Attacks in the Physical World

Xingshuo Han[1], Chen Ling[2], Shiyi Yao[2], Haozhao Wang[3], Hangcheng Liu[1], Yutong Wu[1],
Shengmin Xu[4], Changhai Ou[2], Xinyi Huang[5], Tianwei Zhang[1]

[1]Nanyang Technological University,[2]Wuhan University, [3]Huazhong University of Science and Technology
[4]Fujian Normal University, [5]Jinan University

{xingshuo001, hangcheng.liu, yutong002, tianwei.zhang}@ntu.edu.sg,{chenling, ysy514, ouchanghai}@whu.edu.cn, {smxu1989, xyhuang81}@gmail.com

ABSTRACT

Numerous methodologies have been designed to generate physical adversarial patches (PAPs) against real-world machine learning systems. In these solutions, each PAP can only achieve a single, fixed attack goal, while switching to a different goal requires the re-generation and re-deployment of a totally new PAP. This rigidity undermines the practicality of these attacks in dynamic and unpredictable environments, such as autonomous driving scenarios, where traffic conditions and attack objectives can change rapidly. For example, if there are no obstacles or barriers around the victim vehicle, the attack will not cause substantial damage to it.

To address this limitation, for the first time, this paper introduces a novel PAP, `SwitchPatch`, which is static but can achieve dynamic and controllable attack consequences based on real-time scenarios. The attacker can manipulate various pre-specified conditions, e.g., projecting different natural color lights onto `SwitchPatch`, to seamlessly switch the attack goals. Unlike existing approaches, `SwitchPatch` does not require patch re-generation or re-deployment for new goals, significantly reducing the attack cost and effort. Additionally, `SwitchPatch` remains benign when attack-enabling conditions are absent, thus its stealthiness is enhanced.

We evaluate the effectiveness of `SwitchPatch` on two popular tasks: traffic sign recognition (including both classification and detection) and depth estimation. First, we perform theoretical analysis and empirical experiments to prove the feasibility of `SwitchPatch`, and identify the number of attack goals `SwitchPatch` can support, particularly when utilizing color light projections and occlusion. Second, we conduct dataset simulation experiments and comprehensive ablation studies to validate the effectiveness and transferability of `SwitchPatch`. Third, we carry out extensive outdoor evaluations using a Unmanned Ground Vehicle (UGV) to prove the robustness of `SwitchPatch` in the physical world. Overall, `SwitchPatch` presents a novel and versatile attack strategy that can be flexibly extended to more specific conditions and additional tasks.

1 INTRODUCTION

Physical adversarial attacks against machine learning based systems have been widely studied in recent years, especially in the form of physical adversarial patches (PAPs) [1]. These patches are carefully crafted perturbations embedded into physical objects, capable of consistently deceiving the target models across diverse environments. Their effectiveness and practical applicability in real-world scenarios make them a critical weapon for physical attacks.

However, existing PAPs [1–11] on various tasks suffer from a fundamental limitation: *each patch is designed only for a single, fixed attack objective*. This presents a major challenge in dynamic and

(a) Benign (no light) (b) HA (green light) (c) MA (yellow light)

Figure 1: (a) `SwitchPatch` is benign to vehicles under normal conditions; (b) `SwitchPatch` causes the hiding attack (HA) when the green light is projected on; (c) `SwitchPatch` causes the misclassification attack (MA), e.g., Stop sign is detected as No Passing, when the yellow light is projected on.

unpredictable environments, such as autonomous driving scenarios, where traffic conditions, environmental factors, and attack objectives can change rapidly and unpredictably. For instance, in traffic sign recognition, an attacker may use an adversarial patch to prevent a victim vehicle from recognizing a stop sign [2, 5]. However, if no obstacles are ahead of the victim vehicle, the patch becomes ineffective, as it fails to create catastrophic outcomes like collisions. Current methods require generating and deploying multiple distinct patches for each objective to dynamically adapt the attack target to the environment, resulting in significant inefficiency and limited flexibility. This highlights the need for a more adaptive PAP approach against real-world applications.

We propose `SwitchPatch`, a novel PAP, which is static but can dynamically adapt to various attack goals in real time. Under normal conditions, `SwitchPatch` stays harmless by default for better stealthiness. The attacker is able to seamlessly switch across various attack objectives by setting certain pre-defined physical conditions. Here we present an example of autonomous driving in Figure 1. The attacker sticks a `SwitchPatch` on a stop sign at the street side of the intersection. This patch is benign to the victim vehicle when there are no surrounding objects that could cause accidents by the misrecognition of the traffic sign. However, when the victim vehicle is in a potentially dangerous scenario, e.g., another vehicle is merging into its lane from the opposite site, the attacker could switch the attack objective into the ⟨⟨hiding attack⟩⟩ by projecting a green light onto `SwitchPatch`. Then the victim vehicle will fail to recognize this stop sign, and an collision could occur. Alternatively, the attacker could project a yellow light onto the same `SwitchPatch` to switch to the ⟨⟨misclassification attack⟩⟩, causing the victim vehicle to recognize the stop sign as "No Passing", and leading to heavy traffic congestion. In general, a single `SwitchPatch` can achieve at least 6 attack goals simultaneously for the traffic sign detection

Table 1: Comparison of related PAPs.

Method	Tasks			Attack Objective		
	C	D	DE	Attack Goal	Goal Switchable	One-time Generation for Goal Switching
RP2 [1]	✓	✓		Benign $\longrightarrow MA$		
AdvLB [3]	✓			Benign $\longrightarrow MA$		
Nested-AE [5]		✓		Benign $\longrightarrow AA$ Benign $\longrightarrow HA$		
Poltergeist [4]		✓		Benign $\longrightarrow AA$ Benign $\longrightarrow HA$ Benign $\longrightarrow MA$		
SLAP [6]		✓		Benign $\longrightarrow HA$		
Phantom [7]		✓		Benign $\longrightarrow AA$	✗	✗
ILR [2]	✓	✓		Benign $\longrightarrow MA$		
TPatch [8]	✓	✓		Benign $\leftrightarrows AA$ Benign $\leftrightarrows HA$ Benign $\leftrightarrows MA$		
Chen et al [9]			✓	Benign $\longrightarrow Far$		
π-attack [11]			✓	Benign $\longrightarrow Near$		
AdvRM [10]			✓	Benign $\longrightarrow Far$ Benign $\longrightarrow Near$		
SwitchPatch (Ours)	✓	✓	✓	Benign $\leftrightarrows HA \leftrightarrows MA \leftrightarrows$ Benign Benign $\leftrightarrows MA_1 \leftrightarrows MA_2...MA_n \leftrightarrows$ Benign Benign $\leftrightarrows Far \leftrightarrows Near \leftrightarrows$ Benign	✓	✓

C: Classification **D**: Detection **DE**: Depth Estimation

(a) Existing PAPs only achieve one specific goal, e.g., *Benign → HA*, or *MA* or *FA*.

(b) SwitchPatch can flexibly and seamlessly switch attack goals, e.g., *Benign ⇆ HA ⇆ MA₁...MAₙ ⇆ Benign*; *Benign ⇆ Far ⇆ Near ⇆ Benign.*

Figure 2: Solid arrow: the patch in normal condition; Dashed arrows: point to the patch given a pre-defined condition.

task (as shown in Figure 5), while all previous works can only achieve one specific attack goal [1–11]. Due to this magical feature, the attacker only needs to generate one patch and deploy it once to achieve different attack goals in an controllable manner, while existing works need to generate different patches. SwitchPatch significantly reduces the attack cost in terms of patch generation, deployment, and control.

There are a couple of non-trivial challenges to achieving the feature of switchable attack objectives. (1) No prior works have explored the feasibility of enabling dynamic attacks through static adversarial patches by modifying external environmental conditions. This raises fundamental questions: Does a single static patch have the capability, and to what extent, to achieve multiple attack objectives under predefined conditions? What is the maximum number of attack goals a static patch can support without compromising its effectiveness? Answering these questions requires both theoretical validation and practical testing. (2) To achieve various attack objectives using one patch, we need to ensure that these pre-defined conditions consistently activate the intended consequence without interference or unintended outcomes. We also need to ensure the patch is robust against environmental noise, such as varying lighting or weather conditions. (3) While remaining cost-effective, SwitchPatch must maintain its stealthiness and practicality for real-world deployment. The physical patch as well as the physical conditions to trigger the goal switch, must be natural and cannot raise humans' suspicions.

To address these challenges, we begin by defining a set of pre-defined conditions and their corresponding attack goals to investigate the existence and capabilities of SwitchPatch. Specifically, we leverage the Weierstrass Extreme Value Theorem [12] to demonstrate the existence and capabilities of SwitchPatch. This analysis also highlights the inherent trade-off between the number of attack goals and the difficulty of finding effective adversarial perturbation for SwitchPatch. Then we conduct extensive experiments on two critical tasks: traffic sign recognition (including both classification and detection) and depth estimation, to corroborate these theoretical findings and demonstrate the practical feasibility of SwitchPatch. To ensure the pre-defined conditions consistently

activate the intended consequences in the physical world, we augment SwitchPatch with color shifting and intensity adjustments during the optimization, which is achieved by harnessing the Expectation over Transformation (EoT) technique [1]. To reduce human suspicion, we first design the joint loss function to make the patch similar to the target object. We then introduce a condition-oriented loss function to ensure SwitchPatch remains benign when the pre-defined conditions are unmet. To comprehensively evaluate SwitchPatch, we conduct both simulation and real-world experiments using a UGV with 3 object detectors and 5 image classifiers for traffic sign recognition, and 4 CNN- or transformer-based models for monocular depth estimation. We evaluate SwitchPatch in both white-box and black-box scenarios to demonstrate its robustness and adaptability to different attack scenarios.

In summary, this paper presents a new contribution to physical adversarial attacks, as the *first* static but switchable PAP that can achieve various attack objectives with pre-defined conditions. We theoretically demonstrate the feasibility of SwitchPatch and validate its practicality through extensive experiments in both simulation and real-world scenarios. Notably, SwitchPatch represents more than just a patch: it embodies an innovative and versatile attack strategy, which is not restricted by any specific conditions or tasks.

2 BACKGROUND

2.1 Physical Adversarial Patch

PAPs can be placed anywhere within the line of sight to confuse machine learning models, causing them to produce incorrect predictions. Eykholt et al. [1] demonstrated how to create robust physical perturbations for stop signs that remain effective under various real-world conditions, such as changes in distances and viewing angles. These perturbations take the form of a poster overlaying the stop sign or a sticker patch applied directly to it. Various adversarial attacks have been proposed to target different computer vision tasks, e.g., traffic sign recognition and depth estimation. Table 1 compares these works with our proposed solution. Specifically, previous PAPs are designed to achieve a fixed attack goal. A new patch needs to be re-generated and re-deployed when the attacker wants to switch to a different goal according to the scenarios (as shown in Figure 2(a), solid line). Contrastively, SwitchPatch uses just one static patch to achieve attack objective switch dynamically. This patch can cause

normal benign effects as well as different attack consequences by changing the physical conditions (Figure 2(b), dotted line). This one-time generation and deployment (Figure 2(b), solid line) can significantly reduce the attack cost and enhance the flexibility.

2.2 Threat Model

We describe our threat model from the attack scenarios, goals, requirements, and adversary's capabilities.

Attack scenario. We follow the previous PAP works [1, 5] to consider the following common scenarios. In traffic sign recognition, the adversary pastes `SwitchPatch` onto a stop sign on the side of the road. This patched sign is benign to all passing vehicles under normal conditions. The adversary can change the attack objective by projecting different light colors onto `SwitchPatch`, causing the victim vehicle to misdetect or misrecognize a stop sign, based on the actual traffic scenarios that can most likely result in accidents. In depth estimation, the adversary attaches `SwitchPatch` to an obstacle, e.g., a stone, and adjusts the attack target based on the actual traffic scenarios, causing the victim vehicle to misestimate the obstacle's depth information, e.g., perceiving it as farther or closer than it actually is.

Attack goal. In classification and object detection tasks, the adversary may select the following goals to activate: (1) Misclassification attack (MA) against traffic sign classifiers/detectors: this causes the model to predict the traffic sign as an incorrect one, e.g., stop sign is classified as speed limit 100. (2) Hiding attack (HA) against traffic sign detectors: this makes the model fail to detect the target. It is worth noting that Appearing attack (AA) is also a common physical attack objective in previous works [2, 8]. However, we do not consider AA in this paper because `SwitchPatch` uses a patch and flashlight to project onto an existing traffic sign. It is more natural to perform AA on an existing object than placing a new traffic sign with a pole (or other objects that can be pasted adversarial patch). HA and MA with many conditions and traffic sign categories are sufficient to lead to serious traffic consequences.

In depth estimation task, the adversary may choose the following goals: (1) Far attack (FA): this increases the estimated depth of an obstacle, which may lead to delayed braking responses and potentially cause collisions with the obstacle. (2) Near attack (NA): this decreases the estimated depth of an obstacle in front, which can result in phantom braking by the vehicle.

Attack requirement. `SwitchPatch` is expected to meet the following requirements:

- *Easy objective switch:* It is effective in aligning with various attack goals controllable by the attacker.
- *Inconspicuousness:* The deployed patch looks natural in the context. It remains benign under normal scenarios. The physical conditions to trigger or switch the attack goals are also natural.
- *Easy deployment:* The attack is cost-effective, and easy to implement and deploy. Additionally, it is robust against various environmental conditions, like weather and lighting.

Adversary capability. We assume the adversary possesses the following capabilities to achieve the attack. He can pre-define multiple attack goals that `SwitchPatch` is designed to induce. Once `SwitchPatch` is generated, he can place it in appropriate locations, e.g., onto existing traffic signs, objects or the road. The adversary

then can leverage pre-defined conditions to decide when to activate or switch the attacks according to the actual physical scenarios.

We consider both the white-box and black-box scenarios. In the former, the adversary knows the target model's details, including its network structure, parameters, and training hyperparameters. The latter is more realistic and challenging, where the adversary has no information about the information of the victim model. We further restrict the attacker's knowledge by assuming that they have no details about the camera used by the victim system, including its brand, resolution, and len.

3 THEORETICAL ANALYSIS

3.1 Problem Formulation

Let X denote the input image space, and Y denote the output of the target model $f : X \rightarrow Y$. An adversarial patch x' is defined as $x + \delta$, where δ is the perturbation applied to x. The patch x' should satisfy the following objectives:

- Under normal conditions, the patch is in the "off" state, and does not exhibit adversarial effect:

$$f(x + \delta) = f(x) = y$$

This is different from prior research where the patch is always effective regardless of the physical conditions.

- The patch is activated to be malicious under certain pre-defined conditions. Specifically, the attacker establishes some pairs of (cl_k, y_k), where cl_k is a special condition and y_k is the goal the attacker wants to induce. This is formulated as:

$$f(x + \delta + cl_k) = y_k$$

By changing the conditions, the attacker can switch the goals in real time, ensuring the behaviors of x' adapt to the dynamical real-world scenarios promptly.

- Stealthiness; x' should be close enough to x to evade human inspections, i.e.,

$$||x' - x||_p \leq \epsilon$$

3.2 Feasibility Proof

Given the above problem formulation, we theoretically analyze the feasibility of finding the solutions from two perspectives: (1) Existence of `SwitchPatch` solutions; (2) Capacity of `SwitchPatch`, i.e., the number of goals to support.

3.2.1 Existence of `SwitchPatch` *Solutions.* We denote the solution space for each attack goal y_k as S_k. To find a perturbation δ^* that satisfies all the attack goals, the following condition must be met:

$$\delta \in S = X_\epsilon \cap S_1 \cap S_2 \cap \cdots \cap S_N$$

where $X_\epsilon = \{\delta \mid ||\delta||_p \leq \epsilon\}$ represents the set of perturbations constrained by ϵ. This problem can also be described as a constrained optimization problem:

$$\delta^* = \arg\min_{\delta \in \Delta} \left(\sum_{k=1}^{N} \mathcal{L}(f(x + \delta + cl_k), y_k) \right) + \mathcal{L}(f(x + \delta), y)$$

where $\Delta = \{\delta \mid ||\delta||_p \leq \epsilon\}$ is the set of perturbations that satisfy the L_p-norm constraint, and $\mathcal{L}$ is a loss function (e.g., cross-entropy).

Based on the Weierstrass Extreme Value Theorem [12], any continuous function on a compact set must attain a maximum and

Figure 3: Overview of `SwitchPatch`. It presents a novel attack strategy as it can be flexibly extended to more pre-defined conditions and applied to more tasks.

minimum value. To apply this theorem, we assume that the loss functions $\mathcal{L}(f(x + \delta), y)$ and $\mathcal{L}(f(x + \delta + cl), y_k)$ are continuous. Additionally, the constraint set $\Delta = \{\delta \mid \|\delta\|_p \le \epsilon\}$ is compact, as it is both bounded and closed. Therefore, there must exist an optimal solution δ^* within the set Δ that satisfies the objective function.

Due to the non-convex nature of the neural network loss function $\mathcal{L}$, directly solving the problem to obtain a global optimum may not always be possible. However, finding a local optimum is still meaningful in the context of adversarial attacks, where a satisfactory solution is often sufficient. We analyze the local optima using the Karush-Kuhn-Tucker (KKT) condition. We define the Lagrange function as:

$$\mathcal{L}(\delta, \lambda) = \sum_{k=1}^{N} \mathcal{L}(f(x + \delta + cl_k), y_k) + \mathcal{L}(f(x + \delta), y) + \lambda(\|\delta\|_p - \epsilon)$$

According to the KKT conditions, there exists a multiplier $\lambda \ge 0$ such that:

$$\nabla_\delta \mathcal{L}(\delta, \lambda) = 0, \quad \|\delta\|_p \le \epsilon, \quad \lambda(\|\delta\|_p - \epsilon) = 0$$

These conditions ensure that even if the global solution is not achievable, a local optimum δ^* that meets the KKT conditions can still be found, providing a practical solution for our problem.

3.2.2 Capacity of `SwitchPatch`. Intuitively, `SwitchPatch` is able to achieve an arbitrary number of attack goals under distinct pre-defined conditions, and its solution space is smaller than that of conventional PAPs for one fixed attack goal. We demonstrate that it is difficult for a stationary PAP to achieve unlimited attack goals: as the number of attack goals increases, the generation of `SwitchPatch` becomes increasingly difficult.

THEOREM 1. *The solution space of `SwitchPatch` decreases as the number N of attack goals y_k increases.*

PROOF. Based on the above definitions, the `SwitchPatch` solution needs to satisfy $\delta \in \mathcal{S} = X_\epsilon \cap \mathcal{S}_1 \cap \ldots \cap \mathcal{S}_N$. Since the solution space $\mathcal{S}_k$ is fixed when the types of conditions cl_k are pre-defined, the size of solution space $\mathcal{S}$ decreases with the increase of N. □

Theorem 1 indicates that optimizing δ becomes more difficult as the number of attack goals $\mathcal{L}_{cl}$ increases, due to the reduced solution space. This is also supported by experiments in Figure 6.

THEOREM 2. *The rate of successfully and simultaneously attacking all objectives decreases as the number N of attack goals y_k increases.*

PROOF. By denoting $\tilde{\mathcal{L}}^N$ as the optima of simultaneously attacking all N objectives and $\tilde{x}_{adv}^{N_s}$ as the optima of simultaneously attacking N_s objectives where $N_s \cap [N]$, we have

$$\tilde{\mathcal{L}}^N = \max\{\mathcal{L}_{cl}^k | k = 1, \ldots, N\} \tag{1}$$
$$\ge \max\{\mathcal{L}_{cl}^k | k = 1, \ldots, N, \text{and } k \ne l_1, \ldots, l_{N_s}\} = \tilde{\mathcal{L}}^{N_s},$$

which completes the proof. □

Theorem 2 indicates that attack success rate monotonically decreases as the number of attack targets increases. Empirical evidence supporting this theorem can be found in Figure 5 across different models like Yolov3 and Faster-RCNN.

4 DETAILED METHODOLOGY

We present the concrete methodology of generating and deploying `SwitchPatch` against different machine learning systems.

4.1 Overview

Figure 3 illustrates the overview of `SwitchPatch`. Specifically, in Step-❶, the generation of `SwitchPatch` involves initialization, condition imposition, and joint loss integration. In the initialization phase, a specific object (e.g., a stop sign or an obstacle) and a random patch are identified to synthesize the initial `SwitchPatch`. Then the attacker can integrate pre-defined conditions to help generate `SwitchPatch`. He can choose light projection with different colors, occlusion of different regions, or other conditions. The attacker can use image processing techniques to simulate these conditions, for example, in the light projection phase, masks of different colors with transparency will be applied to the current `SwitchPatch` to simulate the situation of being illuminated by a flashlight of different colors. The specially designed adversarial loss can be calculated based on the selected light colors, which ensures `SwitchPatch` is capable of achieving the corresponding attack goals via gradient optimization. The enhancement phase further improves the visual stealthiness of `SwitchPatch` by applying meaningful content extracted from a reference image, and enhances the robustness of `SwitchPatch` in the real world by addressing both the deformation of the patch affected by various environmental factors. Step-❷: After the generating, an adversary could easily deploy the attack by printing out the optimized `SwitchPatch` and sticking it on a real-world object. He then can dynamically activate various attack goals by changing pre-defined conditions on the deployed `SwitchPatch`.

Below we provide the specific formulations for different tasks.

- **Object Classification.** An object classification model predicts the categories of a given image x as y. The procedure of generating the PAP x_{adv} (`SwitchPatch`) can be formulated as:

$$x_{adv} = \underset{x' \in X_\epsilon}{\operatorname{argmin}} (\sum_{k=1}^{N} \mathcal{L}(f(x' + cl_k), y_k) + \mathcal{L}(f(x'), y))$$

$$\text{s.t.} \begin{cases} f(x) = f(x_{adv}) = y \\ f(x_{adv} + cl_1) = y_1 \\ \ldots \\ f(x_{adv} + cl_N) = y_N \end{cases} \tag{2}$$

where $x^{'}$ belongs to a set of images X_ϵ that satisfy an L_p norm perturbation constraint (i.e, $||x^{'} - x||_p \leq \epsilon$). N is the number of goals that an attacker can achieve.

- **Object Detection.** An object detection model f extracts features from an image x and outputs its bounding box $y = \{y_{loc}, y_{size}, C\}$ with its localization y_{loc}, size y_{size} and the confidence score C of the categories. Attacking this model can be formulated as:

$$x_{adv} = \underset{x^{'} \in X_\epsilon}{\operatorname{argmin}}(\sum_{k=1}^{N} \mathcal{L}(f(x^{'} + cl_k), \{y_{\text{loc}}, y_{\text{size}}, C_k\})$$
$$+ \mathcal{L}(f(x^{'}), \{y_{\text{loc}}, y_{\text{size}}, C\})$$
$$- \mathcal{L}(f(x^{'} + cl_h), \{y_{\text{loc}}^*, y_{\text{size}}, C\}) \quad (3)$$
$$\text{s.t.} \begin{cases} f(x) = f(x_{adv}) = \{y_{\text{loc}}, y_{\text{size}}, C\} \\ f(x_{adv} + cl_h) = \varnothing \\ \dots \\ f(x_{adv} + cl_N) = \{y_{\text{loc}}, y_{\text{size}}, C_k\} \\ k \neq h \end{cases}$$

where $h \in \{1, ..., N\}$ denotes the h_{th} attack goal as HA. Eq 3 can be segmented into three parts: the first term is designed to achieve k attack goals for misclassification; the second term is associated with achieving the benign effect in the absence of attacker-controlled conditions; the third term induces the hiding attack goal when the h_{th} condition is applied.

- **Depth Estimation.** A depth estimation model f predicts a depth map D for the given RGB image x, where the depth map represents the depth information of each pixel in the input. Generating x_{adv} for depth estimation can be formally expressed as:

$$x_{adv} = \underset{x^{'} \in X_\epsilon}{\operatorname{argmin}} \left(\sum_{k=1}^{N} \mathcal{L}(f(x' + cl_k), D_k) + \mathcal{L}(f(x'), D) \right)$$
$$\text{s.t.} \begin{cases} f(x) & = f(x_{adv}) = D, \quad (4) \\ f(x_{adv} + cl_1) & = D_1, \\ & \dots \\ f(x_{adv} + cl_N) & = D_N. \end{cases}$$

where D_k represents the maliciously perturbed depth map under the k_{th} condition.

4.2 Switchable Attack Objectives

To enable attack goal switch in real time, we design a new loss function for optimizing `SwitchPatch`, as shown below:

$$\underset{\texttt{SwitchPatch}}{\operatorname{argmin}} \mathbb{E}_{x \sim X} \mathcal{L}_{no} + \sum_{k=1}^{N} w_k * \mathcal{L}_{cl}^k + \mathcal{L}_{en} \quad (5)$$

where $\mathcal{L}_{no}$ is the normal loss that makes `SwitchPatch` achieve the benign effects without projections and $\mathcal{L}_{cl}^k$ is the k_{th} adversarial loss in N pre-specific conditions. $\mathcal{L}_{en}$ is the enhancement loss for improving the stealthiness and robustness, which will be detailed in the following sections. The detailed optimization process is described in Alg. 1.

Different tasks may require different formats of the adversarial loss function $\mathcal{L}_{cl}^k$ in Eq 5, as described below.

- **Object Classification.** The attacker only considers MA for the classification task. Hence, the goals are set as $G_s = MA_1; ...; MA_N$,

---

Algorithm 1 `SwitchPatch` Generation.

INPUT: image $x \in X$; labels $y \in Y$; model $f : X \rightarrow Y$; Pre-specific conditions $\{cl_1, cl_2, ..., cl_N\}$; weights for attack goals w_k; weights for stealthiness α, β, and γ; attack iterations *iter*
INPUT: attack goal set: $G_s = \{HA; MA_1; ...; MA_N; FA; NA\}$;
OUTPUT: `SwitchPatch` cp
Initialization: $cp = x + \delta$
1: **for** $t = 0, ..., N_{iter} - 1$ **do**
2:    **if** Classification **then**
3:       use $\mathcal{L}_{cl}^k$ in Eq 6
4:    **else if** Detection **then**
5:       use $\mathcal{L}_{cl}^k$ in Eq 7
6:    **else if** Depth Estimation **then**
7:       use $\mathcal{L}_{cl}^k$ in Eq 8
8:    **end if**
9:    Calculate loss $\mathcal{L}$ in Eq 12
10:   Implement Adam optimizer to calculate patch gradient
11:   $grad = Adam(cp, \mathcal{L})$
12:   $cp \leftarrow cp + grad$
13: **end for**
14: **return** `SwitchPatch` cp

---

where they all can be optimized with the cross-entropy loss:

$$\mathcal{L}_{cl}^k = CELoss(f(x^{'} + cl_k), y_k) \quad (6)$$

- **Object Detection.** The attacker can choose HA or MA to target the object detection task. There could be two strategies to establish the goal set: (1) the attacker can adopt the same set as classification: $G_{s1} = \{MA_1; ...; MA_N\}$. (2) The attacker can combine HA and MA in the goal set: $G_{s2} = \{HA; MA_1; ...; MA_{N-1}\}$. Correspondingly, the loss term $\mathcal{L}_{cl}^k$ can be represented as:

$$\mathcal{L}_{cl}^k = \begin{cases} \mathcal{L}_{HA} + CELoss(f(x^{'} + cl_k), y_k), & \text{if } G_{s2} \\ CELoss(f(x^{'} + cl_k), y_k), & \text{otherwise} \end{cases} \quad (7)$$

Where $\mathcal{L}_{HA}$ is the HA loss, widely used in prior works [8, 13].

- **Depth Estimation.** The attacker can choose two attack goals: $G_s = \{FA; NA\}$ The loss term $\mathcal{L}_{cl}^k$ can be represented as:

$$\mathcal{L}_{cl}^k = CELoss(f(x^{'} + cl_k), D_k) \quad (8)$$

4.3 Robustness Enhancement

To preserve the high attack effectiveness in the physical world, it is ideal that `SwitchPatch` can continuously realize all the target goals or stay benign under different environmental conditions. However, it is challenging to directly apply `SwitchPatch` generated in the digital domain to the physical world, due to the influence of unpredictable environmental conditions.

To address this challenge, we adopt the Expectation over Transformation (EoT) technique, which augments the optimization of `SwitchPatch` with random transformations to overcome the environmental factors in the real world. Specifically, we augment `SwitchPatch` in the following dimensions: translation, rotation, resizing, color shifting, and variations in colored light intensity. Translation, rotation, resizing and color shifting are strategies utilized in previous works [6, 8, 14] to enhance the patch robustness

against distance effects and variations in environmental lighting. Additionally, we introduce variations in colored light intensity to complement the patch's use of colored light projections. Below we give some details of adopting these transformations.

Colored light intensity. We apply translation, rotation, resizing, and color shifting with a uniform distribution to ensure the degree of its randomness. For colored light intensity, the effectiveness of colored light from a flashlight can vary significantly with ambient light conditions. For instance, colored light appears more visible at night, while less visible during the day under strong sunlight. To accurately simulate these varying conditions in EoT, we integrate colored light intensity variations that reflect different levels of ambient brightness.

We simulate cl_k using a k_{th} mask of different colors. For example, for blue light, the default setting is [0, 0, 255], representing the brightest state under low ambient light conditions. To simulate situations with higher ambient light and lower colored light intensity, we reduce this value to [0, 0, 127], effectively halving the perceived brightness. In our EoT process, we apply a uniform distribution to these RGB values to randomize the degree of light intensity transformation. This ensures that `SwitchPatch` can adapt to a broad range of real-world lighting conditions, thereby enhancing its practical effectiveness and robustness in physical environments.

4.4 Stealthiness Enhancement

Our optimization process has two designs for improving the stealthiness of `SwitchPatch`. First, we solve Eq. 5 by using Project Gradient Descent (PGD) with the L_∞ distance constraint during the gradient update step, which ensures that the per-dimension moving distance for each pixel in x is smaller than ϵ. We can use ϵ to control how similar `SwitchPatch` looks compared to the benign x: a smaller ϵ indicates stealthier `SwitchPatch`. We set ϵ as 0.4 defaultly.

We also introduce the following three loss items for stealthiness enhancement: content loss, smoothness loss, and photorealism regularization loss. Formally, let H denote a pre-trained CNN model used for feature extraction, I_s and I_d represent a source image and a designated style image, respectively. In this paper, I_s denotes `SwitchPatch` which is initialized with I_d.

Content loss $\mathcal{L}_c$. Proposed by style transfer works [15], this term can regularize `SwitchPatch`, encouraging the patch to learn the content and spatial structure of the target image rather than the details. The content loss is also defined based on the extracted features by H:

$$\mathcal{L}_c = \sum_{l \in K} \|H_l(I_s) - H_l(I_d)\|_2^2. \tag{9}$$

Different from L_s, L_c is calculated based on the Euclidean distance between the feature maps of I_s and I_d.

Smoothness loss $\mathcal{L}_{sm}$. This loss item encourages a locally smooth image, which can improve the stealthiness while also increasing the patch robustness [16]. It is defined as:

$$\mathcal{L}_{sm} = \sum_{i,j}((I_s[i, j+1] - I_s[i, j])^2$$
$$+(I_s[i+1, j] - I_s[i, j])^2)^{\frac{1}{2}}, \tag{10}$$

where $I_s[i, j]$ denotes a pixel corresponding to the coordinate (i, j).

Photorealism regularization loss $\mathcal{L}_r$. This loss is proposed in [17] for imposing certain constraints on color transfer, thereby preventing color distortions. It is defined as follows:

$$\mathcal{L}_r = \sum_{c \in \{R,G,B\}} V_c(I_s)^\top \mathcal{M}(I_s) V_c(I_s), \tag{11}$$

where c denotes one channel of RGB, V_c reshapes its input into a shape of $N \times 1$ (N represents the number of pixels in I_s), $\mathcal{M}(I_s) \in \mathbb{R}^{N \times N}$ represents a standard linear system that can minimize a least-square penalty function described in [18].

Finally, the loss function of generating `SwitchPatch` is:

$$\underset{\texttt{SwitchPatch}}{\arg\min} \ \mathbb{E}_{x \sim X, t \sim T}(\mathcal{L}_{no} + \sum_{k=1}^{N} w_k * \mathcal{L}_{cl}^k)+$$
$$\alpha * \mathcal{L}_c + \beta * \mathcal{L}_{sm} + \gamma * \mathcal{L}_r \tag{12}$$

where w_k is the weight for adjusting the loss of different attack goals. In subsequent experiments, unless otherwise stated, different goals share the same w_k value. α, β and γ are the weights to balance different loss components, which are set as 1e-2, 3e-6 and 3e-6, respectively. In addition, t is the random transformation with its corresponding distribution T, which is designed to improve the robustness of `SwitchPatch` in the physical world.

5 EXPERIMENTAL SETUP

5.1 Evaluation Metrics

We consider the following metrics for evaluation. (1) *Benign Accuracy (BA)*. This represents the performance of the `SwitchPatch` on the patched validation set under normal conditions. We use the Accuracy and Mean Average Precision (mAP) for traffic sign classification and detection, respectively. For depth estimation, we define the correct prediction as within 5% of the ground-truth distance of the target obstacle. This value should be as high as possible. (2) *Attack Success Rate (ASR)*. This is the ratio of the number of successful attacks against the machine learning model to the total number of attacks performed. We define the `SwitchPatch` attack to be successful when it achieves the target confrontation effect under various pre-defined conditions while maintaining benign results in the absence of conditions. Note that *our criteria are significantly more strict than all those used in prior studies* [1–11], *as multiple objectives must be met simultaneously for a successful PAP*. (3) *Goal-i ASR (G_i-ASR)*. This means that an attack goal can be achieved with its corresponding condition while maintaining benign results in the absence of conditions. Such a metric is helpful for us to analyze the performance of `SwitchPatch`.

Note that for the detection model, the Intersection over Union (IoU) threshold in mAP is set to 0.5. For depth estimation, the threshold of distance prediction error is set to 14%, which is the default setting in [10]. For a ground-truth depth of 10 meters, this threshold value means that a prediction error of at least 1.4 (10×14%) meters is considered a successful attack. We also test other threshold values for depth estimation in our ablation study.

Table 2: ASR(%) of `SwitchPatch` on two-stage recognition.

Classification	VGG-13	VGG-16	ResNet-50	ResNet-101	Mobilenetv2
BA	95.2	100.0	74.9	70.1	78.2
G_1-ASR (MA1)	82.1	96.7	87.3	80.0	81.8
G_2-ASR (MA2)	80.9	97.3	99.7	100.0	83.6
ASR	70.9	95.9	65.4	62.3	64.1

Table 3: ASR(%) of `SwitchPatch` on one-stage recognition.

Detection	Yolov3	Yolov5	Faster R-CNN
BA	100.0	95.4	100.0
G_1-ASR (MA)	91.8	75.4	85.1
G_2-ASR (HA)	95.6	91.2	90.6
ASR	85.9	71.9	80.3

Table 4: ASR of `SwitchPatch` on color intensity with VGG-16.

	Goal_1, Blue, 1/4	Goal_1, Blue, 2/4	Goal_1, Blue, 3/4	Goal_1, Blue, 4/4
Goal_2, Green, 1/4	14.7	50.8	56.1	63.9
Goal_2, Green, 2/4	20.0	50.7	77.5	74.9
Goal_2, Green, 3/4	40.1	55.3	73.8	80
Goal_2, Green, 4/4	45.5	65.6	84.7	95.9

Table 5: ASR of `SwitchPatch` on color intensity with Yolov3.

	Goal_1, Blue, 1/4	Goal_1, Blue, 2/4	Goal_1, Blue, 3/4	Goal_1, Blue, 4/4
Goal_2, Green, 1/4	0.3	5.9	6.5	11.0
Goal_2, Green, 2/4	5.7	10.2	18.5	23.6
Goal_2, Green, 3/4	15.5	41.1	45.7	55.1
Goal_2, Green, 4/4	37.2	71.6	79.8	84.7

5.2 Pre-defined Condition Selection

The condition can take various forms to activate or switch attacks, as stated in Section 3. In this paper, we mainly consider using natural lighting and occlusion because these are easy for an attacker to implement. Specifically, for different light colors, we consider common ones such as blue, green, and yellow. For occlusion, we focus on different positions, including top, bottom, left, and right. In the simulation experiments, we use the standard color mask as the default color intensity as we mentioned in section 4.3. We also give experiments with different color intensities in the following subsections. These masks are strategically placed to coincide precisely in size and location with that of the target object. Specifically, in the classification, the mask is applied across the entire image. Conversely, for object detection and depth estimation tasks, the mask is confined to the ground truth region associated with the sign or obstacle.

5.3 Setup of Traffic Sign Recognition

Models and Datasets. For classification, we evaluate `SwitchPatch` using VGG-13/16, ResNet-50/101, and Mobilenetv2, which cover models of different depths and architectures. All of them are trained on the GTSRB [19] dataset, which is one of the most popular benchmarks for traffic sign classification. For object detection, we evaluate `SwitchPatch` using three popular object detectors, including both one-stage models Yolov3/v5 and a two-stage model Faster R-CNN. The backbones of the pre-trained models Faster-RCNN, Yolov3, and Yolov5 are ResNet-50, Darknet-53 and CSPDarknet, respectively. All these models are trained on MS COCO dataset [20] for detection.

For traffic sign classification, we utilize the validation set of GTSRB [19] for evaluation. For traffic sign detection, we use the autonomous driving dataset KITTI [21] which consists of images captured from real driving scenarios, to comprehensively understand the attack effectiveness and transferability of `SwitchPatch` in both white-box and black-box settings. We use 2,000 and 5000 images that are unseen in the training of the `SwitchPatch` for testing detection models and classification models, respectively. We select 10 classes for object detectors and 10 classes for image classifiers as the target classes. The classification or detection results of these classes are security-related in the driving scenarios.

Attack Goals. We specify the attack goals as Goal_1 and Goal_2, which are denoted by classifying a stop sign as a No Vehicles sign and a Pedestrians sign in classification, respectively. The Goal_1 and Goal_2 in detection are HA and MA (identify a stop sign as a Traffic Light sign), respectively. In addition, we set Goal_1 with

blue color and Goal_2 with green color for both classification and object detection, respectively. In Section 6.1, we will discover more attack goals to validate the performance of `SwitchPatch`.

5.4 Setup of Depth Estimation

Models and Datasets. We use 4 state-of-the-art models as the target MDE models, including CNN-based models, i.e., Mono2 [22], Mande [23] and ViT-based models, i.e., Midas [24], DeAny [25]. All these models are trained on the KITTI dataset [26] or a hybrid dataset (consisting of various datasets). We randomly selected 2000 images from KITTI as the training set and 1000 images as the testing set for evaluation.

Attack Goals. We specify the attack goals Goal_1 and Goal_2 as NA and FA, respectively. We set Goal_1 with red color and Goal_2 with green color.

6 RESULTS FOR TRAFFIC SIGN RECOGNITION

We validate the effectiveness of `SwitchPatch` against both object detectors and image classifiers on the dataset simulation, where `SwitchPatch` is attached to digital images directly.

6.1 Attack Effectiveness

Overall Performance. Tables 2 and 3 show the overall performance results for traffic sign classification and detection, respectively. We observe that all the models can achieve higher ASRs, which demonstrates the high effectiveness of `SwitchPatch`. We also observe that the ASRs for all the models are lower than G_i-ASR. Obviously, `SwitchPatch` needs to meet the attack targets under multiple lighting conditions. This also demonstrates that attack performance may gradually decrease with more attack goals, which we have proved through theoretical analysis in Section 3.

Impact of the color lights intensity. In the default settings, we utilize [0, 0, 255] and [0, 255, 0] to represent the standard blue and green colors, respectively. In this subsection, we investigate how the color light intensity can affect the attack performance of `SwitchPatch`. Specifically, we gradually decrease the color light intensity by decreasing the RGB values of the mask to simulate the reduction of color intensity, for example, a lighter blue color mask is [0, 0, 126], which represents half the intensity of the blue color. We conduct experiments on different degrees of color intensity, e.g., [0, 0, 63], [0, 0, 126], [0, 0, 189] and [0, 0, 255]. Tables 4 and 5 show the results for classification and detection, respectively. It is clear that as the color intensity increases, the ASR also increases.

20 * 20   70 * 170   120 * 120   180 * 180

(a)

(b)

Figure 4: (a) `SwitchPatch` with different size of patch region; (b) Impact on VGG-16 and Yolov3, respectively.

However, this phenomenon does not align with the experimental results in the physical world, which will be detailed in Section 8.1. **Impact of the size of patch region.** We further study the impact of the size of the patch region for `SwitchPatch`. Specifically, we conduct experiments on Yolov3 by setting 4 different sizes of perturbed areas, where we set the sizes as $width, height$ = [20,20], [70, 170], [120, 120] and [180, 180], respectively. Figure 4(a) provides the visualization of 4 demos. From the results shown in Figure 4(b), we can draw three observations. First, a relatively small patch region, e.g., [20, 20], rarely achieves successful attacks for both the classification model VGG-16 and the detection model Yolov3. Second, as the size of patch regions arises, it is more likely to be wrongly predicted by the detector. A possible reason behind this is that as the patch region expands, the possible solution space will gradually increase, enabling `SwitchPatch` can meet multiple conditions, thereby enhancing its attack effectiveness. Third, we also observe that the ASR for a patch size of [120, 120] (53.2%) is lower than that of for [70 * 170] (60.4%), even though the former has a larger area (14400 > 11900). This suggests that the position of the patch also influences the performance of `SwitchPatch`.

Evaluation with increasing attack goals. We conduct experiments on object detection models to prove our theoretical analysis. Specifically, we extend the experiments with more numbers (N) of attack goals and colors. We randomly select 100 images from the validation set of KITTI, the ASRs are recorded only when `SwitchPatch` can achieve both benign and N attack goals. More detailedly, we randomly combine the colors and goals in each experiment, and we calculate the mean value 8 times. Figure 5 shows the results. We observe that as we prove in Section 3.2, ASR gradually decreases with the increase of attack goals as well as with the increase of light conditions. When the number of goals is 7, the attack success rate is only 4.7%, while when the number of goals is 8, the attack fails.

Analysis: Adversarial examples involve searching for permutations of each pixel value, such that the resulting image with a particular permutation appears to the model to possess features characteristic of a specific class. There may exist many such permutations. However, in existing attack algorithms like PGD, finding a particular permutation through gradient methods can achieve

Figure 5: ASR with increasing attack goals on Yolov3, Yolov5, Faster-RCNN.

Figure 6: Number of iterations for generating an effective `SwitchPatch` on Yolov3, Yolov5, Faster-RCNN over KITTI.

the desired attack goal, such as misclassification. This permutation often represents a local optimum [27]. In Section 3.2, we demonstrate that as the number of attack goals increases, the number of light conditions constraining the generation of `SwitchPatch` also increases. This leads to a reduction in the solution space available to `SwitchPatch`, thereby diminishing the effectiveness of the attack when multiple light conditions must be satisfied. Figure 6 shows the relations between the optimization complexity of finding the desired perturbation and the number of attack goals. It is clear that the number of optimization epochs grows with the increase of attack goals for different models. Additionally, `SwitchPatch`'s reliance on gradient-based attack algorithms predisposes it to converge to local optima, resulting in a performance that often falls short of theoretical capabilities. We further provide the investigation of the impact of color-goal combinations and attack goal weights W_k on `SwitchPatch` in the Appendix A.

6.2 Attack Transferability

In instances where an adversary lacks little prior knowledge of the model architectures utilized in commercially available autonomous vehicles, employing gradient-based optimization techniques on these unknown models proves impractical. However, the possibility remains for the attacker to bypass the target model through the utilization of AE' transferability across comparable `SwitchPatch`, this study executes a surrogate model and conducts attacks on other victim models. During the evaluations, specifically, we fixed the attack goals, color lights, the location of the AE, etc.

We use each model as a surrogate model to generate `SwitchPatch` and test on other victim models for both classification and object detection. Tables 6 and 7 demonstrate the results. We observe that (1) for classification, similar model architectures show higher ASRs

Table 6: Transferability across classification models in simulation.

	VGG-13			VGG-16			Resnet-50			Resnet-101			Mobilenetv2		
	G_1-ASR	G_2-ASR	ASR	G_1-ASR	G_2-ASR	ASR	G_1-ASR	G_2-ASR	ASR	G_1-ASR	G_2-ASR	ASR	G_1-ASR	G_2-ASR	ASR
VGG-13	82.1	80.9	70.9	65.3	48.9	36.5	25.0	44.1	10.7	39.8	60.1	21.7	33.5	20.2	17.8
VGG-16	77.1	60.3	52.2	96.7	97.3	95.9	32.1	33.7	12.5	29.6	30.8	10.9	30.1	22.1	18.9
Resnet-50	53.0	21.7	5.8	20.6	31.1	19.9	87.3	99.7	65.4	59.8	66.6	41.3	50.5	23.8	20.7
Resnet-101	34.0	20.8	10.3	35.6	36.7	22.0	67.9	69.8	42.5	80.0	100.0	62.3	46.6	24.3	23.0
Mobilenetv2	37.6	27.7	19.8	37.4	34.4	23.2	42.4	44.7	36.4	54.3	58.4	47.9	81.8	83.6	64.1

Table 7: Transferability across detection models in simulation.

Source	Yolov3		Yolov5		Faster-RCNN	
Target	Yolov5	Faster-RCNN	Yolov3	Faster-RCNN	Yolov3	Yolov5
BA	72.9	52.9	90.2	83.8	97.1	71.3
G_1-ASR (MA)	49.9	41.0	42.5	52.7	41.5	32.3
G_2-ASR (HA)	57.5	82.5	50.0	83.9	47.5	36.7
ASR	46.4	35.5	41.3	34.0	32.1	37.5

while different model architectures show lower ASRs. While for object detection, they show not much difference between different model architectures. (2) The G_i-ASRs of HA are higher than the G_i-ASRs of MA in object detection. The possible reason is that HA aims to reduce the model's confidence in detecting any object within a specific bounding box. Its benefit is that they only require changing permutations to reduce detection confidence without aligning with features of a specific category. This allows a wider range of permutations for image modification, i.e., a larger solution space. On the other hand, MA requires a more restricted solution space that must convincingly mimic another class, i.e., a smaller solution space. This requires more precise and complex permutations that directly align each pixel change with features of the target class, increasing complexity and reducing the likelihood of success. Therefore, even under N constraints, HA still shows higher ASRs than MA.

7 RESULTS FOR DEPTH ESTIMATION

We validate the effectiveness of SwitchPatch against CNN-based and transformer-based depth estimation models. Similar to the traffic sign recognition (Section 6), we attach SwitchPatch directly to digital images for simulation evaluation.

7.1 Attack Performance

Table 8 presents the results for different depth estimation models. We have the following observations. (1) All models maintain high benign performance when SwitchPatch is attached but without pre-defined conditions. This indicates that SwitchPatch has minimal impact on the model performance under normal operating conditions. (2) All models demonstrate high ASRs (from 56.29% to 84.88%) in the white-box scenario, where the attack is generated and evaluated on the same model. This is expected as the attack generation process has complete knowledge of the target models's internal workings. (3) Different models have different degrees of vulnerability. Mono2 exhibits high susceptibility to attacks generated for itself and Mande, suggesting potential similarities in their learned feature representations. Midas, on the other hand, shows the lowest overall vulnerability, indicating a higher level of robustness to these types of attacks. (4) In the black-box scenario,

the transferability of SwitchPatch between models is relatively reduced, particularly when transferring attacks between CNN-based models (Mono2, Mande) and transformer-based models (Midas, DeAny). It demonstrates that the internal representations and decision boundaries learned by different model architectures, especially the fundamental differences between CNNs and Transformers in feature extraction and processing, can significantly impact the effectiveness of SwitchPatch. Figure 11 visualizes the attack effect of SwitchPatch on the KITTI dataset using Mono2, which can effectively implement FA and NA attacks when projecting red and green light, respectively, and remain benign when there is no projection. **Impact of color intensity.** Similar to traffic sign recognition, we evaluate how the color intensity can affect SwitchPatch for depth estimation. Table 9 provides the results, showing that as the color intensity increases, the ASR also increases.

8 PHYSICAL WORLD EVALUATION

8.1 Results for Traffic Sign Recognition

Setup. The experiments are carried out on a closed campus road using an Unmanned Ground Vehicle (UGV), our SwitchPatch, along with a color flashlight, shown in Figure 8. (1) *UGV and cameras.* The UGV is originally equipped with an Intel RealSense D435i front-facing camera. To evaluate SwitchPatch with different cameras, we also install DJI Action 3 and iPhone 11 Pro max in the same position as the Realsense camera to ensure fair comparisons. These cameras' fps are set as 30 and mounted at a height of 1.5 meters on the UGV. The resolutions of these cameras are detailed in Table 13. (2) SwitchPatch is color-printed with a 50cm * 50cm size and positioned at a height of 1.7 meters. (2) *Color flashlight.* The flashlight has a light intensity of 3000LM in normal mode. It offers two color filters, green and orange, and three levels of dimming, including strong light (3000LM), weak light (1500LM), and two intensities of high-frequency flash. It is located 2.3 meters in front of SwitchPatch, with its aperture focussed on and covering SwitchPatch. The flashlight is purchased from Amazon for only $16.75. The UGV, equipped with the cameras, is initially positioned 15 meters away from SwitchPatch and moves straight forward. The initial speed is set to 5m/s (18km/h) at the beginning of each recording. This setup is significant as it aligns with real traffic scenarios. We repeat the experiment for 5 runs.

Dynamic and static evaluations. We categorize physical evaluations into two main modes, i.e., dynamic and static modes. (1) Dynamic mode. It focuses on testing the effect of distance on the attack performance of a vehicle as it moves closer to SwitchPatch. Specifically, in the dynamic mode evaluation, we repeated the movement of the vehicle on the same route three times without colored lights, with green light, and with orange light, respectively. Each time we calculate NA and G_i-ASR in the distance intervals of 3-6m,

Table 8: Transferability across depth estimation models in simulation.

Generation \ Test	Mono2 BA	G_1-ASR	G_2-ASR	ASR	Mande BA	G_1-ASR	G_2-ASR	ASR	Midas BA	G_1-ASR	G_2-ASR	ASR	DeAny BA	G_1-ASR	G_2-ASR	ASR
Mono2	97.3	96.7	85.4	84.8	100	43.5	18.7	18.5	81.2	0	25.3	0	100	12.5	6.2	0
Mande	100	43.6	11.36	6.25	100	80.0	93.7	73.21	100	0	12.05	0	87.5	18.7	10.9	5.8
Midas	99.1	11.7	7.53	6.0	100	12.1	0	0	85.7	82.6	59.9	56.2	86.4	25.6	5.2	5.12
DeAny	86.6	2.5	23.3	0.1	99.2	8.3	20.0	1.5	94.1	5.6	16.3	2.4	99.2	72.0	83.96	65.3

Figure 7: Visualizations on KITTI dataset. From left to right: `SwitchPatch` does not affect depth estimation without light projection; `SwitchPatch` is predicted up to more than 20% farther away than its actual distance when red light is projected; `SwitchPatch` is predicted up to less than 20% closer than its actual distance when green light is projected.

Table 9: ASR of `SwitchPatch` on color intensity with Mono2.

	Goal_1, Blue, 1/4	Goal_1, Blue, 2/4	Goal_1, Blue, 3/4	Goal_1, Blue, 4/4
Goal_2, Green, 1/4	47.6	38.0	48.7	25.8
Goal_2, Green, 2/4	38.4	30.7	43.0	58.6
Goal_2, Green, 3/4	52.3	58.7	54.6	57.2
Goal_2, Green, 4/4	72.4	81.9	75.3	80.8

Table 10: Evaluation under different sunlight conditions in the physical world.

	BA	G_1-ASR (MA)	G_2-ASR (HA)	ASR
Daytime	97.4	1.2	20.3	0.1
Twilight time	83.8	58.8	96.7	48.7
Nighttime	7.0	50.0	73.1	3.6

Table 11: Effectiveness of `SwitchPatch` for traffic sign recognition in the physical world.

	Yolov3	Yolov5	Faster-RCNN
BA	84.4	83.8	93.7
G_1-ASR (MA)	60.2	58.8	69.2
G_2-ASR (HA)	98.9	96.7	96.1
ASR	45.5	48.7	36.7

Static Evaluation Results. We first evaluate `SwitchPatch` in static mode to demonstrate its effectiveness in the physical world. We mainly use object detection models, i.e., Yolov3, Yolov5, and Faster-RCNN, for evaluation. We craft 3 `SwitchPatches` which are generated by these three models, respectively. We use the green color for attack goal_1 (HA) and orange color for attack goal_2 (MA, recognize a stop sign as a "Traffic Light" sign).

Figure 9 visualizes the demos in the physical world. And Table 11 shows the overall attack results. We observe that all the models can achieve effective attack objectives. However, the ASR drops to almost half compared to being on the dataset (Table 3). The possible reason for this is that the color difference between the light and the mask is still quite large, and due to the limitation of the experimental equipment, we are not able to fine-grainedly regulate the intensity of the light.

(1) Impact on sunlight intensity. In general, the color intensity of a flashlight is greatly affected by the intensity of sunlight. To explore the effect of sunlight on the ASR, we placed `SwitchPatch` in the daytime (around 2 pm), twilight time (around 5 pm), and nighttime (around 8 pm). The results shown in Table 10 demonstrate that the ASR of `SwitchPatch` is higher in poor light conditions (e.g., nighttime) than in strong light conditions (e.g., daytime). Intuitively, sunlight intensity affects the effectiveness of SwitchPatch,

Figure 8: Experimental setup in real-world. *Left:* `SwitchPatch` is attached on a stop sign for traffic sign recognition; *Right:* `SwitchPatch` is attached on the back of the vehicle for depth estimation.

6-9m, and 9-15m, respectively. (2) Static mode. We introduce static evaluation because, to measure whether `SwitchPatch` succeeds in attacking in a certain frame, we need to project different colored lights on the same frame at the same moment to ensure that `SwitchPatch` can not be detected or misclassified under each kind of light. However, it is impossible to project different colored lights on the same frame while the vehicle is moving. Even if we repeat the experiment on the same route every time, it is difficult to guarantee whether different lightings are projected on the same frame at the same moment. Specifically, we switch the colorless light and the colored light with different intensities by using the high-frequency flashing mode of the flashlight at 4m and 9m, respectively, and record NA, G_i-ASR, and ASR.

Figure 9: Consecutive frames inference by Yolov5 under high-frequency flashlight of `SwitchPatch` in the physical world. Up row: the light color is orange. The intensity of colored lights from left to right: no projection, weak, strong, weak, strong, no projection. HA can be activated successfully under both weak and strong light intensities while maintaining benign without light projections. Down row: the light color is green. The intensity of colored lights from left to right: no projection, weak, strong, strong, weak, no projection. MA only fails when the light intensity is too strong.

Table 12: Transferability across detection models in the physical world.

Source	Yolov3		Yolov5		Faster-RCNN	
Target	Yolov5	Faster-RCNN	Yolov3	Faster-RCNN	Yolov3	Yolov5
BA	85.4	84.3	76.3	70.8	72.1	81.1
G_1-ASR (MA)	59.8	64.3	56.4	61.9	88.7	82.6
G_2-ASR (HA)	81.2	90.6	94.7	86.4	87.6	83.2
ASR	37.1	32.1	39.8	44.5	45.2	41.1

Table 13: Impact on camera type.

Camera	Resolution	BA	G_1-ASR (MA)	G_2-ASR (HA)	ASR
Realsense D435i	1920 * 1080	83.8	58.8	96.7	48.7
iPhone 11 Pro Max	2688 * 1242	82.1	60.9	90.6	54.4
DJI Action 3	1920 * 1080	80.9	56.1	94.6	50.7

Table 14: ASR(%) under different distances (m) for traffic sign recognition.

Model	Yolov3			Yolov5			Faster-RCNN		
Distance	3-6	6-9	9-15	3-6	6-9	9-15	3-6	6-9	9-15
BA	86.8	85.8	85.2	93.4	93.4	93.0	91.5	89.5	88.7
G_1-ASR (MA)	63.0	57.3	46.6	71.5	65.0	66.3	60.7	68.4	46.3
G_2-ASR (HA)	82.8	75.4	78.1	79.1	88.0	86.8	78.7	82.2	81.0

Table 15: ASR(%) under different distances (m) for depth estimation.

Model	Mono2			Mande			Midas		
Distance	3-6	6-9	9-15	3-6	6-9	9-15	3-6	6-9	9-15
BA	87.5	85.4	83.3	97.9	85.4	81.3	91.6	81.6	79.2
G_1-ASR(NA)	66.7	58.3	52.5	70.8	64.5	60.4	69.3	52.1	31.3
G_2-ASR(FA)	45.8	41.7	35.4	41.7	39.6	33.3	43.7	37.5	27.1

despite that we have considered the effect of color light intensity on `SwitchPatch` during the optimization process, but under strong light conditions, the ASR of `SwitchPatch` is quite low, almost close to 0. We discuss the promising methods to improve the attack performance by combining other attack techniques in Section 9. On the other hand, the ASR of `SwitchPatch` is highest during twilight time because `SwitchPatch` has poor visibility at nighttime.

(2) Attacking different models. Table 12 lists the ASR results for `SwitchPatch` transferred between different object detection models. It is indeed possible to achieve transfer attacks in the real world. For HA, the average of G_2-ASR is around 81.2% to 94.7%, which has a better performance than MA. Among these models, `SwitchPatch` generated by Faster-RCNN shows better transferability than other models, but not that too much. This demonstrates that `SwitchPatch` is slightly affected by the different model architectures.

(3) Attacking different cameras. To further study the impact of the `SwitchPatch` on different cameras, we evaluate the attack effectiveness using Real-sense D435i with 1920 * 1080 resolution, iPhone 11 Pro Max with 2688 * 1242 resolution, and DJI Action 3 with 1920 * 1080 resolution, respectively. Table 13 lists the ASR of `SwitchPatch` on the three cameras, we use Yolov5 as the victim model by default. `SwitchPatch` shows not much difference between these cameras, where the ASRs are around 50%.

Dynamic Evaluation Results. A vehicle is driving on a road equipped with high-resolution cameras and advanced image processing algorithms for object recognition. As it approaches a traffic sign, typically, the sign should appear larger in the vehicle's camera feed as the distance between the vehicle and the sign decreases. As we stated in Section 4.3, in the EoT setting, we do not use the

traditional assumption that the distribution of pixels is uniform, but instead set a larger weight on a smaller pixel size. Table 14 shows the results. We observe that the NA shows not too many differences between these models. However, G_1-ASR (MA) will increase slightly as the distance gets closer.

8.2 Results for Monocular Depth Estimation

Setup. The experiments are carried out in the same location on a closed campus road. Our primary target camera model is the Intel RealSense D435i. `SwitchPatch` is mounted on the rear of a BMW X1. which is our target object, with dimensions of 4.95 m in length, 1.97 m in width, and 1.905 m in height. We use Mono2 as the target monocular depth estimation model. We drive the victim vehicle towards the target vehicle at a distance of 10 meters and record the adversarial scenario while driving.

Note that this attack is generic so it can be applied to any class of objects on public roads. This paper focuses on a static vehicle as shown in Figure 8. We choose vehicles because (1) they are common on public roads in regular driving scenarios; (2) they are commonly used in previous works [9–11]. A failure to detect them could lead to life-threatening consequences; (3) they are the most attractive objects for attackers since they are the main targets of perception systems on an autonomous driving car.

Evaluation methodology. We drive four times on each route, with the first one a benign case, the second one pasted `SwitchPatch`, the third one pasted `SwitchPatch` with green light projection and the fourth one pasted `SwitchPatch` with blue light. The depth (z) of the

Table 16: ASR of `SwitchPatch` on different thresholds with Mono2 in the physical world.

Thresholds	10%	14%	18%	22%	26%
BA	87.5	87.5	87.5	87.5	87.5
G1-ASR(NA)	78.0	66.7	60.3	40.6	32.4
G2-ASR(FA)	53.8	45.8	39.7	30.2	28.1

vehicle can be calculated with $z = fH/s$. So, given the focal length (f) of the camera and the height of the vehicle in the physical world (H) and on the image plane (s), we calculate the vehicle's depth. We use this depth as the vehicle's depth ground truth to calculate E_d. We shot at a constant speed and captured multiple frames from distances ranging from 3m to 15m from the vehicle to simulate different travel spacing. A total of 200 frames were captured.

Impact of distance. We first investigate how the distance can affect `SwitchPatch` for depth estimation. For each distance interval (i.e., 3m), we have 50 frames of images for evaluation. Table 15 provides the results, showing that `SwitchPatch` can achieve high ASRs varying distances for all the models. Figure 11 in Appendix A visualizes of `SwitchPatch` attack in the physical world.

Impact of depth threshold. We then evaluate how much the attack can change the depth, the larger the change in depth value the more harmful it is, in our experiments, we set different depth thresholds as the determination criteria and calculate the success rate of the attack respectively. A depth threshold is a critical value set when evaluating the effectiveness of an adversarial attack, and an attack is considered successful if the average value of the depth change exceeds this threshold. As shown in Table 16, the effect of our attack can change the depth value more than 26%, which reflects the effectiveness of the attack.

9 DISCUSSION

More switch conditions. `SwitchPatch` is the first work to demonstrate that an attacker can dynamically switch various attack targets using a PAP in the physical world, establishing it as a novel approach without an existing baseline for comparison. However, `SwitchPatch` is not limited to leveraging colored light projections to switch attack goals. Attackers can also employ other techniques, such as occluding parts of the adversarial example to achieve switchable attack goals. Specifically, an attacker can optimize a global perturbation for traffic signs and then tailor the optimization process to different occlusion positions, specifying corresponding attack targets. Figure 10 illustrates the demos using occlusion. The attacker can use a cube to occlude not only the upper or left part of the adversarial patch but also other areas.

To demonstrate the feasibility of such attacks, we use a cube to occlude the left/up part as attack goal 1 and the right/down part as attack goal 2, respectively. Experiments are conducted using Yolov3. The results are presented in Table 17; the validation set includes 100 images that are randomly selected from the KITTI dataset. We observe that the colored light projections provide more consistent results than occlusion techniques, which although highly effective in the MA (especially for Left and Right) with significant drops in the HA scenario. We encourage researchers to explore further techniques to enhance the strength and stealthiness of `SwitchPatch` in the future.

Table 17: Different attack techniques.

	Colored lights (Blue and Green)	Occlusion (Up and Down)	Occlusion (Left and Right)
BA	100	96.6	100.0
G_1-ASR (HA)	91.8	95.4	100.0
G_2-ASR (MA)	95.6	5.9	15.6
ASR	85.9	5.2	15.6

Figure 10: Occluding different parts as activation conditions.

Countermeasures. We have used different defense methods, e.g., input preprocessing, including image smoothing [28], feature compression [29] and input randomization [30]; defensive dropout [31] and adversarial training [32], to defend `SwitchPatch` with simple experiments. These defense methods can slightly mitigate ASR in the range of 0 to 23% on Yolov5, which means these methods cannot fundamentally defend `SwitchPatch`. Since `SwitchPatch` is a general attack strategy, designing effective defense techniques that can be applied to various tasks will be our future work.

Inconspicuity comparisons. We state that when conducting `SwitchPatch` attack, the adversary doesn't need to cast the light onto the patch for long (explained in AdvLB[2]), while he aims at subtly and abruptly activating `SwitchPatch`, which gives the vehicle little time to react. Therefore, the light strength doesn't affect `SwitchPatch`'s stealthiness. Besides, compared to [6–8, 14], `SwitchPatch` is the only one that preserves the original traffic sign texture. Although it sacrifices some stealthiness compared to RP2 attack [1], it can achieve multiple attack goals.

Improvement of `SwitchPatch` on daytime. Although the light intensity of the flashlight we used was 3000 lux, the intensity of the colored light projections was not that strong at noon, which limited the effectiveness of `SwitchPatch` under bright daylight conditions. This weak performance during high ambient light exposure could reduce the reliability of the attacks in practical scenarios, especially in outdoor environments.

To address this limitation, we propose several possible solutions. The first is increasing the intensity or using higher luminance colored lights could help maintain the visibility and distinction of the colored projections under strong ambient light. The second is combining light projections with other techniques, such as the occluding method as we stated in Section 9. When the sunlight is strong, the attacker can use occlusion techniques to switch the attack target because occlusion is more clearly visible. However, when the sunlight is weak, the attacker can instead use colored light projections. We hope that researchers can develop techniques that are more robust to environmental factors.

10 CONCLUSION

We introduce `SwitchPatch`, a novel and versatile PAP designed for dynamic and strategic manipulation in real-world environments. `SwitchPatch` is unique in its ability to leverage a diverse

set of pre-defined physical conditions, allowing it to seamlessly adapt its attack objectives based on real-time situational awareness. We demonstrate the effectiveness, adaptability, and robustness of `SwitchPatch` through extensive evaluations across both simulation and real-world scenarios. Our results consistently highlight high attack success rates across a wide range of operational conditions.

REFERENCES

[1] 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.

[2] T. Sato, S. H. Bhupathiraju, M. Clifford, T. Sugawara, Q. A. Chen, and S. Rampazzi, "Invisible Reflections: Leveraging Infrared Laser Reflections to Target Traffic Sign Perception," in NDSS, 2024.

[3] R. Duan, X. Mao, A. K. Qin, Y. Chen, S. Ye, Y. He, and Y. Yang, "Adversarial laser beam: Effective physical-world attack to dnns in a blink," in CVPR, 2021.

[4] X. Ji, Y. Cheng, Y. Zhang, K. Wang, C. Yan, W. Xu, and K. Fu, "Poltergeist: Acoustic adversarial machine learning against cameras and computer vision," in S&P, 2021.

[5] 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 ACM CCS, 2019.

[6] 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.

[7] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici, "Phantom of the adas: Securing advanced driver-assistance systems from split-second phantom attacks," in ACM CCS, 2020.

[8] W. Zhu, X. Ji, Y. Cheng, S. Zhang, and W. Xu, "Tpatch: A triggered physical adversarial patch," in Usenix Security, 2023.

[9] Z. Cheng, J. Liang, H. Choi, G. Tao, Z. Cao, D. Liu, and X. Zhang, "Physical attack on monocular depth estimation with optimal adversarial patches," 2022.

[10] H. Liu, Z. Wu, H. Wang, X. Han, S. Guo, T. Xiang, and T. Zhang, "Beware of road markings: A new adversarial patch attack to monocular depth estimation," in NeurIPS, 2024.

[11] T. Zheng, J. Hu, R. Tan, Y. Zhang, Y. He, and J. Luo, "{π-Jack}:{Physical-World} adversarial attack on monocular depth estimation with perspective hijacking," in USENIX Security, 2024.

[12] J. E. Martínez-Legaz, "On weierstrass extreme value theorem," Optimization letters, 2014.

[13] D. Song, K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, and T. Kohno, "Physical adversarial examples for object detectors," in USENIX workshop, 2018.

[14] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, "Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector," in ECML PKDD, 2019.

[15] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in CVPR, 2016.

[16] 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 CCS, 2016.

[17] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in CVPR, 2017.

[18] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," TPAMI, 2007.

[19] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," Neural networks, 2012.

[20] Microsoft, "Common objects in context (coco) dataset," 2018, https://cocodataset.org/.

[21] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in CVPR, 2012.

[22] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in ICCV, 2019.

[23] J. Watson, O. Mac Aodha, V. Prisacariu, G. Brostow, and M. Firman, "The temporal opportunist: Self-supervised multi-frame monocular depth," in CVPR, 2021.

[24] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," TPAMI, 2022.

[25] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," CoRR, 2024.

[26] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in CVPR, 2012.

[27] Y. Tashiro, Y. Song, and S. Ermon, "Diversity can be transferred: Output diversification for white-and black-box attacks," NeurIPS, 2020.

[28] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in ICML, 2019.

[29] X. Jia, X. Wei, X. Cao, and H. Foroosh, "Comdefend: An efficient image compression model to defend adversarial examples," in CVPR, 2019.

[30] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," CoRR, 2017.

[31] S. Wang, X. Wang, P. Zhao, W. Wen, D. Kaeli, P. Chin, and X. Lin, "Defensive dropout for hardening deep neural networks under adversarial attacks," in ICCAD, 2018.

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

Table 18: ASR(%) of `SwitchPatch` on color-goal combinations with traffic sign recognition.

Models		Green (Goal_1)	Blue (Goal_1)	Orange (Goal_1)	Purple (Goal_1)
VGG-16	Green (Goal_2)	✗	95.9	29.7	45.5
	Blue (Goal_2)	69.2	✗	51.1	33.5
	Orange (Goal_2)	39.8	69.6	✗	51.0
	Purple (Goal_2)	44.0	32.3	56.9	✗
ResNet-50	Green (Goal_2)	✗	65.4	35.8	43.9
	Blue (Goal_2)	55.0	✗	49.1	38.8
	Orange (Goal_2)	41.0	63.1	✗	63.6
	Purple (Goal_2)	32.5	55.8	65.3	✗
Mobilenetv2	Green (Goal_2)	✗	64.1	37.9	44.4
	Blue (Goal_2)	64.7	✗	49.2	37.8
	Orange (Goal_2)	43.4	64.3	✗	60.0
	Purple (Goal_2)	37.9	45.5	64.6	✗
Yolov3	Green (Goal_2)	✗	85.9	70.4	65.1
	Blue (Goal_2)	59.4	✗	47.6	45.7
	Orange (Goal_2)	60.2	55.7	✗	55.2
	Purple (Goal_2)	57.3	70.6	62.1	✗
Faster-RCNN	Green (Goal_2)	✗	80.3	75.9	50.8
	Blue (Goal_2)	40.7	✗	65.3	42.6
	Orange (Goal_2)	45.8	55.6	✗	25.8
	Purple (Goal_2)	65.3	35.2	45.7	✗

Table 19: ASR of `SwitchPatch` with different W_k on Yolov3.

Colors	w_k	G_1-ASR	G_2-ASR	G_3-ASR
(Blue, Green, Orange)	0.9, 0.1, 0.0	76.0	30.2	0.0
	0.5, 0.3, 0.2	61.6	84.3	0.7
	0.2, 0.6, 0.2	15.7	85.8	33.5
	0.2, 0.2, 0.6	45.9	65.8	71.3

A ADDITIONAL EXPERIMENTS ON TRAFFIC SIGN RECOGNITION

Impact of the color-goal combinations. We investigate how the color selection can affect the goals. We switch the color and align with other attack goals. Specifically, we choose blue, green, orange and purple color, which are aligned with Goal_1 (No vehicles), Goal_2 (Pedestrians), Goal_3 (Speed limit 80), Goal_4 (Ahead only) in classification and Goal_1 (HA), Goal_2 (Traffic light), Goal_3 (Umbrella), Goal_4 (Bird) in object detection, respectively.

The results are shown in Table 18. We have the following observations. First, Blue generally shows higher effectiveness across different models and goals in both two tasks. For instance, in image classification, VGG-16 achieves an ASR of 95.9% with Green (Goal_2) under Blue. Similarly, in object detection, Faster-RCNN achieves an 80.3% ASR under Blue for Green (Goal_2). On the contrary, Purple tends to be less effective compared to other colors. For example, Mobilenetv2 only achieves a 37.9% ASR under Purple for Green (Goal_2) in image classification. Second, the combination of Blue, Green, and Orange with Goal_1 and Goal_2 shows higher effectiveness than Purple combinations for both two tasks, making them particularly potent for deploying `SwitchPatch` in adversarial settings.

Impact of w_k for attack goals. `SwitchPatch` can adjust the weights of different attack goals according to specific scenarios. For example, the attacker wants to prioritize achieving some of the attack goals among N attack goals. `SwitchPatch` provides the weight w_k for adjusting the attack goal during the generation process. In the

Figure 11: The depth estimation results in the physical world using Mono2. From left to right: `SwitchPatch` under benign condition; `SwitchPatch` with red light projection; `SwitchPatch` with green light projection. The depth estimation becomes farther under red light and closer under green light compared to begin condition. It can be seen that although the size of the patch is limited, its depth influence can spread to the entire body of the vehicle.

previous experiment, we set the weight of each attack goal the same. In this section, we study the impact of w_k on the performance of `SwitchPatch`. Specifically, for each attack goal, we adjust its weight from high to low accordingly. Table 19 gives the results evaluated on Yolov3.

Obviously, G_i-ASR improves with the increase of weight w_k, indicating the attacker can adjust the attack effect by himself, which gives him more freedom to choose the attack effect he wants.