# MAGIC: Mastering Physical Adversarial Generation in Context through Collaborative LLM Agents

**Yun Xing**[1,2,6*], **Nhat Chung**[3,6], **Jie Zhang**[6], **Yue Cao**[4,6]
**Ivor Tsang**[4,6], **Yang Liu**[4], **Lei Ma**[1,5], **Qing Guo**[2†]
[1]University of Alberta, Canada
[2]VCIP, CS, Nankai University, China
[3]National University of Singapore, Singapore
[4]Nanyang Technological University, Singapore
[5]The University of Tokyo, Japan
[6]CFAR and IHPC, Agency for Science, Technology and Research (A*STAR), Singapore

## Abstract

Physical adversarial attacks in driving scenarios can expose critical vulnerabilities in visual perception models. However, developing such attacks remains non-trivial due to diverse real-world environmental influences. Existing approaches either struggle to generalize to dynamic environments or fail to achieve consistent physical attack performance. To address these challenges, we propose MAGIC (**M**astering Physical **A**dversarial **G**eneration **I**n **C**ontext), a novel framework powered by multi-modal LLM agents to automatically understand the scene context during testing time and generate adversarial patches through synergistic interaction of language and vision understanding. Specifically, MAGIC orchestrates three specialized LLM agents: the adv-patch generation agent masters the creation of deceptive patches via strategic prompt manipulation for text-to-image models; the adv-patch deployment agent ensures contextual coherence by determining optimal deployment strategies based on scene understanding; and the self-examination agent completes this trilogy by providing critical oversight and iterative refinement of both processes. We validate our approach with both digital and physical scenarios, *i.e.*, nuImage and real-world scenes, where both statistical and visual results demonstrate that our MAGIC is powerful and effective for attacking widely applied object detection systems, such as YOLO and DETR series.

**Project website** — https://magic-atk.github.io

## Introduction

Adversarial attacks serve as a crucial method for evaluating the robustness and safety of deep learning models. Physical adversarial attacks, in particular, focus on creating adversarial patches that can be printed and deployed in the real world to mislead visual perception models (Kong et al. 2024; Sato et al. 2024; Cao et al. 2024). Unlike their digital counterparts, physical adversarial attacks can directly expose vulnerabilities in deployed models during practical application (Sato et al. 2024). This is crucial for safety-sensitive domains such as autonomous driving (AD) (Chen et al. 2024; Zhao et al. 2024). When object detectors widely used in autonomous vehicles misidentify critical traffic participants (Mao et al. 2023b; Jia et al. 2023) they may trigger incorrect vehicle responses, potentially creating hazardous situations for both drivers and pedestrians such as sudden stop on highway. By exposing these vulnerabilities, physical adversarial attacks highlight the urgency for more robust and reliable visual perception in safety-critical autonomous systems.

Fundamentally, physical adversarial attacks must address two key difficulties: ❶ *ensuring patch effectiveness against diverse real-world factors (*e.g.*, color shifts, scale variation, view angle changes)*, and ❷ *preserving patch stealthiness within the scene, that is fooling deep models without misleading human's judgment*. Both challenges essentially require a deep understanding of scene context and the ability to incorporate that context into both adversarial patch generation and deployment, as shown in Fig. 1. Traditional optimization-based methods (Zhao et al. 2019; Jia et al. 2022; Wang et al. 2023a) has attempted to generate environment-resilient patches, but they struggle to achieve consistent physical adversarial attack performance and generalize across dynamic environments due to the inherent digital-to-physical gap. Recently, natural denoising diffusion attack (NDDA) (Sato et al. 2024) demonstrated that diffusion models can generate physical, robust adversarial patches via prompt engineering without optimization. However, it overlooks the critical influence of physical scene context, compromising both attack effectiveness and stealthiness. Specifically, NDDA fails to consider two critical aspects: ❶ *the patches' ability to remain effective and stealthy across different scenes*, and ❷ *the appropriate deployment setup of the patch within the scene*. In this work, our analyses reveal that both scene context and deployment strategy substantially affect NDDA patch performance.

To address the challenges, we reformulate physical adversarial attacks as a one-shot patch generation problem, and we develop a framework that automatically creates and deploys patches conditioned on the test-time environ-

*This work was done during Yun Xing was an intern at CFAR & IHPC, A*STAR and Nankai University.

†Corresponding author, email: tsingqguo@ieee.org.

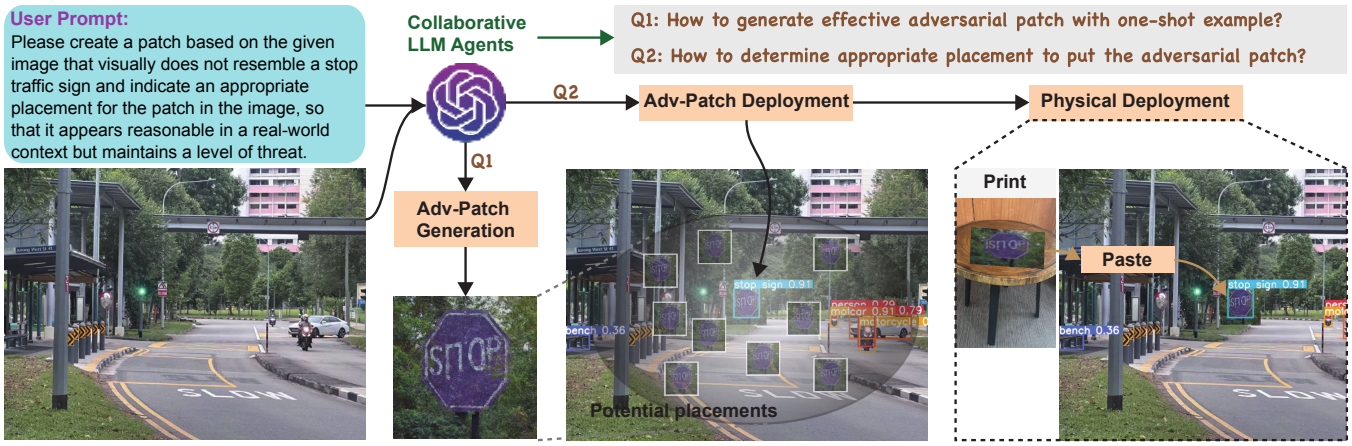


Figure 1: Intuitive idea of our proposed MAGIC framework. Given the user instruction indicating the attack objective and an image of a physical scene, our method aims to *Q1: generate an attack-effective adversarial patch* and *Q2: automatically deploy it into the real world with stealthiness*. We propose to achieve the goals through multi-modal agent reasoning and planning where multiple agents are leveraged to collaboratively realize the generation and deployment for physical attack effectiveness.

ment and user-specified attack objectives. Thus, we propose MAGIC *(**M**aster Physical **A**dversarial **G**eneration **I**n **C**ontext through Collaborative Multi-modal LLM Agents)*, a framework where multiple LLM agents collaboratively generate and deploy effective, stealthy adversarial patches based on comprehensive test-time scene analyses. MAGIC operates in three key stages: (i) patch proposal, (ii) patch deployment, and (iii) patch refinement, powered by three specialized LLM agents to master physical adversarial attack in real-world AD settings. Our contributions are summarized:

- We investigate the diffusion-based physical attacks (Sato et al. 2024) across diverse environments and deployment strategies, revealing that the attack performance is highly dependent on scene context and deployment setup.
- We propose MAGIC, a novel framework that leverages collaborative LLM agents to realize physical adversarial patch generation and deployment based on comprehensive test-time scene analyses. To the best of our knowledge, this is the first approach that apply LLM agents for physical adversarial generation.
- We evaluate MAGIC by attacking three well-known object detectors (YOLOv5, RT-DETR, YOLOv10) using both synthetic (*i.e.*, digital patch insertion) and physical (*i.e.*, real-world patch capturing) approaches, providing extensive analysis of their vulnerabilities for use in AD.

## Related Works

**Physical Adversarial Attacks.** Physical adversarial attacks are designed to stress-test the perception of autonomous system in the real world (Wang et al. 2023b), particularly against classification (Casper et al. 2022; Doan et al. 2022; Zhong et al. 2022), detection (Suryanto et al. 2022; Xu et al. 2020), and other applications (Ding et al. 2021; Chung et al. 2024; Guo et al. 2020). These attacks manipulate models to misclassify or overlook the targets (Chen et al. 2024), revealing practical threats. In particular, physical adversarial patches (Zhang et al. 2023; Wei et al. 2023; Du et al. 2022; Doan et al. 2022; Sato et al. 2024) are easily replicable and widely used to induce mispredictions in deployed models. Consequently, research on physical adversarial attacks, including attack effectiveness (Thys, Van Ranst, and Goedemé 2019; Wei et al. 2023) and attack stealthiness (Wang et al. 2021; Tan et al. 2021; Huang et al. 2022b), remains essential for informing the development of safety-critical systems.

**Adversarial Attack Design.** Adversarial attacks exploit neural network vulnerabilities by misleading model predictions (Goodfellow, Shlens, and Szegedy 2015; Gao et al. 2024; Xing et al. 2024), which can diverge from human judgment to raise safety concerns (Wang et al. 2023b). Conventional works add subtle perturbations that are unnoticeable to humans (Gu et al. 2022), put adversarial stickers (Eykholt et al. 2018) or place small patches to the attack scenarios (Xue et al. 2023; Doan et al. 2022). (Hendrycks et al. 2021) found that even normal images in the wild can pose adversarial effects. It has been hypothesized that these adversarial attacks are caused by non-robust, target-specific features rather than inherent model issues (Hendrycks et al. 2021). Thus, a recent study (Sato et al. 2024), NDDA, employs diffusion models to generate natural adversarial examples by manipulating the target object's robust features.

**Multimodal LLM Reasoning.** Inspired by the emergent capabilities of LLMs in key techniques such as zero-shot prompting (Wei et al. 2022), in-context reasoning (Wang et al. 2024b), multi-modal reasoning (Zhang et al. 2024), and self-feedback (Huang et al. 2022a), autonomous agents have made significant strides in mimicking human interactions (Du et al. 2024). While language-based agents (Wang et al. 2024b) pioneered such interactions, multi-modal embodied systems (Mao et al. 2023a; Zhang and Zhang 2024; Qin et al. 2024) have extended these capabilities to real-world scenarios by integrating image (Wang et al. 2024c; Fu et al. 2024), video (Shen et al. 2024) and audio (Huang et al. 2024). Notably, LLM-based reasoning can be incorporated to refine diffusion outputs, highlighting the potential for improved self-feedback (Yang et al. 2024).

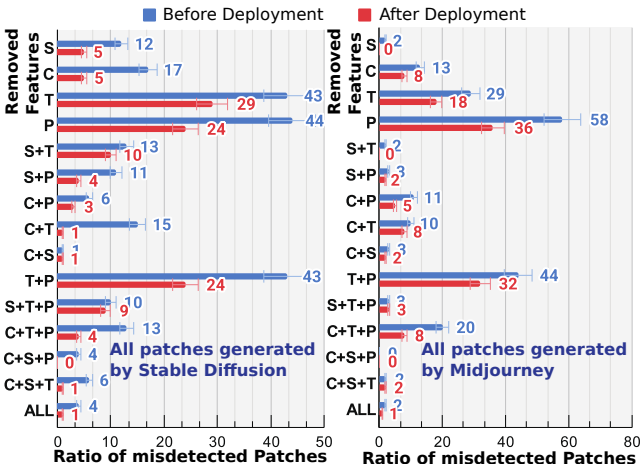


Figure 2: Detection error ratios ($\epsilon_i^{\mathbf{I}}$, $\epsilon_i^{\mathbb{I}}$) of two image sets ($\{\mathbf{I}_i^k\}_{k=1}^K$, $\{\mathbb{I}_i^k\}_{k=1}^K$) across 15 types of text prompt ($i = 1, \ldots, 15$). The y-axis labels indicate removed features: Shape (S), Color (C), Text (T), and Pattern (P). Patches are generated by (a) Stable Diffusion and (b) Midjourney.

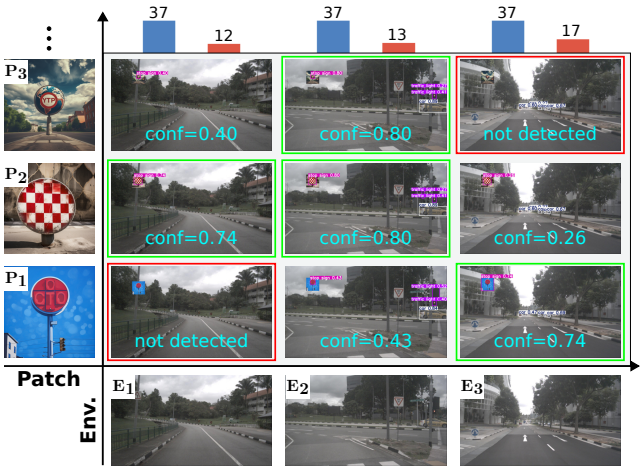


Figure 3: YOLOv5 detection results across three NDDA patches ($\mathbf{P}_1$,$\mathbf{P}_2$,$\mathbf{P}_3$) and three environments ($\mathbf{E}_1$,$\mathbf{E}_2$,$\mathbf{E}_3$). Patch realizes the attack goal when confidence score (conf) $\geq 0.5$. Bar plot at the top: detection error ratios before and after deployment over all the patches for each environment.

## Revisit Natural Denoising Diffusion Attack

NDDA leverages SOTA text-to-image (T2I) models, *e.g.*, Stable Diffusion (Rombach et al. 2022a), to generate adversarial patches through strategic manipulation of the text prompt. Given a text prompt $\mathcal{T}$ that specifies a subject to attack, *e.g.*, stop sign, and its robust features, *i.e.*, shape, color, text, and pattern (Ge et al. 2022; Grill-Spector and Malach 2004), the diffusion model generates a corresponding patch. NDDA then systematically modifies the patch by altering the prompt $\mathcal{T}$, which selectively removes robust features from the normal prompt. For example, the original prompt "Stop sign." may become "Blue square stop sign." by removing the robust features of color and shape. As reported in (Sato et al. 2024), the selective removal of robust features yields patches that could achieve high detection error rates, *i.e.*, *successful attack where detector identifies the patch as a stop sign while human does not*. Such adversarial attack effectiveness aligns with the findings from (Ilyas et al. 2019), which demonstrate adversarial attacks that exploit non-robust features are predictive for DNNs but incomprehensible to humans.

### NDDA in Deployment

To investigate the performance of NDDA after deployment, we set "stop sign" as the subject and aim to generate patches for attacking YOLOv5 (Jocher 2020). Following NDDA's setup, we have the benign prompt for the T2I generation, *i.e.*, "Stop sign.". We adopt a prompt set $\{\mathcal{T}_i\}_{i=1}^{15}$ containing 15 types of prompt where we remove different robust features from the benign prompt. We show the 15 types in the y-axis in Fig. 2. For each prompt $\mathcal{T}_i$, we generate $K$ patches $\{\mathbf{P}_i^k\}_{k=1}^K$ by feeding the prompt into the diffusion model and sampling $K$ times. Here, we use the patches officially released by NDDA to avoid any confusion.

Given an environment image $\mathbf{E}$ and a generated patch $\mathbf{P}_i^k$, we conduct two types of testing: ❶ *After deployment*. We create $\mathbf{I}_i^k$ by digitally inserting $\mathbf{P}_i^k$ at location $p$ in $\mathbf{E}$ with scale $s$ representing the patch size. ❷ *Before deployment*. We also create an image $\mathbb{I}_i^k$ by inserting $\mathbf{P}_i^k$ to an empty image $\mathbf{E}^0$ with the same $p$ and $s$, where $\mathbf{E}^0$ has the same size to $\mathbf{E}$ but all pixels are zero. For $K$ patches generated from the $i$th prompt, we obtain $2 \times K$ images, *i.e.*, $\{\mathbf{I}_i^k\}_{k=1}^K$ and $\{\mathbb{I}_i^k\}_{k=1}^K$, and perform detection on all images using YOLOv5 (Jocher 2020). For each respective set of patches, we calculate the ratio of incorrectly detected patches, *i.e.*, detected wrongly as stop signs, over all generated patches, denoted as $\epsilon_i^{\mathbf{I}}$ and $\epsilon_i^{\mathbb{I}}$. Basically, these metrics measured the attack effectiveness of the $i$th prompt both before and after the deployment.

Based on the above setups, we investigate three research questions: **RQ1:** Whether different type of patches maintain their attack effectiveness after deployment? **RQ2:** How do different environments influence the attack effectiveness of the generated patches? **RQ3:** What are the influences of deployment setup on the patches' attack performance?

### Empirical Results and Discussion

**Response to RQ1.** We investigate the environmental influence on patch attack effectiveness by comparing detection error ratios between $\{\mathbb{I}_i^k\}_{k=1}^K$ and $\{\mathbf{I}_i^k\}_{k=1}^K$. Using the environment $\mathbf{E}_1$ shown in Fig. 3, the results across all 15 prompt types are presented in Fig. 2, revealing following key findings: ❶ The detection error ratio before deployment ($\epsilon_i^{\mathbb{I}}$) generally exceeds its after-deployment counterpart ($\epsilon_i^{\mathbf{I}}$) across all prompt types, demonstrating that environment significantly degraded the patches' performance. ❷ The magnitude of reduction in detection error ratios varies notably across prompt types, indicating that text prompts have influence on the environment deployment. ❸ The pattern of detection error ratios across prompts remains consistent between Stable Diffusion and Midjourney generations when comparing the two plots in Fig. 2, suggesting these findings generalize across different image generation architectures.

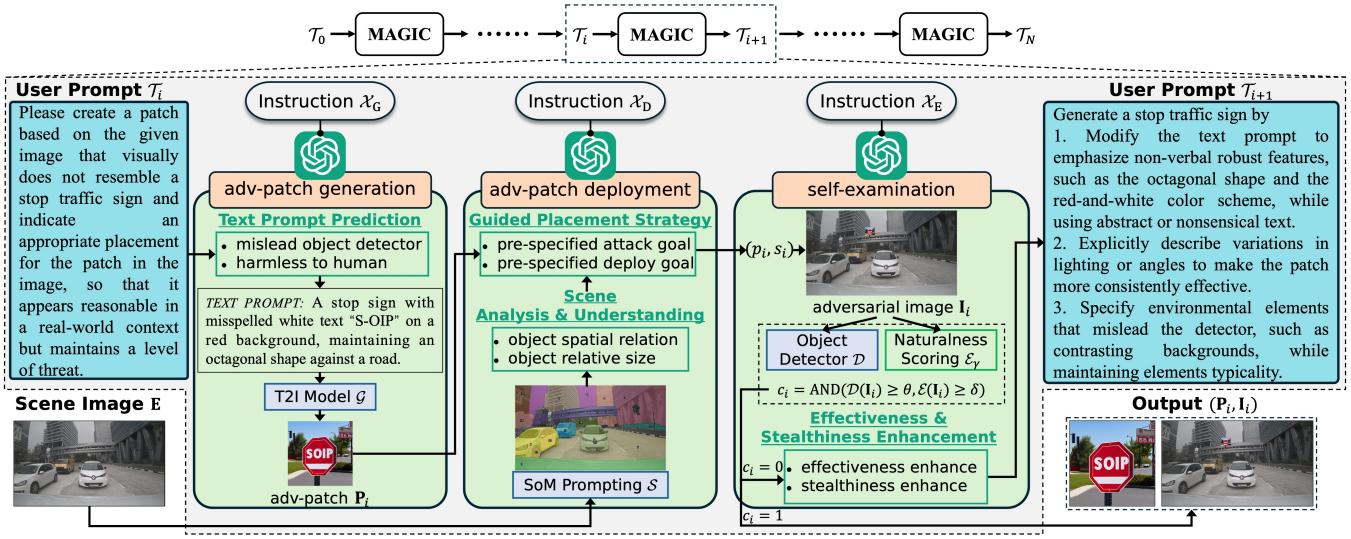


Figure 4: Overall pipeline of the proposed MAGIC framework. Please zoom in for better visualization.

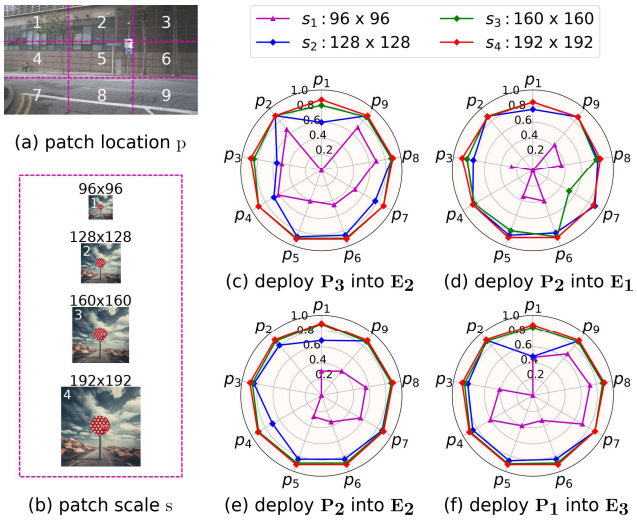


Figure 5: (a)-(b): candidate patch locations and scales to evaluate. (c)-(f): the detection confidence across all candidate locations and scales for effective $(\mathbf{P}, \mathbf{E})$ pairs in Fig. 3.

**Response to RQ2.** As Fig. 3 shows, we deploy 3 generated patches into 3 different environments to see whether the environments would affect the effectiveness of the attack. Meanwhile, we summarize the detection error ratios before and after deployment for all generated patches. We observe that the same patch presents different attack effectiveness in different environments. For example, $\mathbf{P}_3$ is identified as stop sign in the $\mathbf{E}_2$ with a high confidence 0.80, while it is undetectable in $\mathbf{E}_3$. Notably, such variations in detection error ratios can be observed across all different environments.

**Response to RQ3.** To investigate the influence of deployment setup, as shown in Fig. 5 (a) (b), we divide the scene image into a $3 \times 3$ grid, designating each region's center as the location for patch insertion, and test four different patch scales. We re-evaluate the performance for all the effective $(\mathbf{P}, \mathbf{E})$ pairs from Fig. 3 across different locations and scales. The results are shown in Fig. 5 as radar plots. We observe that large-scale patches yield consistent results across locations, while small-scale patches show substantial location-dependent variation influenced by patch and environment. Furthermore, the deployment of the patch should be physically feasible and contextually plausible. However, some of the deployment locations are obviously impractical and will potentially distract human observers. For example, in Fig. 3, the patches in $\mathbf{E}_2$ are positioned in the sky, clearly violating physical constraints, and those in $\mathbf{E}_3$ are placed on building facades, creating contextually incongruous arrangements.

**Motivations.** Our analysis above shows that the scene context and deployment setup, neglected by existing works like NDDA and traditional optimization-based methods, lead to significant instabilities in patch performances. This highlights the need for an advanced diffusion-based adversarial patch generation framework that ❶ generate patches that adaptable to diverse real-world scenes, ❷ determines context-appropriate deployment strategies, and ❸ keep misleading object detection systems after deployment.

## MAGIC Methodology

As shown in Fig. 4, we propose to use a multi-modal LLM denoted as $\mathcal{V}$ to handle the three distinct requirements by iteratively refining adversarial patch generation and deployment. Given the initial user instruction $\mathcal{T}$ indicating the attack objective, we first build the adv-patch generation agent, *i.e.*GAgent $\mathcal{V}_\text{G}$, which predicts a text prompt based on $\mathcal{T}$ and feed it into the T2I diffusion model to generate an adversarial patch. The process is formulated as,

$$\mathbf{P}_i = \mathcal{V}_\text{G}(\mathcal{T}_i, \mathcal{G}(\cdot), \mathcal{X}_\text{G}), \tag{1}$$

where $\mathcal{T}_i$ is the attack objective at the $i$th iteration. We have $\mathcal{T}_0 = \mathcal{T}$, $\mathcal{G}(\cdot)$ is the T2I diffusion model to generate the adversarial patch according to the text prompt, and $\mathcal{X}_\text{G}$ is the instruction defining $\mathcal{V}_\text{G}$'s capability. To allow the refinement, we update $\mathcal{T}_i$ according to the final result of iteration $i-1$.

With the generated adv-patch $\mathbf{P}_i$, we setup the adv-patch deployment agent *i.e.*, DAgent $\mathcal{V}_\text{D}$, empowered with set-of-mark (SoM) (Yang et al. 2023) to determine its appropriate location $p_i$ and scale $s_i$ within the target environment $\mathbf{E}$:

$$(p_i, s_i) = \mathcal{V}_\text{D}(\mathbf{E}, \mathbf{P}_i, \mathcal{S}(\cdot), \mathcal{X}_\text{D}), \tag{2}$$

where instruction $\mathcal{X}_\text{D}$ defines $\mathcal{V}_\text{D}$'s capability and $\mathcal{S}(\cdot)$ is the SoM prompting. Subsequently, we insert the patch $\mathbf{P}_i$ into $\mathbf{E}$ at $p_i$ with scale $s_i$ obtaining the adversarial image $\mathbf{I}_i$.

However, we do not know whether the adversarial image is good enough to mislead the object detector $\mathcal{D}(\cdot)$ and meet the stealthiness requirement or not. To address this issue, we propose to build a self-examination agent, *i.e.*, EAgent $\mathcal{V}_\text{E}$, that is designed to analyze the adversarial image $\mathbf{I}_i$ and optimize the user prompt $\mathcal{T}_{i+1}$. We formulate the process as,

$$(\mathcal{T}_{i+1}, c_i) = \mathcal{V}_\text{E}(\mathbf{I}_i, \mathcal{T}_i, \mathcal{D}(\cdot), \mathcal{X}_\text{E}), \tag{3}$$

where the instruction $\mathcal{X}_\text{E}$ defines $\mathcal{V}_\text{E}$'s capability and $c_i$ is a binary examination result. If $c_i = 0$, we trigger the next round iteration by setting $i = i + 1$, otherwise, we stop iteration and output $\mathbf{P}_i$ and $\mathbf{I}_i$ as the final result.

## Adv-Patch Generation Agent

At the $i$th iteration, we have the user instruction $\mathcal{T}_i$ and the environment image $\mathbf{E}$ as input. We first prompt the LLM $\mathcal{V}$ with the instruction $\mathcal{X}_\text{G}$ to predict a text prompt $\mathcal{P}_i$, which describes a patch approaching the attack goal, based on the instruction $\mathcal{T}_i$. Next, we feed $\mathcal{P}_i$ into $\mathcal{G}$, *i.e.*, Stable Diffusion v2 (Rombach et al. 2022a), and obtain an adversarial patch $\mathbf{P}_i = \mathcal{G}(\mathcal{P}_i)$. The text prompt prediction capability of $\mathcal{V}$ is defined by the instruction $\mathcal{X}_\text{G}$ which summarized as:

**Instruction: $\mathcal{X}_\text{G}$**

Given an user instruction which described the expected objective of a visual patch. You are required to predict a text description of the subject that capable of realizing the objective. The ultimate goal is to let the visual patch generated based on the text description has a deceptive appearance, such that

- an object detector will recognize the visual patch as an instance of a specific semantic category with high confidence,
- but human will recognize the visual patch as just an abstract art which does not belongs to any specific category.

## Adv-Patch Deployment Agent

With the generated patch $\mathbf{P}_i$, we aim to determine a practical deployment place and its specific location $p_i$ and scale $s_i$ within the given environment $\mathbf{E}$. To achieve this, SoM prompting $\mathcal{S}$ (Yang et al. 2023) is applied first to tag the key elements within $\mathbf{E}$. Then we prompt the LLM $\mathcal{V}$ with instruction $\mathcal{X}_\text{D}$ as summarized below to comprehend SoM results and determine the most potential location and scale within the $\mathbf{E}$ that can achieve the objectives specified by $\mathcal{T}_i$.

**Instruction: $\mathcal{X}_\text{D}$**

You are provided with an image of a scene and the corresponding semantic elements marked with numbers. For a given visual patch, you are required to accomplish the following two tasks:

1. **Scene Analysis & Understanding: a.** Perceive all the elements within the given scene. **b.** Comprehend the spatial relationship and relative size of all the elements inside the scene by referring to the numbered segmentations.
2. **Guided Placement Strategy: a.** Check the content of the given patch and compare it with the elements in the environment. **b.** Comprehend the pre-defined attack effectiveness and deployment stealthiness goal. (See supplemental material for details) **c.** Based on the patch content, determine where should the patch be placed so that the goals can be achieved. **d.** Based on the determined patch place, decide what specific scale and location should be applied to the patch so that the attack goals can be achieved.

Once the deployment location $p_i$ and scale $s_i$ are determined, we resize the patch $\mathbf{P}_i$ to $s_i$ and insert it at $p_i$ into the scene image $\mathbf{E}$, which results the adversarial image $\mathbf{I}_i$.

## Self-Examination Agent

With the adversarial image ready, we proceed to assess the patch's performance for attacking the object detector and keeping stealthy within the scene. We first feed $\mathbf{I}_i$ into the target object detector $\mathcal{D}$ to get detection results $\hat{\mathbf{I}}_i = \mathcal{D}(\mathbf{I}_i)$. Then we inspect if the detection confidence of the patch passes the pre-defined threshold $\theta$ as $\mathcal{D}(\mathbf{I}_i) \geq \theta$. To examine stealthiness, we set up an independent LLM $\mathcal{E}$ with instruction $\gamma$ as the simulated human observer to generate a naturalness score for the patch. Similarly, a threshold $\delta$ is applied to check whether the patch satisfies $\mathcal{E}(\mathbf{I}_i, \gamma) \geq \delta$. The patch $\mathbf{P}_i$ is then recognized as stealthy if the inequality holds true. The instruction $\gamma$ is summarized as:

**Naturalness Score Instruction: $\gamma$**

You are provided with a visual patch and an image where the patch is deployed into an environment. Supposing you are a human, you are required to judge whether the patch is stealthy or not within the environment regarding to its location and scale.

1. **Image Understanding:** Perceive all the elements inside the given image regarding their spatial relation and relative size.
2. **Location Inspection:** Check the patch's location within the image and determine whether it is a reasonable place.
3. **Scale Inspection:** Check the patch's scale and determine whether its size is reasonable compared to other objects.
4. **Naturalness Summarization:** Give a score range from 0 to 1 basedon the reasoning of the patch's location and scale.

Afterwards, we set $c_i$ as a binary operator, calculated by $c_i = \text{AND}(\mathcal{D}(\mathbf{I}_i) \geq \theta, \mathcal{E}(\mathbf{I}_i, \gamma) \geq \delta)$, to get the final examination results. If $c_i = 1$, the iteration exits and $(\mathbf{P}_i, \mathbf{I}_i)$ is returned. Otherwise, refinement is required and we further prompt the LLM $\mathcal{V}$ with instruction $\mathcal{X}_\text{E}$ to update the user prompt $\mathcal{T}_i$ based on the assessment results. Then a new round of agent planning is started by setting $\mathcal{T}_i = \mathcal{T}_{i+1}$.

**Instruction: $\mathcal{X}_\text{E}$**

You are provided with an image where a visual patch is deployed into an environment. The image has been evaluated for its effectiveness and stealthiness of attacking the target object detector. Given the evaluation results, you are required to reflect on why the patch can or cannot be effective or stealthy then propose suggestions of how to change the patch's appearance.

1. **Attack Rules Understanding:** Comprehend the predefined rules of how to attack an object detector by deploying a visual patch within a scene. (See supplemental material for the definition of rules)
2. **Attack Effectiveness Enhancement: a.** Perceive patch's appearance in the image and summarize the detection result. **b.** Compare the patch to rules then summarize your understanding of why the patch cannot be attack effective.
3. **Deployment Stealthiness Enhancement: a.** Perceive patch's location and scale in the image and summarize the naturalness score results. **b.** Compare the patch in the image to the rules then suggest how to adjust its location and scale to make it more stealthy.

## Extension to Physical World

Finally, we get the optimal patch $\mathbf{P}^*$ and the corresponding deployment strategy $(p^*, s^*)$. We physically deploy the patch $\mathbf{P}^*$ by printing it out, place it at the location $p^*$ in $\mathbf{E}$ and re-capture an image of the scene with the patch size as $s^*$ to match the reference image $\mathbf{I}$. While MAGIC operates via digital feedback, the generated patches enjoy both the great digital-to-physical transferability and the advancing test-time environment generalization. We present a comprehensive analysis and results in the supplement.

# Experiments

## Experimental Setup

**Digital & Physical Environments.** We use the images from nuImage (Fong et al. 2021) for digital evaluation, which consists of images captured by 6 car-mounting cameras from different views, *i.e.*, back, back left, back right, front, front left, front right. We select one image for each view (See Fig. 6) as an initial study. We further physically verify our MAGIC framework with two different physical scenarios (See Fig. 7). One is a real-world bus bay and another is a regular road with few pedestrians. More comprehensive experimental scenarios can be found in the supplement.

**Baselines.** We set two variants of NDDA as the diffusion-based physical attack baseline: ❶ we first deploy NDDA patches with random location, dubbed "NDDA Rand"; ❷ then we deploy NDDA patches with our DAgent as "NDDA+DAgent" to demonstrate the advantage of our MAGIC framework. For optimization-based baselines, the patches from (Zhao et al. 2019) and (Wang et al. 2023a) are evaluated, dubbed as OPT1 and OPT2 respectively.

**Generator & Detectors.** To keep the fairness of comparison, we follow NDDA and adopt Stable Diffusion v2 (Rombach et al. 2022b) as the text-to-image generator. For the attack targets, YOLOv5 (Jocher 2020) and RT-DETR (Lv et al. 2023) are evaluated. Moreover, we empirically found YOLOv5 and DETR are easily disturbed, thus YOLOv10 (Wang et al. 2024a) is further adopted as the main evaluation target. More results are presented in supplement.

**Metrics.** We evaluate the patches' attack performance by Attack Success Rate (ASR), which measures the percentage of the patches that successfully deceived the target object detector. ASR has a range from 0 to 100, where higher values signify greater adversarial attack performance.

| | | Features | | | | Object Detectors | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | S | C | T | P | YOLOv5 | RT-DETR | YOLOv10 | Avg. |
| Environment ① | NDDA | ✔ | | | | 23.00% | 23.00% | 10.00% | 18.66% |
| | | | ✔ | | | 15.00% | 48.00% | 6.00% | 23.00% |
| | +Rand | | | ✔ | | 46.00% | 56.00% | 30.00% | 44.00% |
| | | | | | ✔ | 47.00% | 56.00% | 32.00% | 45.00% |
| | | ✔ | ✔ | ✔ | ✔ | 8.66% | 9.33% | 4.66% | 7.55% |
| | +DAgent | | | ✔ | | *48.00%* | 51.00% | 33.00% | 44.00% |
| | +DAgent | | | | ✔ | 47.00% | *57.00%* | *36.00%* | 46.66% |
| | OPT1 | | | | | 6.00% | 6.00% | 0.00% | 4.00% |
| | OPT2 | | | | | 10.00% | 6.00% | 6.00% | 7.33% |
| | MAGIC | | | | | **88.00%** | **80.00%** | **74.00%** | 80.66% |
| Environment ② | NDDA | ✔ | | | | 14.00% | 40.00% | 17.00% | 23.66% |
| | | | ✔ | | | 9.00% | 51.00% | 10.00% | 23.33% |
| | +Rand | | | ✔ | | 42.00% | 74.00% | 32.66% | 49.55% |
| | | | | | ✔ | 41.00% | 71.00% | 36.00% | 49.33% |
| | | ✔ | ✔ | ✔ | ✔ | 6.00% | 21.00% | 4.00% | 10.33% |
| | +DAgent | | | ✔ | | 45.00% | *78.00%* | 36.00% | 53.00% |
| | +DAgent | | | | ✔ | *51.00%* | *78.00%* | *39.33%* | 56.11% |
| | OPT1 | | | | | 6.00% | 8.00% | 2.00% | 5.33% |
| | OPT2 | | | | | 10.00% | 12.00% | 8.00% | 10.00% |
| | MAGIC | | | | | **66.00%** | **94.00%** | **92.00%** | 84.00% |
| Environment ③ | NDDA | ✔ | | | | 15.00% | 21.00% | 6.00% | 14.00% |
| | | | ✔ | | | 5.00% | 39.00% | 11.00% | 18.33% |
| | +Rand | | | ✔ | | 40.00% | 58.00% | 26.66% | 41.55% |
| | | | | | ✔ | 38.00% | 59.00% | 3.00% | 33.33% |
| | | ✔ | ✔ | ✔ | ✔ | 2.00% | 12.00% | 2.00% | 4.66% |
| | +DAgent | | | ✔ | | *43.00%* | *60.00%* | *33.00%* | 45.33% |
| | +DAgent | | | | ✔ | *43.00%* | *60.00%* | 28.00% | 43.66% |
| | OPT1 | | | | | 4.00% | 6.00% | 0.00% | 3.33% |
| | OPT2 | | | | | 14.00% | 12.00% | 6.00% | 10.67% |
| | MAGIC | | | | | **84.00%** | **94.00%** | **90.00%** | 89.33% |
| Environment ④ | NDDA | ✔ | | | | 16.00% | 14.00% | 7.00% | 12.33% |
| | | | ✔ | | | 10.00% | 33.00% | 6.00% | 16.33% |
| | +Rand | | | ✔ | | 42.66% | 53.33% | 23.33% | 39.77% |
| | | | | | ✔ | 45.00% | 47.00% | 28.00% | 40.00% |
| | | ✔ | ✔ | ✔ | ✔ | 4.66% | 10.00% | 1.33% | 5.33% |
| | +DAgent | | | ✔ | | 43.00% | *58.00%* | *32.00%* | 44.33% |
| | +DAgent | | | | ✔ | *49.00%* | 49.00% | *32.00%* | 43.33% |
| | OPT1 | | | | | 4.00% | 10.00% | 4.00% | 6.00% |
| | OPT2 | | | | | 6.00% | 12.00% | 8.00% | 8.67% |
| | MAGIC | | | | | **78.00%** | **90.00%** | **80.00%** | 82.66% |
| Environment ⑤ | NDDA | ✔ | | | | 13.00% | 30.00% | 6.00% | 16.33% |
| | | | ✔ | | | 10.00% | 49.00% | 9.00% | 22.66% |
| | +Rand | | | ✔ | | 44.66% | 72.66% | 32.00% | 49.77% |
| | | | | | ✔ | 49.00% | 67.00% | 36.00% | 50.66% |
| | | ✔ | ✔ | ✔ | ✔ | 7.33% | 19.33% | 2.66% | 9.77% |
| | +DAgent | | | ✔ | | 49.00% | *71.00%* | 39.00% | 53.00% |
| | +DAgent | | | | ✔ | *51.00%* | *71.00%* | *40.00%* | 54.00% |
| | OPT1 | | | | | 4.00% | 6.00% | 2.00% | 4.00% |
| | OPT2 | | | | | 6.00% | 6.00% | 4.00% | 5.33% |
| | MAGIC | | | | | **72.00%** | **92.00%** | **74.00%** | 79.33% |
| Environment ⑥ | NDDA | ✔ | | | | 14.00% | 27.00% | 10.00% | 17.00% |
| | | | ✔ | | | 8.00% | 49.00% | 10.00% | 22.33% |
| | +Rand | | | ✔ | | 39.33% | 60.00% | 25.33% | 41.55% |
| | | | | | ✔ | 43.00% | 57.00% | 28.00% | 42.66% |
| | | ✔ | ✔ | ✔ | ✔ | 3.33% | 12.00% | 2.66% | 5.99% |
| | +DAgent | | | ✔ | | 42.00% | *66.00%* | *32.00%* | 46.66% |
| | +DAgent | | | | ✔ | *45.00%* | 60.00% | *32.00%* | 45.66% |
| | OPT1 | | | | | 24.00% | 10.00% | 6.00% | 13.33% |
| | OPT2 | | | | | 18.00% | 20.00% | 14.00% | 17.33% |
| | MAGIC | | | | | **92.00%** | **96.00%** | **84.00%** | 90.66% |

Table 1: ASR results of our MAGIC and baseline patches with different detectors for nuImage environments. The confidence threshold is set to 0.5 as the same of NDDA. The best results are highlighted in red and the second best in blue, while the best results of each detector for a given environment are marked as **bold** and the second best as *italic*.

## Digital Comparative Results

**Attack Effectiveness.** There are 50 patches for each text prompt in NDDA where several text prompts removed the same robust feature, *e.g.*, both 'square_stop_sign' and 'tri-

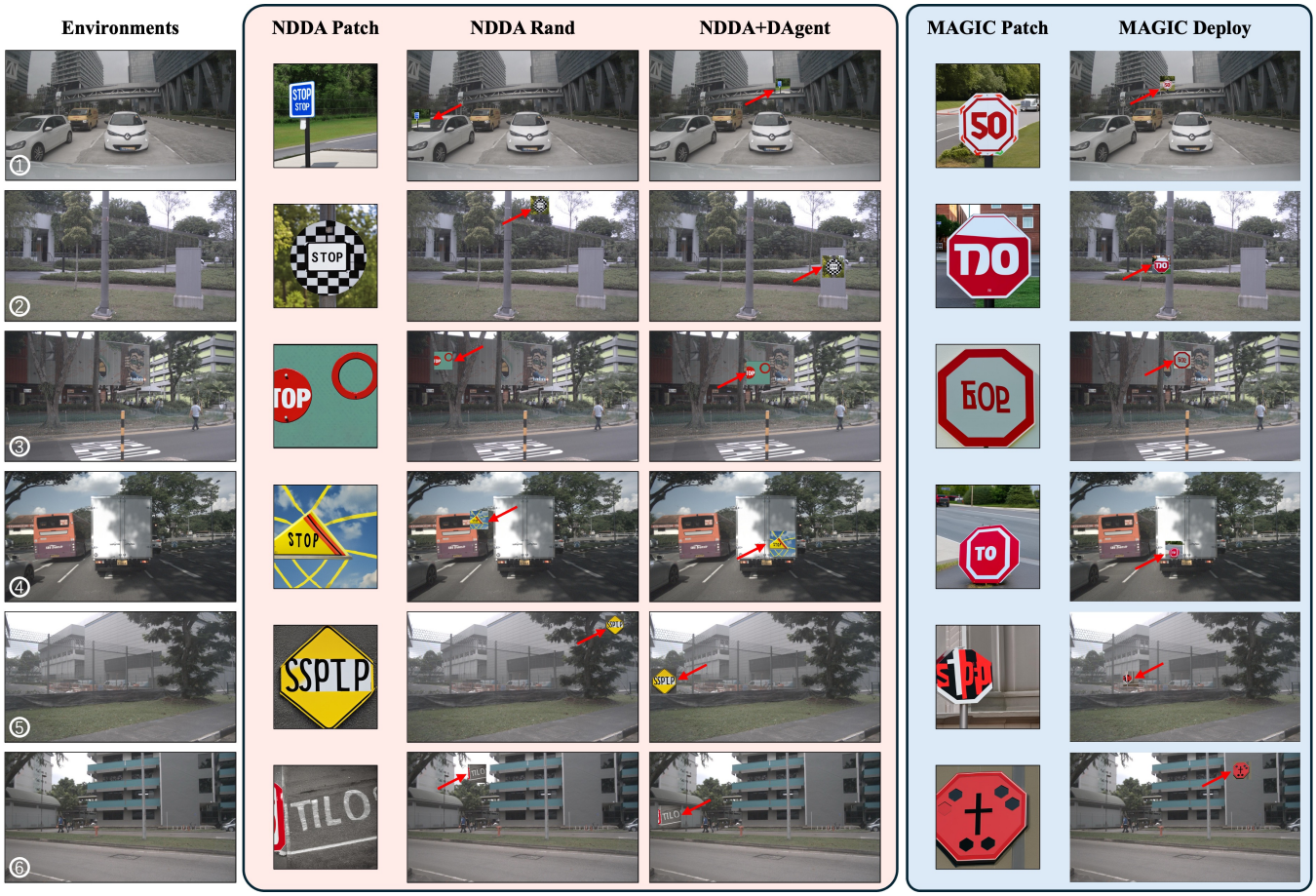


Figure 6: Illustration of the deployment naturality of NDDA and our MAGIC in different environments. Pink region enclosed the deployments of NDDA while the blue region enclosed our MAGIC deployments. Patches are pointed out with red arrows.

angle_stop_sign' removed shape. For a fair comparison, we generate same number of patches for a given environment and report the average results over all the patches for the same text prompt type in NDDA. Note that for the NDDA+DAGent baseline, we setup two variants, *i.e.*, removing text feature and removing pattern feature, as they are empirically more effective than other prompt types. For optimization-based patches, we follow the same routine for each environment and report the average attack success rate.

*Results.* Following NDDA, we evaluate the patches with a confidence threshold of 0.5. Please see supplement for the results with a higher threshold. The statistical results are tabulated in Tab. 1. ❶ Comparing to the baselines, we see our MAGIC enjoys a great attack effectiveness for all the environments. Such cross-environment superior ASR verified the effective patch generation capability of our MAGIC. ❷ While the NDDA Rand performs randomly *i.e.*, around 50%, under the best robust feature removing setup, we note they still receive boosting with our DAgent proving it can facilitates the attack by place the patch appropriately.

**Deployment Stealthiness.** We randomly sample patches from NDDA dataset for comparing the deployment stealthiness, and the patches adopted for our MAGIC are generated through the pipeline in Fig. 4. Note that a better placement means the patch is more easy for physical deployment implementation and more natural to human observers. For the statistical results, please find them in supplement.

*Results.* As shown in Fig. 6, we visually compare the patch deployment results of our MAGIC and NDDA baselines over the selected 6 environments. ❶ Comparing the placement of NDDA Rand against the placement planned by our DAgent in NDDA+DAgent, we see clearly that our DAgent is able to determine a more appropriate location in the scene where the patch is more practical for deployment. ❷ Furthermore, it can be observed that the digital-level deployment results of our MAGIC patches are visually better consistent within the given scene when compared to NDDA+DAgent since our MAGIC considered contextually consistency of the patches during the deployment.

## Physical Evaluation Results

We test in two physical environments: ❶ one bus stop bay with heavy traffics; ❷ one regular road next to a college with some pedestrians. In order to verify MAGIC flexibility, two patches are generated for the first scene. Note that we ensured the patches were not observed by vehicles on the road during testing. More physical experiments and video results, as well as the ethical issues are discussed in the supplement.

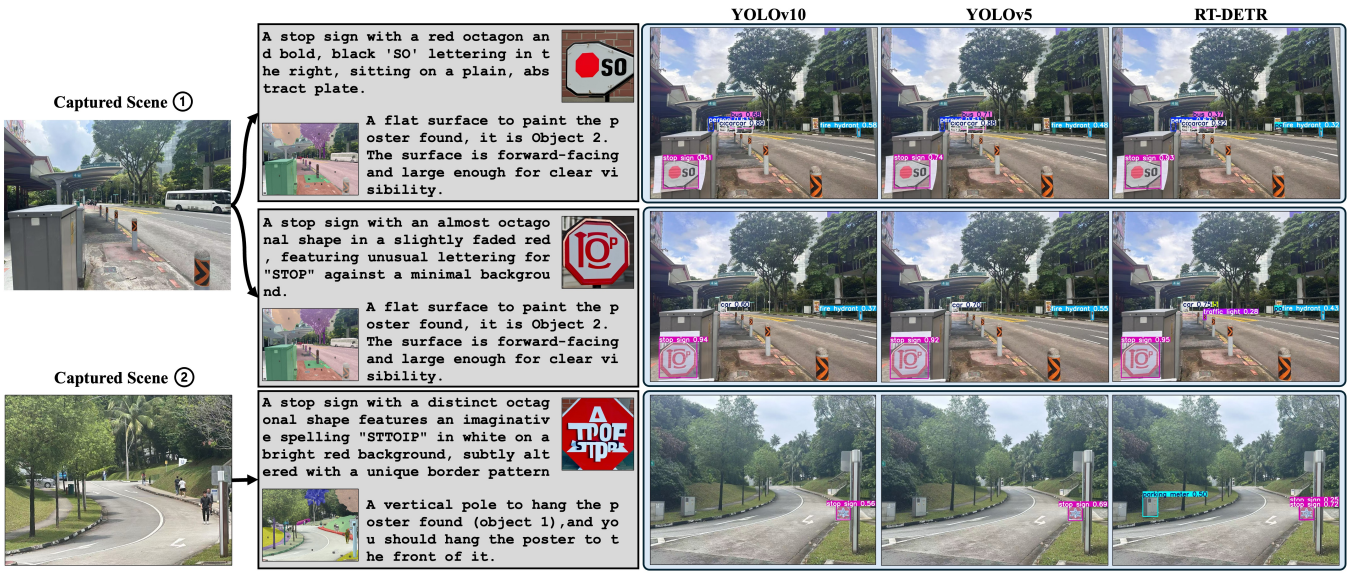


Figure 7: Illustration of our MAGIC patches and the detection results after deployed into two physical scenes. Gray region enclosed the patches and deployment suggestions output by our MAGIC, and the blue region enclosed the detection results.

*Results.* As visualized in Fig. 7, we show the generated patch with prompt, the deployment suggestion and the detection results. ❶ Observing the generally lower detection confidence, we see that the physical attacks are more harder to realize due to the environmental influence. However, our MAGIC still generates extremely effective attack patch for scene ①, *i.e.*, the second patch. ❷ Comparing the results of three detectors, we observe that YOLOv5 and RT-DETR are more prone being attack while YOLOv10 is more robust. However, all the generated patches are proven to be physically attack effective for YOLOv10 which further verified the superiority of our MAGIC. ❸ By comparing the two scenes' results, we see that our MAGIC is capable of giving out contextually appropriate patch and location for real-world deployment. In summary, we can conclude from the results that our MAGIC is full of power for attacking the object detection system in the real-world scenarios.

## Ablation Study

We start ablation from ❶ the basic patch generation where GAgent is isolated without deployment and self-examination, *i.e.*, "GAgent-*naïve*". ❷ The further capability our MAGIC provided is the contextually appropriate deployment of DAgent where we combine GAgent with patch deployment planning, *i.e.*, "↪ *w/* DAgent". ❸ Then, we note the EAgent is responsible for supervising both attack effectiveness (ae) and deployment stealthiness (ds), so we combine GAgent with the patch attack supervision of EAgent which denoted as "↪ *w/* EAgent-ae". Finally, we involve the stealthiness of EAgent getting the proposed MAGIC.

*Results.* As tabulated in Tab. 2, we see that naive patch generation without any text prompt manipulation cannot attack the detectors at all. Consequently, it is obvious that the DAgent cannot significantly improve the attack performance of GAgent, *i.e.*, GAgent-*naïve w/* DAgent. On the contrary, the involvement of attack effectiveness supervision from EAgent greatly boosted the attack effectiveness of the generated patch against all detectors, achieving 39.67%, 45.00% attack effectiveness improvements compared to the DAgent baseline. Finally, our MAGIC achieves its best attack performance by benefiting to the supervision of EAgent and also the appropriate deployment of DAgent.

| | Model | Object Detectors YOLOv5 | RT-DETR | YOLOv10 | Avg. |
|---|---|---|---|---|---|
| Env. ① | GAgent-*naïve* | 4.00% | 6.00% | 0% | 3.33% |
| | ↪ *w/* DAgent | 7.00% | 9.00% | 0% | 5.33% |
| | ↪ *w/* EAgent-ae | 46.00% | 56.00% | 33.00% | 45.00% |
| | MAGIC | **88.00%** | **80.00%** | **74.00%** | 80.66% |
| Env. ② | GAgent-*naïve* | 2.00% | 2.00% | 0% | 1.33% |
| | ↪ *w/* DAgent | 6.00% | 6.00% | 3.00% | 5.00% |
| | ↪ *w/* EAgent-ae | 23.00% | 60.00% | 67.00% | 50.00% |
| | MAGIC | **66.00%** | **94.00%** | **92.00%** | 84.00% |

Table 2: Ablation results of the proposed MAGIC with two different environment image. The best average results are highlighted in red, while the best results for each detector and environment are marked in **bold**.

## Conclusion

In this work, we propose the MAGIC which reformulates and addresses physical adversarial attacks as an one-shot patch generation problem. With multi-agent LLMs, our approach generates adversarial patches that consider the influences of scene context, enabling direct physical deployment in matching environments. Experiments on both digital and physical levels demonstrates our method can effectively generate context-aware patches, deploy them in the real world and attack various applied object detectors. To the best of our knowledge, our work is the very initial study to improve and extend diffusion-based attack in physical scenarios.

## Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG4-GC-2023-008-1B), and the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority. This work was supported in part by JST CRONOS Grant (No. JPMJCS24K8), JSPS KAKENHI Grant (No.JP21H04877, No.JP23H03372, and No.JP24K02920), Canada CIFAR AI Chairs Program, the Natural Sciences and Engineering Research Council of Canada, and the Autoware Foundation.

## References

Cao, Y.; Xing, Y.; Zhang, J.; Lin, D.; Zhang, T.; Tsang, I.; Liu, Y.; and Guo, Q. 2024. SceneTAP: Scene-Coherent Typographic Adversarial Planner against Vision-Language Models in Real-World Environments. *arXiv preprint arXiv:2412.00114*.

Casper, S.; Nadeau, M.; Hadfield-Menell, D.; and Kreiman, G. 2022. Robust feature-level adversaries are interpretability tools. *Advances in Neural Information Processing Systems*, 35: 33093–33106.

Chen, L.; Wu, P.; Chitta, K.; Jaeger, B.; Geiger, A.; and Li, H. 2024. End-to-End Autonomous Driving: Challenges and Frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 10164–10183.

Chung, N.; Gao, S.; Vu, T.-A.; Zhang, J.; Liu, A.; Lin, Y.; Dong, J. S.; and Guo, Q. 2024. Towards transferable attacks against vision-llms in autonomous driving with typography. *arXiv preprint arXiv:2405.14169*.

Ding, L.; Wang, Y.; Yuan, K.; Jiang, M.; Wang, P.; Huang, H.; and Wang, Z. J. 2021. Towards universal physical attacks on single object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1236–1245.

Doan, B. G.; Xue, M.; Ma, S.; Abbasnejad, E.; and Ranasinghe, D. C. 2022. Tnt attacks! universal naturalistic adversarial patches against deep neural network systems. *IEEE Transactions on Information Forensics and Security*, 17: 3816–3830.

Du, A.; Chen, B.; Chin, T.-J.; Law, Y. W.; Sasdelli, M.; Rajasegaran, R.; and Campbell, D. 2022. Physical adversarial attacks on an aerial imagery object detector. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1796–1806.

Du, Y.; Li, S.; Torralba, A.; Tenenbaum, J. B.; and Mordatch, I. 2024. Improving Factuality and Reasoning in Language Models through Multiagent Debate. In *Forty-first International Conference on Machine Learning*.

Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust Physical-World Attacks on Deep Learning Visual Classification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 1625–1634.

Fong, W. K.; Mohan, R.; Hurtado, J. V.; Zhou, L.; Caesar, H.; Beijbom, O.; and Valada, A. 2021. Panoptic nuScenes: A Large-Scale Benchmark for LiDAR Panoptic Segmentation and Tracking. *arXiv preprint arXiv:2109.03805*.

Fu, X.; Hu, Y.; Li, B.; Feng, Y.; Wang, H.; Lin, X.; Roth, D.; Smith, N. A.; Ma, W.; and Krishna, R. 2024. BLINK: Multimodal Large Language Models Can See but Not Perceive. *CoRR*, abs/2404.12390.

Gao, S.; Jia, X.; Ren, X.; Tsang, I. W.; and Guo, Q. 2024. Boosting Transferability in Vision-Language Attacks via Diversification Along the Intersection Region of Adversarial Trajectory. In *Computer Vision - ECCV 2024*, 442–460.

Ge, Y.; Xiao, Y.; Xu, Z.; Wang, X.; and Itti, L. 2022. Contributions of shape, texture, and color in visual recognition. In *European Conference on Computer Vision*, 369–386. Springer.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations*.

Grill-Spector, K.; and Malach, R. 2004. The human visual cortex. *Annu. Rev. Neurosci.*, 27(1): 649–677.

Gu, J.; Zhao, H.; Tresp, V.; and Torr, P. H. S. 2022. SegPGD: An Effective and Efficient Adversarial Attack for Evaluating and Boosting Segmentation Robustness. In *Computer Vision - ECCV*, volume 13689, 308–325.

Guo, Q.; Xie, X.; Juefei-Xu, F.; Ma, L.; Li, Z.; Xue, W.; Feng, W.; and Liu, Y. 2020. Spark: Spatial-aware online incremental attack against visual tracking. In *European conference on computer vision*, 202–219. Springer.

Hendrycks, D.; Zhao, K.; Basart, S.; Steinhardt, J.; and Song, D. 2021. Natural Adversarial Examples. In *IEEE Conference on Computer Vision and Pattern Recognition*, 15262–15271.

Huang, R.; Li, M.; Yang, D.; Shi, J.; Chang, X.; Ye, Z.; Wu, Y.; Hong, Z.; Huang, J.; Liu, J.; Ren, Y.; Zou, Y.; Zhao, Z.; and Watanabe, S. 2024. AudioGPT: Understanding and Generating Speech, Music, Sound, and Talking Head. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, 23802–23804.

Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; Sermanet, P.; Jackson, T.; Brown, N.; Luu, L.; Levine, S.; Hausman, K.; and Ichter, B. 2022a. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *Conference on Robot Learning*, volume 205, 1769–1782.

Huang, Y.; Sun, L.; Guo, Q.; Juefei-Xu, F.; Zhu, J.; Feng, J.; Liu, Y.; and Pu, G. 2022b. Ala: Naturalness-aware adversarial lightness attack. *arXiv preprint arXiv:2201.06070*.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32.

Jia, W.; Lu, Z.; Zhang, H.; Liu, Z.; Wang, J.; and Qu, G. 2022. Fooling the eyes of autonomous vehicles: Robust physical adversarial examples against traffic sign recognition systems. *arXiv preprint arXiv:2201.06192*.

Jia, X.; Tong, Y.; Qiao, H.; Li, M.; Tong, J.; and Liang, B. 2023. Fast and accurate object detector for autonomous driving based on improved YOLOv5. *Scientific Reports*, 13.

Jocher, G. 2020. Ultralytics YOLOv5.

Kong, D.; Liang, S.; Zhu, X.; Zhong, Y.; and Ren, W. 2024. Patch is enough: naturalistic adversarial patch against vision-language pre-training models. *Visual Intelligence*, 2(1): 33.

Lv, W.; Xu, S.; Zhao, Y.; Wang, G.; Wei, J.; Cui, C.; Du, Y.; Dang, Q.; and Liu, Y. 2023. DETRs Beat YOLOs on Real-time Object Detection. arXiv:2304.08069.

Mao, J.; Qian, Y.; Zhao, H.; and Wang, Y. 2023a. GPT-Driver: Learning to Drive with GPT. *CoRR*, abs/2310.01415.

Mao, J.; Shi, S.; Wang, X.; and Li, H. 2023b. 3D Object Detection for Autonomous Driving: A Comprehensive Survey. *International Journal of Computer Vision*, 131(8): 1909–1963.

Qin, Y.; Zhou, E.; Liu, Q.; Yin, Z.; Sheng, L.; Zhang, R.; Qiao, Y.; and Shao, J. 2024. MP5: A Multi-modal Open-ended Embodied System in Minecraft via Active Perception. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition+*, 16307–16316.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022a. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022b. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Sato, T.; Yue, J.; Chen, N.; Wang, N.; and Chen, Q. A. 2024. Intriguing Properties of Diffusion Models: An Empirical Study of the Natural Attack Capability in Text-to-Image Generative Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24635–24644. IEEE.

Shen, X.; Xiong, Y.; Zhao, C.; Wu, L.; Chen, J.; Zhu, C.; Liu, Z.; Xiao, F.; Varadarajan, B.; Bordes, F.; Liu, Z.; Xu, H.; J. Kim, H.; Soran, B.; Krishnamoorthi, R.; Elhoseiny, M.; and Chandra, V. 2024. LongVU: Spatiotemporal Adaptive Compression for Long Video-Language Understanding. *arXiv:2410.17434*.

Suryanto, N.; Kim, Y.; Kang, H.; Larasati, H. T.; Yun, Y.; Le, T.-T.-H.; Yang, H.; Oh, S.-Y.; and Kim, H. 2022. Dta: Physical camouflage attacks using differentiable transformation network. In *CVPR*, 15305–15314.

Tan, J.; Ji, N.; Xie, H.; and Xiang, X. 2021. Legitimate Adversarial Patches: Evading Human Eyes and Detection Models in the Physical World. In *ACMMM*, 5307–5315.

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

Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; and Ding, G. 2024a. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2024b. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Trans. Mach. Learn. Res.*

Wang, J.; Liu, A.; Yin, Z.; Liu, S.; Tang, S.; and Liu, X. 2021. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *CVPR*, 8565–8574.

Wang, N.; Luo, Y.; Sato, T.; Xu, K.; and Chen, Q. A. 2023a. Does physical adversarial example really matter to autonomous driving? towards system-level effect of adversarial object evasion attack. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4412–4423.

Wang, N.; Luo, Y.; Sato, T.; Xu, K.; and Chen, Q. A. 2023b. Does Physical Adversarial Example Really Matter to Autonomous Driving? Towards System-Level Effect of Adversarial Object Evasion Attack. In *IEEE/CVF International Conference on Computer Vision*, 4389–4400.

Wang, X.; Zhang, S.; Li, S.; Kallidromitis, K.; Li, K.; Kato, Y.; Kozuka, K.; and Darrell, T. 2024c. SegLLM: Multi-round Reasoning Segmentation. *arXiv:2410.18923*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*.

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

Xing, Y.; Guo, Q.; Cao, X.; Tsang, I. W.; and Ma, L. 2024. MetaRepair: Learning to Repair Deep Neural Networks from Repairing Experiences. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 1781–1790.

Xu, K.; Zhang, G.; Liu, S.; Fan, Q.; Sun, M.; Chen, H.; Chen, P.-Y.; Wang, Y.; and Lin, X. 2020. Adversarial t-shirt! evading person detectors in a physical world. In *ECCV*, 665–681. Springer.

Xue, H.; Araujo, A.; Hu, B.; and Chen, Y. 2023. Diffusion-Based Adversarial Sample Generation for Improved Stealthiness and Controllability. In *Advances in Neural Information Processing Systems*.

Yang, J.; Zhang, H.; Li, F.; Zou, X.; Li, C.; and Gao, J. 2023. Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V. arXiv:2310.11441.

Yang, L.; Yu, Z.; Meng, C.; Xu, M.; Ermon, S.; and Cui, B. 2024. Mastering Text-to-Image Diffusion: Recaptioning, Planning, and Generating with Multimodal LLMs. In *Forty-first International Conference on Machine Learning*.

Zhang, H.; Li, H.; Li, F.; Ren, T.; Zou, X.; Liu, S.; Huang, S.; Gao, J.; Leizhang; Li, C.; and Yang, J. 2024. LLaVA-Grounding: Grounded Visual Chat with Large Multimodal Models. In *Computer Vision - ECCV*, volume 15101, 19–35.

Zhang, S.; Cheng, Y.; Zhu, W.; Ji, X.; and Xu, W. 2023. {CAPatch}: Physical Adversarial Patch against Image Captioning Systems. In *32nd USENIX Security Symposium (USENIX Security 23)*, 679–696.

Zhang, Z.; and Zhang, A. 2024. You Only Look at Screens: Multimodal Chain-of-Action Agents. In *Findings of the Association for Computational Linguistics*, 3132–3149. Association for Computational Linguistics.

Zhao, J.; Zhao, W.; Deng, B.; Wang, Z.; Zhang, F.; Zheng, W.; Cao, W.; Nan, J.; Lian, Y.; and Burke, A. F. 2024. Autonomous driving system: A comprehensive survey. *Expert Syst. Appl.*, 242.

Zhao, Y.; Zhu, H.; Liang, R.; Shen, Q.; Zhang, S.; and Chen, K. 2019. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 1989–2004.

Zhong, Y.; Liu, X.; Zhai, D.; Jiang, J.; and Ji, X. 2022. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In *CVPR*, 15345–15354.