SAPatch: A Reinforcement Learning-based Framework for Spatial Adversarial Patch Attacks

Longpeng Hao[1], Libing Wu[2,*], Zhuangzhuang Zhang[3,*], Jiaqi Feng[4], Li Yi[5], and Taiyu Bao[6]
School of Cyber Science and Engineering, Wuhan University, Wuhan, China
2019300003066@whu.edu.cn, wu@whu.edu.cn, zhzhuangzhuang@whu.edu.cn,
jiaqiFeng@whu.edu.cn, yyiillii@whu.edu.cn, bty0717@whu.edu.cn

***Abstract*—With the extensive deployment of Deep Neural Networks in autonomous driving perception systems, the robustness of object detectors has become a crucial problem in public safety. Contemporary adversarial attacks targeting these detectors are predominantly categorized into global adversarial perturbations and adversarial patch attacks. Although adversarial patches have demonstrated significant potency against visual models, existing methodologies often rely on fixed strategies when handling discrete variables such as spatial positioning. Consequently, they struggle to reconcile attack success rates with physical realizability within complex, dynamic driving environments. To address this challenge, we propose a new attack framework named SAPatch, which adopts a bilevel optimization structure and combines reinforcement learning-based dynamic parameter tuning and adversarial patch texture generation. The adversarial patch within the effective target region is generated through a two-layer loop. In the outer loop, a reinforcement learning-based agent selects the optimal adversarial patch parameter settings according to a clean image to precisely attack the most sensitive semantic regions of object detectors. In the inner loop, a gradient-based optimization algorithm and a box-bound constraint mechanism are adopted to generate the adversarial texture with the best attack effect. The experiment results show that SAPatch has superior attack efficiency in various complex simulation environments highly simulated by CARLA and superior average attack accuracy on various mainstream object detectors, reflecting the severe vulnerability of existing autonomous driving perception systems.**

***Index Terms*—Autonomous Driving; Object Detection; Adversarial Patch; Intra-box Constraint; Reinforcement Learning; Gradient Loss**

I. Introduction

The advent of autonomous driving technology [1] represents an important milestone in modern transportation, and its evolution has seen artificial intelligence transform from its infancy as a mere concept to its practical application [2]. Early autonomous driving technologies employed mostly rule-based systems [3] and computer vision technologies [4]. Despite their relative success in some applications, these technologies often failed to perform satisfactorily in handling the complexities of the real world. The advent of DL technology, particularly the superiority of Deep Neural Networks (DNNs) in feature extraction [5] and pattern recognition, has seen autonomous driving technology grow exponentially. Today, autonomous driving technologies are gradually evolving from driver assistance technologies to fully autonomous systems [6], with the goal of greatly enhancing road safety and efficiency by reducing the level of human error. However, with the advent of this technology in the real world, its safety and reliability have become major concerns to both the academic and business communities.

In the autonomous driving stack [7], the perception module acts as a sensory interface [8] for real-time data acquisition and interpretation in the environment [9]. Object detection has been a cornerstone in this regard. Modern autonomous systems utilize multi-modal sensor configurations with LiDAR [10], millimeter wave radar [11], and high-definition cameras [12]. Among these, camera systems have been highly valued due to their inherent ability to capture rich textural features in images. Using deep convolutional neural networks, prominent object detectors like YOLO and Faster R-CNN analyze images to understand objects [13] like cars, pedestrians, and traffic signs. At the same time, these detectors also estimate the spatial coordinates of objects using bounding box regression. Thus, these detectors demonstrate a high degree of semantic understanding of objects in images. The predictive accuracy and adversarial robustness of these object detectors are critical determinants of the safety of autonomous vehicles.

Despite the outstanding performance of DNNs in object detection, empirical research has revealed their significant susceptibility to adversarial attacks [14], thus making object detectors highly vulnerable to adversarial examples [15]. Present research on adversarial attacks includes various methodologies [16], with the first being the global infinitesimal perturbation attack. This type of adversarial attack [17] directly affects the object detection process through the injection of imperceptible noise in the pixel level of the image, thus causing the object detection model to produce false predictions. This type of adversarial attack has shown outstanding success rates in the digital domain. The second type of adversarial attack is the Adversarial Patch attack [18], which places no restrictions on the level of the perturbation but rather relies on the execution of the adversarial attack through the superimposition of image patches with adversarial textures on the image. Unlike the global infinitesimal perturbation, the adversarial patch is more robust to physical distortions [19] such as lighting, distance,

This work was supported by the National Natural Science Foundation of China (62441237, U24A20336, 62272348, U22B2022), and Wuhan City Joint Innovation Laboratory for Next-Generation Wireless Communication Industry Featuring Satellite-Terrestrial Integration under Grant 4050902040448.

and angle, thus becoming a standard adversarial attack for object detection models. With respect to the security risks faced by object detection models in the context of autonomous vehicles, the following are the contributions of this paper:

- We develop a unified framework that incorporates the unique attack modes of Impersonation and Dodging. By defining unique reward functions for each type of attack, we ensure that the reinforcement learning agent receives reliable feedback that facilitates accurate policy updates.
- We propose a strict mask method, in which the agent must place the patches in the effective area of the target object, ensuring the physical validity and stealthiness of the attack.
- We propose a framework that uses outer-layer reinforcement learning policy optimization and inner-layer gradient texture generation. The outer loop allows the agent to optimize the hyperparameters of the adversarial patches, and the inner loop uses a gradient descent method to generate the texture of the adversarial patches.

II. Related Work

A. Adversarial Attacks

Although Deep Neural Networks perform remarkably well in computer vision tasks, the threat of adversarial examples poses a major security threat. Traditionally, attacks like FGSM, PGD, and C&W have been based on digital-domain global perturbations. In these attacks, infinitesimal norm-bounded noises [20] are applied over the entire pixel space. However, recent research suggests that these attacks are not robust in physical attacks [21] because the camera imaging process is affected by lighting changes, viewpoint changes [22], and signal quantization [23]. As a result, the research community is now focused on robust physical attacks. Among these attacks, adversarial patches have gained prominence as a major threat. Unlike global noises, patches can be applied with arbitrary pixel modifications within a region using strong feature saliency. Interestingly, the "Adversarial Sticker" method showed that semantic texture is a promising attack vector. In addition, the method also showed the phenomenon of "Regional Aggregation," where successful attacks are spatially correlated.

B. Black-box and White-box Attacks

Adversarial attacks are classified based on the knowledge level of the attacker. In white-box attacks, the attacker [24] has full knowledge of the architecture and model parameters, which allows for the efficient generation of adversarial examples using backpropagation. However, in most realistic cases, like autonomous driving, it is not feasible to obtain such knowledge, making black-box attacks the most viable option. Black-box attacks [25] can be broadly classified as query-based or transfer-based. In query-based attacks, zeroth-order optimization is employed to approximate gradient information. However, query-based attacks are hindered by query budgets and latency. In transfer-based attacks, the cross-model generalization of adversarial examples [26] is employed. To ensure maximum generalization of the attack, ensemble-based attack strategies are employed. These are of particular interest in assessing the security of autonomous driving systems, as a rigorous framework can be employed to assess the security of perception systems [27] in the absence of model parameters. Using gradient information from multiple surrogates [28] and dynamic weighting of the individual models, common features of vulnerability are captured, thus constraining the search space.

C. Application of Reinforcement Learning in Adversarial Attacks

Reinforcement learning [29] is increasingly being used in adversarial attacks. The main motivation is to overcome the non-differentiable barriers that come with physical attacks [30]. Reinforcement learning is helping to overcome these barriers because it does not rely on gradients. Instead, it considers the attack as a Markov Decision Process. The agent is able to interact with the environment and learn the best attack strategy [31]. Initially, people were taking a two-step approach. First, they would choose the texture from a predefined set. Then, they would choose the location of the patch [32]. The relationship between the location of the object in space and the texture it is conveying is very important [33]. A good perturbation usually resembles the features of the region it is in. Recently, there has been a shift towards "Simultaneous Optimization", which considers the location of the object and the texture settings (such as step size and weights) as a whole. The SAPatch framework is an example of this.

III. Methodology

In this section, we present a brief overview of our proposed unified adversarial patch attack framework using reinforcement learning, as shown in Fig 1. Additionally, we present the formal definition of the Impersonation and Dodging attack paradigms, as well as the strict intra-box constraint mechanism.

A. Overview

While adversarial patch attacks have been useful in testing the robustness of Deep Neural Networks, using them in a real-world setting as in autonomous driving in dynamic environments has remained very challenging. The primary issue is that the spatial position parameters are discrete and non-differentiable. This makes it difficult to optimize them directly using gradient-based methods.

In order to address the above challenges, we propose a framework for adversarial patching attacks, which we refer to as SAPatch. It supports two types of attacks: impersonation and camouflage, while satisfying in-box constraints, which limit the placement of the patch within the effective region of the target. In addition, we also propose a bi-level optimization paradigm, in which the outer loop iteratively optimizes the hyperparameters through reinforcement learning to find better hyperparameters, while the inner loop continuously generates

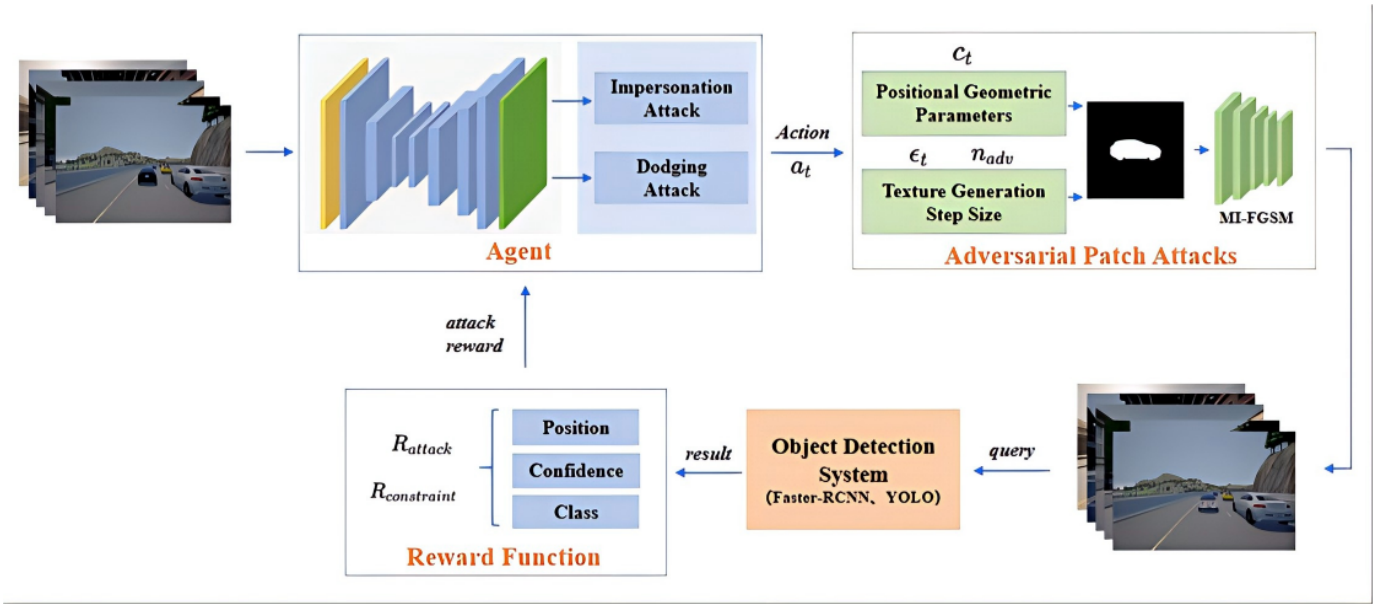


Fig. 1: The Overview of the Bi-level Optimization Framework for Adversarial Patch Generation.

the adversarial textures using the MI-FGSM method. It aims to integrate the discrete spatial parameter search with the continuous texture generation into a whole optimization process.

B. Attack Types and Definitions

The core objective of an adversarial attack is to superimpose an adversarial patch on the input image $x \in \mathbb{R}^{H\times W\times C}$ such that the target model $f(\cdot)$ outputs an incorrect prediction result. Let y_{true} be the Ground Truth of the input image.We define a binary mask matrix $\mathcal{A} \in \{0,1\}^{H\times W}$ to determine the spatial position and shape of the patch in the image. When $\mathcal{A}_{i,j} = 1$, the pixel at that location is replaced by the patch texture; otherwise, the original pixel is retained. The generation process of the adversarial example x_{adv} can be expressed as an element-wise operation:

$$x_{adv} = (1 - \mathcal{A}) \odot x + \mathcal{A} \odot \tilde{x} \tag{1}$$

Where $\odot$ denotes the Hadamard Product, and $\tilde{x}$ is the patch texture tensor mapped to the image coordinate system after geometric transformations.

Impersonation Attack: The Impersonation Attack constitutes a targeted adversarial paradigm wherein the objective is to coerce the detector into misclassifying a specific target object O_{target} as an attacker-specified category y_{target} . In this mode, the primary focus is not on suppressing the confidence of the original class, but rather on maximizing the confidence of the target class. Consequently, we define the loss function for the impersonation attack, denoted as $\mathcal{L}_{imp}$, as the negative log-likelihood of the target class probability:

$$\mathcal{L}_{\text{imp}} = -\log\left(\max_{i\in\mathcal{D}} P(c_i = y_{target}|b_i)\right) \tag{2}$$

where $\mathcal{D}(x_{adv})$ represents the set of all candidate bounding boxes output by the detector on the adversarial image.

Dodging Attack: The Dodging Attack is an untargeted adversarial attack aimed at reducing the confidence level of the model in the correct category. If the confidence level falls below the threshold, the detector fails to detect the object, and a false negative is recorded. There are two possibilities in which the Dodging Attack can cause a false negative: it can happen when the confidence level falls below the threshold, or it can happen when the bounding box is displaced to such an extent that object detection becomes difficult. As the effect of geometric displacement is minimal in the intra-box constraint, our main focus is to reduce the confidence level. Unlike the Impersonation Attack, in the Dodging Attack, our focus is on minimizing the maximum confidence level in the correct category. We define the loss function corresponding to the dodging attack as $\mathcal{L}_{dod}$ and formulate it as follows:

$$\mathcal{L}_{\text{dod}} = \max\left(\max_{i\in\mathcal{D},|a_i|>\tau} P(c_i = y|b_i) - \epsilon, 0\right) \tag{3}$$

where ϵ denotes a minimal safety margin. This formulation essentially functions as a truncated Hinge Loss.

C. Strict Intra-box Constraint Mechanism

In order to ensure that the attack is not only physically feasible but also semantically meaningful, we also impose an important geometrical constraint: the adversarial patch should remain strictly inside the legal area of the object under attack. Unlike many other attacks that seek to clutter the background in order to produce false positives by contextual means, SAPatch instead focuses on putting the patch inside the object's bounding box, in the attempt to mess with the object's feature extraction.

In order to maintain the attack plausibly realistic in the real world and semantically coherent, we impose a strict geometric constraint during optimization: the adversarial patch

must remain completely within the valid bounding box of the target object. This makes our approach differ from most other existing methods, which introduce perturbations to the background to affect context and trigger false positives. In contrast, SAPatch adopts an object-centered strategy, placing the adversarial patch directly on the surface of the target object. In doing so, it directly attacks the detector's internal feature extraction process, undermining the representation of the object itself rather than just polluting the background.

Valid Region Mask Generation. For a given input image, we first derive a binary validity mask, denoted as $M_{valid} \in \{0,1\}^{H\times W}$, utilizing ground truth bounding boxes or keypoint detection algorithms. In the context of autonomous driving object detection, M_{valid} represents the pixel-wise region within the target's bounding box. Formally, the mask is defined as:

$$M_{valid}(i,j) = \begin{cases} 1, & \text{if pixel } (i,j) \in \mathcal{R}_{valid} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{R}_{valid}$ denotes the allowable pasting region.

Action Space Masking. To make sure that the reinforcement learning agent only samples from the valid spatial parameters, we directly apply a mask on the policy's action space. After we get a probability map or logits from the position head, we set all probabilities to zero or set all logits to - in the invalid regions where $M_{valid}(i,j) = 0$. This ensures that the sampled coordinate (c_x, c_y) will be within a legitimate region $\mathcal{R}_{valid}$.

Constraint Enforcement. The core constraint of SAPatch requires the generated patch mask $\mathcal{A}$ to be a strict subset of the validity mask M_{valid}. Mathematically, this condition is expressed as:

$$\mathcal{A} \odot M_{valid} = \mathcal{A} \tag{5}$$

where $\odot$ denotes the Hadamard product.

IV. System Model

In this section, we have provided a comprehensive elaboration of the proposed reinforcement learning-based bi-level adversarial patch generation framework.

A. Bi-level Optimization Framework

SAPatch is based on a bi-level optimization paradigm facilitated by Deep Reinforcement Learning. This architecture strategically delegates the non-differentiable optimization of spatial configurations and hyperparameters to the RL agent, while entrusting the high-dimensional texture synthesis to efficient gradient descent methods (MI-FGSM). Our proposed SAPatch achieves the complementary advantages of combining two approaches:

- The Outer Loop is orchestrated by the reinforcement learning agent which observes the current state of the input image to predict a set of optimal actions.
- The Inner Loop is executed by a gradient-guided adversarial generator that synthesizes the adversarial patch conditioned on the hyperparameters prescribed by the agent.

B. Outer Loop: Strategy Optimization Based on Deep Reinforcement Learning

Following the training procedure outlined in Algorithm 1, the Outer Loop seeks an optimal attack configuration by instructing the agent to forecast a given action set a_t. This set consists of the spatial location c_t, the texture step size ϵ_t, and the weights w_t, all of which depend on the input image. The optimization process occurs within a hybrid reward framework that seeks to enhance adversarial attack efficacy by incorporating attack-specific objectives, as well as maintain physical validity by satisfying strong intra-box geometric constraints.

Impersonation attacks are all about increasing the confidence level of the target category y_{target}. To achieve this, we combine a progressive incentive scheme with a success bonus at the end of the process. The reward is given by:

$$\mathrm{R}_{imp}^{(t)} = \begin{cases} \lambda_1(p_{target}^{(t)} - p_{target}^{(t-1)}) & \text{if attack fails} \\ C_{success} + p_{target}^{(t)} & \text{if } \arg\max(O) = y_{target} \end{cases} \tag{6}$$

Dodging Attack Reward. Dodging attacks seek to suppress the true class confidence y_{true}. The reward structure combines the suppression of confidence with the term that represents the vanish outcome, defined as:

$$R_{dod}^{(t)} = \begin{cases} \lambda_2(p_{true}^{(t-1)} - p_{true}^{(t)}) & \text{if detection box persists} \\ C_{vanish} & \text{if target vanishes} \end{cases} \tag{7}$$

Intra-Box Constraint Reward. To ensure physical feasibility, we restrict patch generation strictly within the target's semantic region. Deviations from the ground truth box incur a geometric penalty $R_{constraint}$, defined as:

$$R_{constraint} = \begin{cases} -\lambda \cdot \frac{S_{outside}}{S_{overlap}} & \text{if } S_{overlap} > 0 \\ -C_{miss} & \text{if } S_{overlap} = 0 \end{cases} \tag{8}$$

where $S_{overlap}$ denotes the patch-ground truth intersection, and $S_{outside}$ represents the overflow area.

C. Inner Loop: Adversarial Texture Generation Based on Ensemble Gradients

In Algorithm 2, textures that are adversarial in nature are created using the Momentum Iterative Fast Gradient Sign Method (MI-FGSM). The momentum helps stabilize the path and also increases the transferability of the textures, and the generation is controlled by three hyperparameters provided by the agent.

Spatial Configuration c_t. This parameter defines the patch's location (c_x, c_y) and dimensions (c_a, c_b). WWe focus on the configuration in space, finding the probability distribution of the position, as follows:

$$P_{position} = \text{softmax}\left(\frac{1}{n}\sum_{i=1}^{n} M_i\right) \tag{9}$$

Texture Generation Step Size ϵ_t. This controls the magnitude of the gradient update during texture generation. To balance optimization speed with visual quality, the agent

Algorithm 1 SAPatch: Spatial Adversarial Patch

Input: Training dataset $\mathcal{D}$, Target Detector $F(\cdot)$, Max episodes M, Policy parameters θ
Output: Trained policy network π_θ

Initialize policy network parameters θ, Baseline $b \leftarrow 0$
for episode $m = 1$ to M **do**
  Sample image x and ground truth y_{true} from $\mathcal{D}$
  for step $t = 1$ to T_{max} **do**
    Get state s_t
    Action $a_t = \{c_t, \epsilon_t, \mathbf{w}_t\}$ via policy π_θ
    Generate optimal texture p^*
      $p^* \leftarrow \text{InnerLoop}(x, c_t, \epsilon_t, \mathbf{w}_t)$
    Call Algorithm 2
    Attack $x_{adv} = (1 - \mathcal{A}) \odot x + \mathcal{A} \odot \mathcal{T}(p^*, c_t)$
    Query black-box $F(x_{adv})$ to get P and B
    Calculate reward $R_t = R_{type} + R_{constraint}$
    Store transition tuple $\tau = (s_t, a_t, R_t)$
  end for
  Update Policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$
  Update baseline b
end for
return π_θ

selects an optimal step size from a predefined discrete set $\mathcal{E} = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_K\}$.

Model Ensemble Weights $\mathbf{w}_t$. We approximate the target gradients using an ensemble of N white-box surrogate models $\{f_1, f_2, \ldots, f_N\}$. The agent dynamically assigns a weight vector $\mathbf{w}_t$ to determine the contribution of each model, derived via the following formulation:

$$w_i = \frac{e^{\rho_i}}{\sum_{j=1}^{N} e^{\rho_j}} \quad \text{where } \rho_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \tag{10}$$

D. Policy Update Mechanism

Once the adversarial patch is generated, we optimize the agent's parameters θ utilizing the REINFORCE algorithm within the Policy Gradient framework. This optimization aims to maximize the cumulative expected reward $J(\theta)$ by effectively integrating the multi-component incentive signals derived from the adversarial evaluation. The update of the parameters is approximated using the following gradient:

$$\nabla_\theta J \approx \sum_t (R_t - b) \nabla_\theta \log \pi(a_t | s_t) \tag{11}$$

where b represents a baseline term introduced to reduce the variance of the gradient estimation.

V. Experimental

In this section, we delineate the experimental setup, dataset specifications, and the comprehensive evaluation metrics we used to evaluate the performance of the proposed model.

Algorithm 2 Adversarial Texture Generation

Input: Clean image x, Action parameters $\{c_t, \epsilon_t, \mathbf{w}_t\}$, Substitute models $\{f_1, \ldots, f_N\}$, Iterations K
Output: Adversarial patch texture p^*

Texture p_0, Momentum $g_0 = \mathbf{0}$
for $k = 0$ to $K - 1$ **do**
  Adversarial Image p_k
    $x_k^{adv} = (1 - \mathcal{M}(c_t)) \odot x + \mathcal{M}(c_t) \odot \mathcal{T}(p_k, c_t)$
  Calculate Ensemble Loss
    $\mathcal{L}_{ens} = \sum_{n=1}^{N} w_{t,n} \cdot \mathcal{L}_{f_n}(x_k^{adv}, y)$
  Compute Gradient $\nabla_p \mathcal{L} = \nabla_p \mathcal{L}_{ens}(x_k^{adv}, y)$
  Update Momentum
    $g_{k+1} = \mu \cdot g_k + \frac{\nabla_p \mathcal{L}}{\|\nabla_p \mathcal{L}\|_1}$
  Update Texture
    $p_{k+1} = \text{Clip}_{[0,1]}(p_k - \epsilon_t \cdot \text{sign}(g_{k+1}))$
end for
return $p^* = p_K$

A. Experimental Setup

Datasets. The experiments are based on the data generated by the CARLA simulator, which is a high-fidelity, open-source simulator for autonomous driving research. This 3D virtual environment enables the synthesis of diverse training scenarios by flexibly configuring parameters such as capture viewpoints and object distances. To evaluate the performance of the proposed SAPatch method, we construct a dataset based on the specifications in Table I.

Models. In order to show the flexibility of our approach, we choose three popular object detection methods, commonly used in the industry and academia, as the victim object detection methods. The victim object detection methods we chose are Faster R-CNN, YOLOv3, YOLOv5, and YOLOv8, and we start them from their official configurations.

Evaluation Metrics. To evaluate the effectiveness of the SAPatch framework, we use Mean Average Precision (mAP) and Attack Success Rate (ASR) as the primary metrics for evaluation. mAP is defined as the average precision of all the categories. With respect to ASR, the definition of ASR is dependent on the attack modality. For Impersonation Attacks, ASR is defined as the percentage of samples successfully misclassified as the particular target category, while for Dodging Attacks, ASR is defined as the percentage of samples for which the confidence level is below 0.5.

TABLE I: the CARLA autonomous driving dataset

Scenario Type	Weather Conditions	Number of Images	Usage
Urban Road	Sunny/Rainy/Foggy	750	Training/Testing
Highway	Sunny/Rainy/Foggy	800	Training/Testing

B. Comparative Analysis of Adversarial Patch Efficacy

To evaluate the robustness of SAPatch against adversarial attacks, we measure its performance in terms of Mean Average Precision (mAP) and Attack Success Rate (ASR). The results

TABLE II: Comparison of different attack methods

Target Model	2*Method	Impersonation Attack		Dodging Attack	
		mAP (%) ↓	ASR (%) ↑	mAP (%) ↓	ASR (%) ↑
YOLOv3	Clean Data	87.21	-	87.21	-
	AdvPatch	56.45	68.32	48.24	76.55
	Adv-Sticker	49.88	75.64	41.35	84.12
	GDPA	43.52	82.16	35.78	89.47
	DOEPatch	37.81	88.24	28.36	94.18
	SAPatch (Ours)	**35.92**	**89.43**	**26.54**	**95.34**
YOLOv5	Clean Data	89.52	-	89.52	-
	AdvPatch	62.37	61.28	55.42	70.56
	Adv-Sticker	56.12	69.85	48.90	79.32
	GDPA	50.48	77.46	42.43	85.17
	DOEPatch	44.52	84.15	36.17	91.03
	SAPatch (Ours)	**43.18**	**85.24**	**34.86**	**92.12**
YOLOv8	Clean Data	93.54	-	93.54	-
	AdvPatch	72.31	51.05	64.92	61.08
	Adv-Sticker	65.38	60.47	57.21	71.34
	GDPA	60.54	67.12	52.78	77.56
	DOEPatch	55.22	74.81	46.41	86.85
	SAPatch (Ours)	**53.47**	**76.12**	**44.72**	**88.03**
Faster R-CNN	Clean Data	91.43	-	91.43	-
	AdvPatch	77.12	40.54	71.65	49.82
	Adv-Sticker	72.76	47.88	67.11	57.95
	GDPA	68.41	55.26	60.84	65.31
	DOEPatch	62.15	66.83	54.81	77.54
	SAPatch (Ours)	**60.84**	**67.96**	**53.27**	**78.60**

are compared with the state-of-the-art methods. The numerical results of different architectures are shown in Table II.

1) Analysis of Object Detection Models: We evaluate our approach across various detectors by benchmarking mean Average Precision and Attack Success Rate. A comparative analysis of SAPatch against existing methods is presented, as detailed in Table II. Our experiments on Faster R-CNN, YOLOv3, YOLOv5, and YOLOv8 reveal that our adversarial patch is significantly more effective against the YOLO series than Faster R-CNN. We attribute this to YOLO's single-stage paradigm, which predicts targets directly from image grids, making it highly vulnerable when critical regions are occluded. Conversely, Faster R-CNN's two-stage mechanism demonstrates greater inherent resilience against such attacks.

2) Attack Success Rate Comparison: The experimental results demonstrate that both proposed attack modalities achieve high Attack Success Rates across all four evaluated detection models. Taking the YOLOv5 object detector as an instance, the ASR for the impersonation attack is 85.24%, whereas the ASR for the dodging attack reaches 92.12%. Overall, the dodging attack consistently exhibits a higher ASR than the impersonation attack. We attribute this discrepancy to the fundamental objective of the impersonation attack, which aims to mislead the detection model into classifying the target as a specific, predefined category—a task inherently more challenging than merely causing the detector to evade detecting the object entirely.

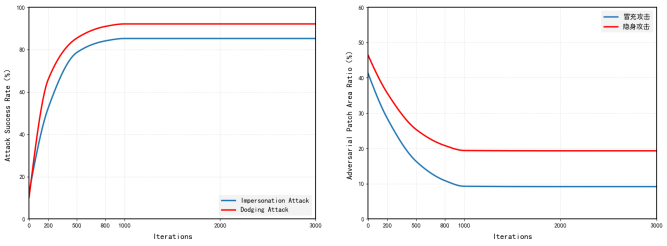
TABLE III: Attack Success Rate (ASR) comparison with different number of iterations.

Attack Type	$K = 3$	$K = 6$	$K = 8$	$K = 10$	$K = 15$	$K = 20$
Impersonation Attack	35.27%	62.58%	76.23%	83.51%	83.26%	83.14%
Dodging Attack	42.18%	70.34%	85.47%	90.82%	90.15%	90.09%

C. Convergence Analysis of Inner Loop Texture Synthesis

We evaluate the convergence behavior of the ensemble gradient-based adversarial texture synthesis within the inner loop. With the hyperparameters from the outer loop held constant, we investigate the impact of the gradient-based iteration count on attack performance. The correlation between the Attack Success Rate and the number of inner loop iterations is detailed in Table III.

Disparity in Convergence Rates. Both impersonation and dodging attacks demonstrate a substantial surge in Attack Success Rate at the 6th optimization iteration. The ASR reaches its peak at the 10th iteration, after which further increases in the iteration count yield only marginal variations in attack efficacy. To strike an optimal balance between the adversarial performance of the generated textures and the computational overhead, we ultimately set the number of optimization iterations for the inner adversarial texture to 10.



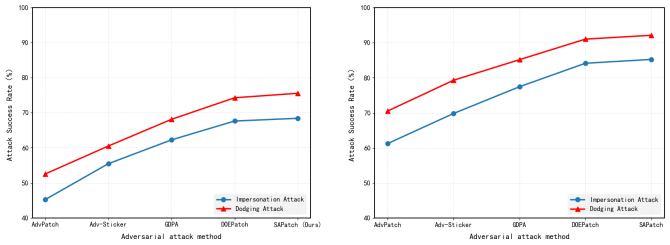
(a) Attack Success Rate (b) Adversarial Patch Area Ratio

Fig. 2: Performance Comparison of Reinforcement Learning Agents over Training Iterations

D. Analysis of Outer Loop Attack Efficacy

We investigate the interaction dynamics between the agent and the environment within the Outer Loop, analyzing the evolution of attack efficacy across training epochs as the agent learns to optimize the adversarial patch's spatial configuration and the inner-loop step size.

1) Agent Learning Capability: The experimental results show that, as shown in Fig 2(a), there is a notable increasing trend in the Attack Success Rate of impersonation and dodging attacks as the number of training epochs increases. As the number of iterations in the training process approaches 1,000, the ASR metrics for impersonation and dodging attacks start to plateau. However, as the number of epochs increases beyond 1,000, the performance of the agent is observed to deteriorate, which is a result of overfitting in the policy of the agent.



(a) Urban Road (b) Highway

Fig. 3: Performance Comparison of Reinforcement Learning Agents over Training Iterations

2) Differential Evolution of Patch Size: The area proportion of the patch in the impersonation attack gradually converges from an initial 41.23% to a significantly smaller 9.27%, as shown in Fig 2(b). This occurs because impersonation attacks often depend on specific, highly discriminative feature regions; the agent discovers that shrinking the patch and focusing on these critical points achieves maximum target confidence with minimal perturbation cost, which aligns with our expectation for precise feature manipulation. Conversely, the patch area proportion for the dodging attack ultimately stabilizes at approximately 19.41%, noticeably larger than that of the impersonation attack. We attribute this to the fact that, to successfully render the object undetectable, the patch must occlude or disrupt a broader extent of the object's primary features. Consequently, the agent is inclined to retain a relatively larger area to obscure more of the target's visual information.

E. Robustness Analysis Across Environmental Variations

To evaluate the transferability of the proposed SAPatch method across diverse scenarios, weather conditions, and observation angles, we conducted corresponding comparative experiments.

1) Efficacy Evaluation Across Traffic Scenarios: Through evaluations in both highway and urban traffic scenarios, we observe that the adversarial patch attack exhibits substantial efficacy across both environments, as shown in Fig 3. Compared to the urban traffic scenario, the highway environment features significantly less occlusion among traffic participants; consequently, the Attack Success Rate of the adversarial patch in this setting is generally higher by approximately 28%.

2) Efficacy Evaluation Across Weather Conditions: Through evaluations across three distinct weather conditions, namely Sunny, Rainy, and Foggy, we observe that the adversarial patch attack achieves the highest Attack Success Rate under Sunny conditions, as detailed in Table IV. We attribute this phenomenon to the fact that under adverse weather conditions, the image data captured by the camera undergoes significant degradation and blurring. Consequently, the discriminative visual features extracted by the object detector are substantially diminished, which ultimately leads to a precipitous decline in the overall efficacy of the adversarial patch attack.

TABLE IV: Comparison of attack performance (ASR) under different weather conditions.

Weather	Attack Type	AdvPatch	Adv-Sticker	GDPA	DOEPatch	SAPatch
Sunny	Impersonation Attack	61.28%	69.85%	77.46%	84.15%	**85.54%**
	Dodging Attack	70.56%	79.32%	85.17%	91.03%	**92.12%**
Rainy	Impersonation Attack	29.17%	41.26%	48.64%	58.12%	**71.39%**
	Dodging Attack	38.45%	48.73%	55.29%	65.41%	**78.56%**
Foggy	Impersonation Attack	15.82%	22.41%	31.55%	44.18%	**61.27%**
	Dodging Attack	24.31%	32.16%	41.87%	52.32%	**68.43%**

TABLE V: Attack Success Rate (ASR) comparison at different viewing angles.

Attack Type	-45°	-30°	-15°	0°	15°	30°	45°
Impersonation Attack	67.89%	77.05%	82.91%	86.85%	83.24%	76.42%	68.55%
Dodging Attack	76.13%	84.71%	89.84%	93.62%	90.15%	84.35%	75.82%

3) Efficacy Evaluation Across Observation Angles: Through evaluations across varying observation angles, we observe that the adversarial patch achieves the highest attack efficacy in images captured from a frontal viewpoint, as detailed in Table V. A progressive enlargement of the observation angle leads to a continuous reduction in the Attack Success Rate of the adversarial patches. We attribute this performance degradation to the geometric distortions experienced by the adversarial patches within the image data captured by the vehicle-mounted camera; as the viewing angle shifts, these perspective deformations inherently diminish the overall effectiveness of the adversarial perturbations.

VI. Conclusion

This paper proposes SAPatch, a novel two-stage optimization framework engineered for the robust synthesis of adversarial patches. Through the construction of a collaborative mechanism between the outer layer of reinforcement learning policy optimization and the inner layer of gradient texture generation, the framework efficiently generates adversarial patches for both impersonation and dodging attacks. Meanwhile, we integrate a rigorous intra-box spatial constraint mechanism that strictly confines adversarial perturbations within the semantic boundaries of the target object, thereby significantly enhancing the physical plausibility of the patches. The performance of the proposed SAPatch is evaluated across disparate object detection models within the high-fidelity CARLA simulation platform. The experimental results demonstrate that SAPatch can effectively implement impersonation and dodging attacks, exhibiting substantial attack efficacy across diverse traffic scenarios, weather conditions, and observation angles, thus providing a formidable adversarial tool for evaluating the security of autonomous driving perception systems.

References

[1] J. Liu, X. Gong, T. Wang, Y. Hu, and H. Chen, "A proxy-data-based hierarchical adversarial patch generation method," *Computer Vision and Image Understanding*, vol. 246, p. 104066, 2024.

[2] J. Wang, F. Li, and L. He, "A unified framework for adversarial patch attacks against visual 3d object detection in autonomous driving," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 5, pp. 4949–4962, 2025.

[3] K. N. T. Nguyen, W. Zhang, K. Lu, Y.-H. Wu, X. Zheng, H. Li Tan, and L. Zhen, "A survey and evaluation of adversarial attacks in object detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 9, pp. 15 706–15 722, 2025.

[4] Y. Wang, Z. Zeng, T. Guan, W. Yang, Z. Chen, W. Liu, L. Xu, and Y. Luo, "Adaptive patch deformation for textureless-resilient multi-view stereo," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 1621–1630.

[5] J. Li, Z. Wang, and J. Li, "Advdenoise: Fast generation framework of universal and robust adversarial patches using denoise," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 3481–3490.

[6] Z. Cheng, Z. Liu, T. Guo, S. Feng, D. Liu, M. Tang, and X. Zhang, "Badpart: Unified black-box adversarial patch attacks against pixel-wise regression tasks," in *Forty-first International Conference on Machine Learning*, 2024.

[7] R. Ran, J. Wei, C. Zhang, G. Wang, Y. Yang, and H. T. Shen, "Adaptive multi-scale degradation-based attack for boosting the adversarial transferability," *IEEE Transactions on Multimedia*, vol. 26, pp. 10 979–10 990, 2024.

[8] G. Tang, W. Yao, C. Li, T. Jiang, and S. Yang, "Black-box adversarial patch attacks using differential evolution against aerial imagery object detectors," *Engineering Applications of Artificial Intelligence*, vol. 137, p. 109141, 2024.

[9] D. Wang, C. Li, S. Wen, Q.-L. Han, S. Nepal, X. Zhang, and Y. Xiang, "Daedalus: Breaking non-maximum suppression in object detection via adversarial examples," 2020.

[10] P. N. Williams and K. Li, "Black-box sparse adversarial attack via multi-objective optimisation cvpr proceedings," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12 291–12 301.

[11] P. Shekhar, B. Devkota, D. Samaraweera, L. N. Kandel, and M. Babu, "Do adversarial patches generalize? attack transferability study across real-time segmentation models in autonomous vehicles," in *2025 IEEE Security and Privacy Workshops (SPW)*, 2025, pp. 322–328.

[12] W. Tan, Y. Li, C. Zhao, Z. Liu, and Q. Pan, "Doepatch: Dynamically optimized ensemble model for adversarial patches generation," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 9039–9054, 2024.

[13] P. Guo, C. Gong, X. Lin, F. Liu, Z. Lu, Q. Zhang, and Z. Wang, "Mos-attack: A scalable multi-objective adversarial attack framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 5041–5051.

[14] X. Li, Y. Zhu, Y. Huang, W. Zhang, Y. He, J. Shi, and X. Hu, "Pbcat: Patch-based composite adversarial training against physically realizable attacks on object detection," 2025.

[15] Y. Zhang, J. Chen, Z. Peng, Y. Dang, Z. Shi, and Z. Zou, "Physical adversarial attacks against aerial object detection with feature-aligned expandable textures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–15, 2024.

[16] Q. Guo, X. Jia, S. Pang, S. Qin, L. Wang, J. Jia, Y. Liu, and Q. Guo, "Physpatch: A physically realizable and transferable adversarial patch attack for multimodal large language models-based autonomous driving systems," 2025.

[17] K. Oonishi and T. Nakai, "Universal shape of strong remote adversarial patches for object detection with convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2025, pp. 4369–4379.

[18] X. Wei, Y. Guo, and J. Yu, "Adversarial sticker: A stealthy attack method in the physical world," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 2711–2725, 2023.

[19] H. P. Silva, F. Becattini, and L. Seidenari, "Attacking attention of foundation models disrupts downstream tasks," 2025.

[20] A. Nazeri, C. Zhao, and P. Pisu, "Evaluating the adversarial robustness of detection transformers," 2024.

[21] W. Jiang, D. K. Jangid, S.-J. Lee, and H. R. Sheikh, "Latent patched efficient diffusion model for high resolution image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2025, pp. 6426–6432.

[22] Y. Wang, L. Wu, Y. Cao, J. Jin, Z. Zhang, E. Wang, C. Ma, and Y. Zhao, "A highly transferable camouflage attack against object detectors in the physical world," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 7, pp. 10 373–10 385, 2025.

[23] Y. Yu, W. Han, L. Wu, B. Liu, E. Wang, and Z. Zhang, "Enduring, efficient and robust trajectory prediction attack in autonomous driving via optimization-driven multi-frame perturbation framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 17 229–17 238.

[24] W. Yang, G. Liu, and D. Ming, "Smp-attack: Boosting the transferability of feature importance-based adversarial attack with semantics-aware multi-granularity patchout," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2025, pp. 4444–4454.

[25] J. Peng, Z. Tao, H. Wang, M. Wang, and Y. Wang, "Boosting adversarial transferability via residual perturbation attack," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2025, pp. 1261–1270.

[26] X. Wang and K. He, "Enhancing the transferability of adversarial attacks through variance tuning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 1924–1933.

[27] X. Sun, G. Cheng, H. Li, C. Lang, and J. Han, "Stdatav2: Accessing efficient black-box stealing for adversarial attacks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 4, pp. 2429–2445, 2025.

[28] G. Shi, Z. Lin, A. Peng, and H. Zeng, "An enhanced transferable adversarial attack against object detection," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 1–7.

[29] J. Zhang, J. Ye, X. Ma, Y. Li, Y. Yang, Y. Chen, J. Sang, and D.-Y. Yeung, "Anyattack: Towards large-scale self-supervised adversarial attacks on vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 19 900–19 909.

[30] J. Chen, H. Chen, K. Chen, Y. Zhang, Z. Zou, and Z. Shi, "Diffusion models for imperceptible and transferable adversarial attack," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 2, pp. 961–977, 2025.

[31] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[32] M. Lee and D. Kim, "Robust evaluation of diffusion-based adversarial purification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 134–144.

[33] Y. Wang, L. Wu, J. Jin, E. Wang, Z. Zhang, and Y. Zhao, "An imperceptible adversarial attack against 3-d object detectors in autonomous driving," *IEEE Internet of Things Journal*, vol. 12, no. 12, pp. 21 404–21 414, 2025.