Scale-free and Task-agnostic Attack: Generating Photo-realistic Adversarial Patterns with Patch Quilting Generator

Xiangbo Gao
University of California, Irvine
xiangbog@uci.edu

Cheng Luo
Shenzhen University
luocheng2020@email.szu.edu.cn

Qinliang Lin
Shenzhen University
2017192020@email.szu.edu.cn

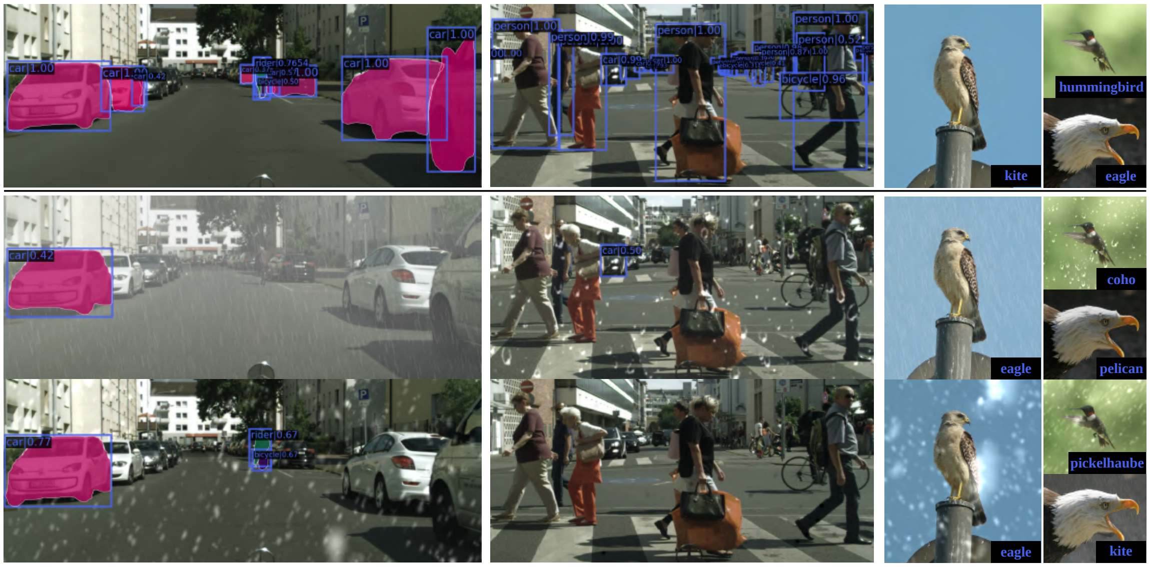
Weicheng Xie*
Shenzhen University
wcxie@szu.edu.cn

Minmin Liu
Shenzhen University
liuminmin2020@email.szu.edu.cn

Linlin Shen
Shenzhen University
llshen@szu.edu.cn

Keerthy Kusumam
University of Nottingham
keerthy.kusumam2@nottingham.ac.uk

Siyang Song†
University of Cambridge
ss2796@cam.ac.uk



(a) Instance Segmentation (b) Object Detection (c) Classification

Figure 1. Our approach can **attack various vision tasks**, *i.e,* (a) instance segmentation, (b) object detection and (c) classification with **images of arbitrary scales**. The first row shows benign examples and the last two rows display images attacked by our approach.

Abstract

Traditional L_p norm-restricted image attack algorithms suffer from poor transferability to black box scenarios and poor robustness to defense algorithms. Recent CNN generator-based attack approaches can synthesize unrestricted and semantically meaningful entities to the image, which is shown to be transferable and robust. However, such methods attack images by either synthesizing local adversarial entities, which are only suitable for attacking specific contents or performing global attacks, which are only applicable to a specific image scale. In this paper, we propose a novel Patch Quilting Generative Adversarial Networks (PQ-GAN) to learn the first scale-free CNN generator that can be applied to attack images with arbitrary scales for various computer vision tasks. The principal investigation on transferability of the generated adversarial examples, robustness to defense frameworks, and vi-

Work	Photo-realistic	Scale-free	Task-agnostic	Black-box	Defense
UAP $_{\text{CVPR'2018}}$ [52]		✓	✓		✓
AdvFaces $_{\text{IJCB'2020}}$ [14]	✓			✓	
ColorFool $_{\text{CVPR'2020}}$ [61]	✓	✓		✓	✓
RA-AVA $_{\text{IJCAI'2021}}$ [68]	✓	✓			✓
Shadows Attack $_{\text{CVPR'2022}}$ [78]	✓	✓		✓	✓
Ours	✓	✓	✓	✓	✓

Table 1. The comparison of our attack method to the previous works. Our attack method is scale-free, task-agnostic, and imperceptible. Meanwhile, we experimentally show that our method has high black-box transferability and is robust to existing defense algorithms.

sual quality assessment show that the proposed PQG-based attack framework outperforms the other nine state-of-the-art adversarial attack approaches when attacking the neural networks trained on two standard evaluation datasets (i.e., ImageNet and CityScapes). Our anonymous code is made available at https://anonymous.4open.science/r/PQAttack-0781.

1. Introduction

Deep Neural Networks (DNNs) are vulnerable to adversarial examples, *e.g.*, images with carefully designed adversarial perturbations can easily mislead well-trained DNNs to output incorrect predictions. To overcome such malicious attacks, several adversarial defense algorithms have been proposed, which, in turns, simulate the development of robust adversarial attack algorithms to disrupt these defenses. Therefore, investigating robust and powerful image attack algorithms plays a crucial role in progressing current research toward developing strong defense algorithms.

Traditional image attack approaches [7, 17, 20, 24, 47, 51, 65, 72] focus on generating perturbations at the pixel-level with L_p-norm restrictions, which possess strong capabilities to mislead predictors but are imperceptible to human eyes. However, the attack strength of such restricted approaches cannot be transferred to unseen networks and the noises are easily defensed [12, 77]. Subsequently, many studies devote their efforts to formulating methods that deliver more robust attack patterns against the defense algorithms. Some of them add noisy perturbation with weaker restriction [52, 66] or even without restriction [10]. Despite that larger perturbation boosts up the transferability and robustness of attack algorithms, the perturbation is perceptible to human eyes and the adversarial examples not photo-realistic.

To solve this problem, some studies employ generative adversarial networks (GAN) [23] to generate semantically meaningful local entities and synthesize them to the image [30, 38, 39, 62, 75, 78] or to change the texture of a particular area of the image [18, 32]. Such GAN-style methods can generate robust and transferable adversarial examples with high image quality. However, such methods are designed for a specific task, such as face recognition [38] and vehicle motion prediction [39]. (**Problem 1**). Alternatively, instead of generating a local entity, some works propose methods that generate adversarial examples in an end-to-end manner where a global adversarial perturbation is carefully hidden in the target image [4, 14, 33, 43, 53, 55, 73, 76]. However, due to the limitation of traditional GAN structure, these methods can only generate adversarial examples of one particular or highly limited scale (**Problem 2**). The whole generative network must be re-trained when changing the target image scale. It is worth mentioning that some works can generate adversarial examples with global semantic patterns without using GAN [5, 21, 54, 61, 68]. However, these methods can only generate a specific attack pattern with carefully designed math formulation, which cannot be extended to general usage.

In this paper, we propose a novel Patch Quilting Generative Adversarial Network (PQ-GAN) to address the aforementioned problems of existing image attack algorithms. The PQ-GAN learns three cascaded generators that can synthesize photo-realistic, scale-free patterns to attack target images of any scale on the whole-image level (globally) (**addressing the problem 2**). Importantly, the synthesized realistic and semantically meaningful pattern ensures the visual quality of the adversarial examples. Task agnostic property allows our approach to be applied to generating various photo-realistic patterns, e.g., rain streaks, snow flakes, and camera lens dirt. It can be applied to many computer vision tasks such as image classification, object detection, instance segmentation, etc. (**addressing the problem 1**). In addition, our approach generates patterns with unrestricted pixel value, ensuring transferability and robustness. The main advantages of our approach compared to existing approaches are listed in Table 1. The main contributions and novelties of the proposed approach are summarized as follows:

- We propose a PQ-GAN-based unrestricted adversarial attack pipeline that generates various global adversarial patterns to attack images. This method is not limited to attack images of any particular scale or computer vision task, namely, scale-free and task-agnostic.
- We propose a novel PQ-GAN which can learn three cascaded generators that synthesize photo-realistic and semantically meaningful images of any scale without any distortion or discontinuity. To the best of our

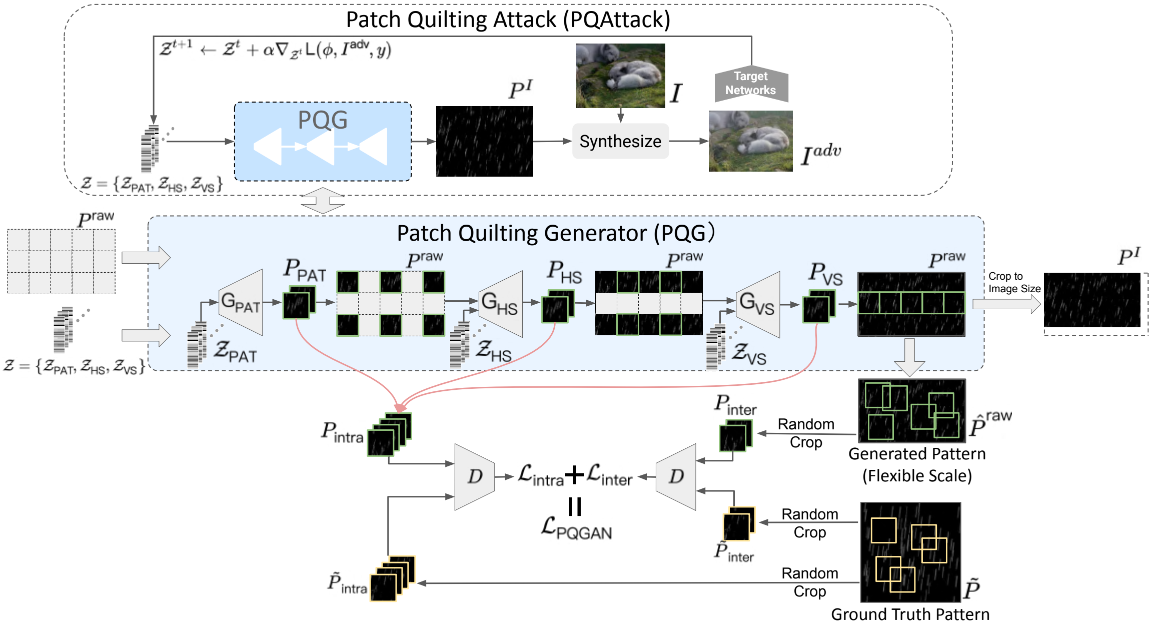

Patch Quilting Attack (PQAttack)
$\mathcal{Z}^{t+1} \leftarrow \mathcal{Z}^{t} + \alpha \nabla_{\mathcal{Z}^{t}} \mathrm{L}(\phi, I^{\mathrm{adv}}, y)$
Target Networks
PQG
P^{I}
I
Synthesize
I^{adv}
$\mathcal{Z} = \{z_{\mathrm{PAT}}, z_{\mathrm{HS}}, z_{\mathrm{VS}}\}$
P^{raw}
Patch Quilting Generator (PQG)
G_{PAT}
P_{PAT}
z_{PAT}
P^{raw}
G_{HS}
P_{HS}
z_{HS}
P^{raw}
G_{VS}
P_{VS}
z_{VS}
P^{raw}
Crop to Image Size
P^{I}
$\mathcal{Z} = \{z_{\mathrm{PAT}}, z_{\mathrm{HS}}, z_{\mathrm{VS}}\}$
P_{intra}
P_{inter}
Random Crop
$\hat{P}^{\mathrm{raw}}$
Generated Pattern (Flexible Scale)
D
$\mathcal{L}_{\mathrm{intra}} + \mathcal{L}_{\mathrm{inter}}$
$=$
$\mathcal{L}_{\mathrm{PQGAN}}$
D
Random Crop
$\tilde{P}_{\mathrm{inter}}$
Random Crop
$\tilde{P}$
$\tilde{P}_{\mathrm{intra}}$
Ground Truth Pattern

Figure 2. Illustration of the Patch Quilting Attack pipeline (Top) and the training strategy of the Patch Quilting Generator (Bottom).

knowledge, this is the first deep learning generator that can synthesize images of any scale.

- The principal investigation results demonstrate that our approach delivers state-of-the-art attack strength and transferability among the existing synthesis-based attack methods, and our experiments verify the dominance of the proposed approach against various types of defense algorithms.

2. Related Work

Task/contents-specific Adversarial Attack: Traditional adversarial attack strategies [7,16,17,24,65] are extensively researched to generate adversarial examples by adding L_p bounded adversarial perturbations to the target images. Such strategies usually lack robustness to the defense algorithms [13,41,48,59,63,77] as the highly restricted perturbations are easily removed. As a result, some recent studies attempted to replace the L_p constraint with either perceptual similarity [71] or JND$_p$ [18] constraint. Recently, Luo *et al*. [46] further propose to add perturbations to attack images based on semantic similarity. To further improve the robustness, others [40,44,74] propose to generate a semantic meaningful local patch to attack images. For example, some studies attack facial images by adding a glass [62], a hat [38], or makeups [30,75] to the target face. Eykholt *et al*. [19] propose to add graffiti to attack road signs. Duan *et al*. [18] and Zhong *et al*. [78] perform style transfer or add shadow to attack a selected region in the image. Besides, some studies are built on specific math formulas to generate specific robust and global attack patterns, including haze [21], vignetting [68], and moire pattern [54]. While the aforementioned strategies can generate robust attacks, most of them still suffer from three main problems: (i) some of them can be only applied to limited application scenarios such as the human face [30, 38, 62, 75] or road sign [19]; (ii) most of them are only suitable for attacking networks of a specific computer vision task (e.g., image classification networks [6, 74] or object detection networks [40, 44]); and (iii) they are still not robust enough to recently proposed advanced defense algorithms [13,41,48,63,77].

Scale-specific Generative Model-based Adversarial Attack: To obtain human-realistic and more robust attack patterns with high diversity for various scenarios, recent approaches frequently employ Generative Adversarial Networks (GAN)-style generators to synthesize semantic patterns [4, 38, 43, 75]. This is because GAN [23] has been widely used in many areas due to its capability of learning and generating various robust data distributions. However, these approaches can only generate attack patterns of a fixed scale decided by the pre-defined generator because the traditional GAN structure is designed for fix-scale image generation. Although some researchers propose hierarchical [8, 34, 60] or growing convolution [35] architectures to generate images of different resolutions by picking the feature map from different layers, they can only generate patterns of a small set of pre-defined resolutions, which cannot be fully reusable under scale-agnostic adversarial attack scenario. In this paper, we propose an image Patch Quilting Generative Adversarial Network (PQ-GAN) that can generate patterns of any scale without retraining the model.

3. Methodology

3.1. Patch Quiliting Attack

Given a target network of agnostic image analysis task with model parameters ϕ and a loss function $\mathcal{L}(\phi, I, y)$ used for model training, where y is the label of the benign image I, the goal to find an adversarial example I^{adv} that maximizes $\mathcal{L}(\phi, I^{\text{adv}}, y)$ under the restriction that I^{adv} is perceptually natural. In particular, Fig. 2 illustrates our scale-agnostic generative model (whose model parameters are represented as ψ), namely Patch Quilting Generator (PQG). The PQG takes a set of latent embeddings $\mathcal{Z}$ initialized with Gaussian distribution of mean 0 and standard deviation 1 as the input, which controls the characteristic of the pattern, and outputs a photo-realistic pattern P^I. Then, P^I is synthesized to the target image I to produce an adversarial example I^{adv} that is perceptually natural. Now the problem is transformed to find a set of latent embeddings $\mathcal{Z}$ that maximize $\mathcal{L}(\phi, I^{\text{adv}}, y)$, which is formulated as:

$$\begin{aligned} & \max_{\mathcal{Z}} && \mathcal{L}(\phi, I^{\text{adv}}, y) \\ & \textbf{Subject to} && I^{\text{adv}} = \text{Syn}(I, P^I) \\ & \text{where} && P^I = \text{PQG}(\mathcal{Z}, \psi) \end{aligned} \tag{1}$$

Notice that PQG is a pre-trained model that does not need to be retrained in this attack stage. We explain the latent embeddings $\mathcal{Z}$ and how the PQG is designed to be scale-agnostic in detail in Sec. 3.2. $\text{Syn}(\cdot)$ is a customized synthesis function depends on the target pattern, (*i.e.* pixel-wised addition, or depth-aware synthesis [31]).

To achieve the adversarial objective, we simply apply gradient ascent to the loss function being use for model train to update the latent embeddings $\mathcal{Z}$ through iterations, *i.e.*,

$$\mathcal{Z}^{t+1} \leftarrow \mathcal{Z}^{t} + \alpha \nabla_{\mathcal{Z}^t} \mathcal{L}(\phi, I^{\text{adv}}, y) \tag{2}$$

where t is the time-stamp and α denotes the learning rate. We put $\mathcal{Z}$ of the last iteration into PQG to generate the adversarial example.

3.2. Patch Quilting GAN

3.2.1 Scale-free Image Generation via PQG

The Patch Quilting Generator (PQG) consists of three cascaded generators G_{PAT}, G_{HS}, and G_{VS} with the learnable weights $\psi_{\text{PAT}}, \psi_{\text{HS}}, \psi_{\text{VS}}$, where each can generate image patches of the scale $h \times w$. PQG takes three sets of latent embeddings $\mathcal{Z} = \{\mathcal{Z}_{\text{PAT}}, \mathcal{Z}_{\text{HS}}, \mathcal{Z}_{\text{VS}}\}$ as the input, and outputs a set of patches of the scale $h \times w$, which contain the required attack pattern. These patches can then be combined as a smooth and photo-realistic global attack pattern P^I whose scale can be customized based on the target image.

As shown in Fig. 2, given a target image I with the scale of $H \times W$, our PQG first initializes an attack pattern $P^{\text{raw}} \in \mathbb{R}^{\hat{H} \times \hat{W}}$, which consists of a integral number of empty patches of size $h \times w$, *i.e.* $\hat{H}, \hat{W}$ are formulated as:

$$\hat{H} = N_h \times h, \quad \hat{W} = N_w \times w \tag{3}$$

where $N_h = \text{ceil}(\frac{H}{h})$, $N_w = \text{ceil}(\frac{W}{w})$ denote the minimum number of patches that are required to fill up each row and column, $\text{ceil}(\cdot)$ means rounding up to an integer. Then three generators then generates a set of attack patches as follows:

Firstly, G_{PAT} takes a set of latent embeddings $\mathcal{Z}_{\text{PAT}}$ to generate a set of attack patches P_{PAT} to fill up non-adjacent odd rows and columns in the P^{raw}. Let $N_h^{\text{PAT}} = \text{ceil}(\frac{N_h}{2})$ and $N_w^{\text{PAT}} = \text{ceil}(\frac{N_w}{2})$. This step can be formulated as:

$$\begin{aligned} & && p^{\text{raw}}_{2a-1,2b-1} \in P_{\text{PAT}} = \text{G}_{\text{PAT}}(\mathcal{Z}_{\text{PAT}}, \psi_{\text{PAT}}) \\ & \textbf{Subject to} && \mathcal{Z}_{\text{PAT}} = \{z_{2a-1,2b-1} \in \mathcal{N}(0,1)^k\} \\ & && a \in \{1, 2, \cdots, N_h^{\text{PAT}}\},\ b \in \{1, 2, \cdots, N_w^{\text{PAT}}\} \end{aligned} \tag{4}$$

where $p^{\text{raw}}_{2a-1,2b-1}$ denotes the image patch located at the $2a - 1_{\text{th}}$ row and $2b - 1_{\text{th}}$, while $z_{2a-1,2b-1} \in \mathcal{N}(0,1)^k$ denotes the latent embedding of dimension k being used to generate $p^{\text{raw}}_{2a-1,2b-1}$.

Secondly, G_{HS} generates a set of horizontal context-aware realistic adversarial attack patches P_{HS} where each pathc fills up a horizontal gap in P^{raw} based on not only Z_{HS} but also its horizontal neighbours in P^{raw}, which are generated from G_{PAT}. By letting $N_h^{\text{HS}} = \text{ceil}(\frac{N_h}{2})$ and $N_w^{\text{HS}} = \text{ceil}(\frac{N_w}{2}) - 1$, this process is formulated as:

$$\begin{aligned} & && p^{\text{raw}}_{2a-1,2b} \in P_{\text{HS}} = \text{G}_{\text{HS}}(\mathcal{Z}_{\text{HS}}, p^{\text{raw}}_{2a\pm1,2b+1}, \psi_{\text{HS}}) \\ & \textbf{Subject to} && \mathcal{Z}_{\text{HS}} = \{z_{2a-1,2b} \in \mathcal{N}(0,1)^k\} \\ & && a \in \{1, 2, \cdots, N_h^{\text{HS}}\},\ b \in \{1, 2, \cdots, N_w^{\text{HS}}\} \end{aligned} \tag{5}$$

Notice that $p^{\text{raw}}_{2a-1,2b\pm1} \in P_{\text{PAT}}$.

Finally, G_{VS} generates a set of vertical context-aware realistic adversarial attack patches P_{VS}, targeting on filling up the rest regions (all vertical gaps) in P^{raw}. Specifically, each patch generated by G_{VS} fills up a gap based on not only $\mathcal{Z}_{\text{VS}}$ but also its vertical neighbours in P^{raw}. By letting $N_h^{\text{VS}} = \text{ceil}(\frac{N_h}{2}) - 1$ and $N_w^{\text{VS}} = N_w$, which are produced from G_{PAT} as:

$$\begin{aligned} & && p^{\text{raw}}_{2a,b} \in P_{\text{VS}} = \text{G}_{\text{VS}}(\mathcal{Z}_{\text{VS}}, p^{\text{raw}}_{2a\pm1,b}, \psi_{\text{VS}}) \\ & \textbf{Subject to} && \mathcal{Z}_{\text{VS}} = \{z_{2a,b} \in \mathcal{N}(0,1)^k\} \\ & && a \in \{1, 2, \cdots, N_h^{\text{VS}}\},\ b \in \{1, 2, \cdots, N_w^{\text{VS}}\} \end{aligned} \tag{6}$$

Notice that $p^{\text{raw}}_{2a\pm1,b} \in P_{\text{PAT}} \bigcup P_{\text{HS}}$.

Consequently, a global pattern $\hat{P}^{\text{raw}}$ is obtained by filling all patches of the P^{raw}, where attack patches produced by three generators are concatenated. We then remove the extra pixels of the $\hat{P}^{\text{raw}} \in \mathbb{R}^{\hat{H} \times \hat{W}}$ to make it have the same size $H \times W$ to the target image I, which is denoted as the final P^I. In summary, the proposed PQG can not only synthesize

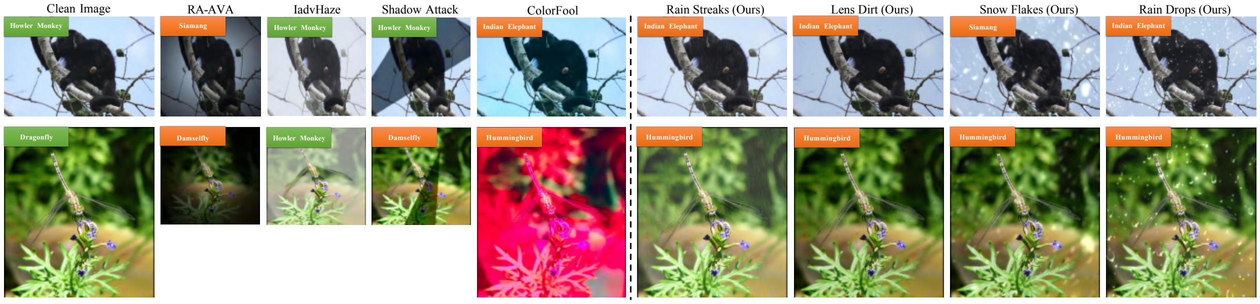


Figure 3. Visualization of adversarial examples generated by various attack methods. Our adversarial examples successfully mislead the target classifier. Note that our attack method can generate high-resolution adversarial examples with the original image scale.

global image attack patterns of any required scale without requiring re-training the network, but also allow the final produced pattern to be smooth, continuous and semantically meaningful.

3.2.2 PQ-GAN Optimization

To learn three generators of the POG, we propose a GAN-style training strategy (called PQ-GAN). To produce a globally smooth and continuous attack pattern, we propose a **Intra-Patch Smoothness Loss** to ensure each generated attack patch to be smooth and photo-realistic, and a **Inter-Patch Smoothness Loss** to enforce the smoothness between neighbor patches.

Intra-Patch Smoothness Loss: As displayed at the bottom of the Fig. 2, we collect all the patches $P_{\text{intra}} = P_{\text{PAT}} \bigcup P_{\text{HS}} \bigcup P_{\text{VS}}$ generated by generators of the POG, treating them as negative samples, while a set of positive samples $\tilde{P}_{\text{intra}} = \{\tilde{p}^m_{\text{intra}} | \, m = 1, 2, \cdots, M_{\text{intra}}\}$ are obtained by randomly cropping a set of ground truth patches of size $h \times w$ from the a ground truth Pattern $\tilde{P}$, where M_{intra} equals the number of patches in P_{intra}. All negative and positive sample are then fed into a discriminator D to calculate the generator loss and discriminator loss, respectively, based on the standard formulation of the Wasserstein GAN with gradient penalty [3]. This process is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{intra}} =& \mathcal{L}_{\text{G}_{\text{PAT}}}(P_{\text{intra}}, \psi_{\text{PAT}}) + \mathcal{L}_{\text{G}_{\text{HS}}}(P_{\text{intra}}, \psi_{\text{HS}}) + \mathcal{L}_{\text{G}_{\text{VS}}}(P_{\text{intra}}, \psi_{\text{VS}}) + \\ & \mathcal{L}_{\text{D}}(P_{\text{intra}}, \psi_{\text{D}}) + \mathcal{L}_{\text{D}}(\tilde{P}_{\text{intra}}, \psi_{\text{D}}) \end{aligned} \tag{7}$$

where $\mathcal{L}_{\text{G}_{\text{PAT}}}(P_{\text{intra}}, \psi_{\text{PAT}})$, $\mathcal{L}_{\text{G}_{\text{HS}}}(P_{\text{intra}}, \psi_{\text{HS}})$, and $\mathcal{L}_{\text{G}_{\text{VS}}}(P_{\text{intra}}, \psi_{\text{VS}})$ denotes the generator loss of patches $P_{\text{PAT}}, P_{\text{HS}}$, and P_{VS}, respectively. $\mathcal{L}_{\text{D}}(P_{\text{intra}}, \psi_{\text{D}})$ denotes the discriminator loss of negative samples P_{intra}, and $\mathcal{L}_{\text{D}}(\tilde{P}_{\text{intra}}, \psi_{\text{D}})$ denotes the discriminator loss of positive samples $\tilde{P}_{\text{intra}}$.

Inter-Patch Smoothness Loss: To ensure the **smoothness and continuity** among neighboring patches, we additionally randomly crop M_{inter} patches $P_{\text{inter}} = \{p^m_{\text{inter}} | \, m = 1, 2, \cdots, M_{\text{inter}}\}$ of size $h \times w$ from the generated attack pattern $\hat{P}^{\text{raw}}$ and treat them as negative samples, where M_{inter} is a hyper-parameter. To balance between the positive and negative samples, we then randomly crop the same number of patches $\tilde{P}_{\text{inter}} = \{\tilde{p}^m_{\text{inter}} | \, m = 1, 2, \cdots, M_{\text{inter}}\}$ from the ground truth pattern $\tilde{P}$. We feed P^{inter} and $\tilde{P}^{\text{inter}}$ to D and compute the loss as:

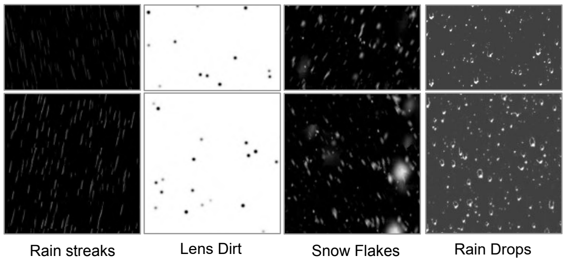


Figure 4. Four different patterns generated by our Patch Quilting Generator. PQG can generate patterns of any scale with great variety which is the key to the adversarial attack strength.

$$\mathcal{L}_{\text{inter}} = \mathcal{L}_{\text{D}}(P_{\text{inter}}, \psi_{\text{D}}) + \mathcal{L}_{\text{D}}(\tilde{P}_{\text{inter}}, \psi_{\text{D}}) \tag{8}$$

where $\mathcal{L}_{\text{D}}(P^{\text{inter}}, \psi_{\text{D}})$ and $\mathcal{L}_{\text{D}}(\tilde{P}^{\text{inter}}, \psi_{\text{D}})$ denotes the discriminator loss obtained from negative samples P^{inter} and positive samples $\tilde{P}^{\text{inter}}$, respectively. For the details of how the generator loss and discriminator loss respect to the negative samples and positive samples being calculated, please refer to Arjovsky *et al.* [3].

Consequently, the final loss for training PQ-GAN is obtained by combining Inter- and intra-Patch Smoothness losses as:

$$\mathcal{L}_{\text{PQGAN}} = \mathcal{L}_{\text{intra}} + \mathcal{L}_{\text{inter}} \tag{9}$$

The combined loss would enforce three generators to be jointly learned for generating a smooth and continuous global attack pattern of any scale.

4. Experiment

In this section, we demonstrate the effectiveness of the proposed method under various settings. The experimental setup is introduced in Sec. 4.1. To evaluate the task-agnostic property of our method, we compare it with existing approaches based on three computer vision tasks, *i.e.*, image

Model	ATK Region	Attack Methods	ResNet-18	VGG-19	ResNet-50	Inception-V3	MobileNet-V3
		None	67.3	68.3	74.0	71.3	66.1
ResNet-18	Pixel-wise	FGSM $_{\text{ICLR'2014}}$ [24]	3.8*(63.5$\downarrow$)	54.3 (14.0$\downarrow$)	56.4 (17.6$\downarrow$)	59.2 (12.1$\downarrow$)	53.0 (13.1$\downarrow$)
		C&W $_{\text{IEEE SP'2017}}$ [7]	**0.0*****(67.3**$\downarrow$**)**	55.8 (12.5$\downarrow$)	57.6 (16.4$\downarrow$)	60.1 (11.2$\downarrow$)	53.1 (13.0$\downarrow$)
	Local	Shadow ATK $_{\text{CVPR'2022}}$ [78]	16.7*(50.6$\downarrow$)	56.4 (11.9$\downarrow$)	56.9 (17.1$\downarrow$)	57.3 (14.0$\downarrow$)	51.4 (14.7$\downarrow$)
		ColorFool $_{\text{CVPR'2020}}$ [61]	9.3*(58.0$\downarrow$)	55.2 (13.1$\downarrow$)	56.4 (17.6$\downarrow$)	60.1 (11.2$\downarrow$)	53.5 (12.6$\downarrow$)
	Global	IadvHaze $_{\text{Arxiv'2021}}$ [21]	21.4*(55.9$\downarrow$)	54.5 (13.8$\downarrow$)	56.0 (18.0$\downarrow$)	58.4 (12.9$\downarrow$)	48.3 (18.8$\downarrow$)
		RA-AVA $_{\text{IJCAI'2021}}$ [68]	4.2*(63.1$\downarrow$)	55.5 (12.8$\downarrow$)	54.7 (19.3$\downarrow$)	55.6 (15.7$\downarrow$)	48.0 (18.1$\downarrow$)
		Rain Streaks (Ours)	3.6*(63.7$\downarrow$)	52.2 (16.1$\downarrow$)	54.6 (19.4$\downarrow$)	54.3 (17.0$\downarrow$)	48.0 (18.1$\downarrow$)
		Lens Dirt (Ours)	4.3*(63.0$\downarrow$)	50.0 (18.3$\downarrow$)	54.5 (19.5$\downarrow$)	55.7 (15.6$\downarrow$)	48.3 (17.8$\downarrow$)
		Snow Flakes (Ours)	2.5*(64.8$\downarrow$)	51.0 (17.3$\downarrow$)	**53.6 (20.4**$\downarrow$**)**	56.1 (15.2$\downarrow$)	48.8 (17.3$\downarrow$)
		Rain Drops (Ours)	2.1*(65.2$\downarrow$)	**49.1 (19.2**$\downarrow$**)**	54.0 (20.0$\downarrow$)	**53.8 (17.5**$\downarrow$**)**	**46.5 (19.6**$\downarrow$**)**
VGG-19	Pixel-wise	FGSM $_{\text{ICLR'2014}}$ [24]	54.8 (12.5$\downarrow$)	5.2* (63.1$\downarrow$)	57.1 (16.9$\downarrow$)	58.3 (13.0$\downarrow$)	53.0 (13.1$\downarrow$)
		C&W $_{\text{IEEE SP'2017}}$ [7]	55.6 (11.7$\downarrow$)	**0.0*** **(68.3**$\downarrow$**)**	58.2 (15.8$\downarrow$)	58.4 (12.9$\downarrow$)	52.6 (13.5$\downarrow$)
	Local	Shadow ATK $_{\text{CVPR'2022}}$ [78]	54.6 (12.7$\downarrow$)	14.8*(53.5$\downarrow$)	55.2 (18.8$\downarrow$)	57.4 (13.9$\downarrow$)	50.6 (15.5$\downarrow$)
		ColorFool $_{\text{CVPR'2020}}$ [61]	53.4 (13.9$\downarrow$)	10.1* (58.2$\downarrow$)	55.2 (18.8$\downarrow$)	58.1 (13.2$\downarrow$)	53.0 (13.1$\downarrow$)
	Global	IadvHaze $_{\text{Arxiv'2021}}$ [21]	55.1 (12.2$\downarrow$)	25.1* (43.2$\downarrow$)	55.8 (18.2$\downarrow$)	60.0 (11.3$\downarrow$)	48.2 (17.9$\downarrow$)
		RA-AVA $_{\text{IJCAI'2021}}$ [68]	51.3 (16.0$\downarrow$)	5.1* (63.2$\downarrow$)	54.6 (19.4$\downarrow$)	55.6 (15.7$\downarrow$)	48.1 (18.0$\downarrow$)
		Rain Streaks (Ours)	50.5 (16.8$\downarrow$)	3.4* (64.9$\downarrow$)	52.8 (21.2$\downarrow$)	55.1 (16.2$\downarrow$)	**47.4 (18.7**$\downarrow$**)**
		Lens Dirt (Ours)	49.5 (17.8$\downarrow$)	4.3* (64.0$\downarrow$)	53.1 (20.9$\downarrow$)	55.6 (15.7$\downarrow$)	48.3 (17.8$\downarrow$)
		Snow Flakes (Ours)	**49.3 (18.0**$\downarrow$**)**	2.1*(66.2$\downarrow$)	52.5 (21.5$\downarrow$)	54.0 (17.3$\downarrow$)	47.4 (18.7$\downarrow$)
		Rain Drops (Ours)	49.8 (17.5$\downarrow$)	2.7* (65.6$\downarrow$)	**52.1 (21.9**$\downarrow$**)**	**53.0 (18.3**$\downarrow$**)**	47.8 (18.3$\downarrow$)

Table 2. Accuracy (%) of five trained models on clean images and adversarial examples. * indicates the result of the white box attack; ATK Region stands for attack region. Our attack methods beat all unrestricted attack methods under the white-box scenario. Although the traditional L_p restricted attack methods (C&W and FGSM) have great performance in the white-box scenario, the attack strength cannot be transferred to the unseen models. Under the black-box scenario, our methods beat all other methods with great improvement, indicating that our attack method is more transferable across unseen models.

classification, object detection, and instance segmentation. Specifically, we evaluate the white-box attack strength and black-box transferability in Sec. 4.2, as well as the robustness against the defense algorithms in Sec. 4.3. We also compare the quality of the adversarial examples produced by our method and others. In addition, we adopt three no-reference image quality assessment metrics to quantify the image quality in Sec. 4.4.

4.1. Experimental Setup

Dataset: The image classification experiments are conducted on 5000 randomly selected images from the validation set of the ImageNet [15] while the object detection and instance segmentation experiments are conducted on the entire validation set of the CityScapes dataset [11].

Implementation details: We employed 2000 (256×256) training samples for each of the four patterns: rain streak, rain drop, snow flakes, and camera lens dirt. Chen *et al.* [9] provides a set of snow flakes patterns where we randomly chose 2000 images for PQG training. Camera lens dirt patterns were generated by randomly adding 30 to 60 white points on a dark image and applying Gaussian blur with kernel standard deviation $\sigma = 5$. Rain streak patterns are generated according to Garg *et al.* [22]; rain drop pattern is an internet image and we perform Aguerrebere *et al.* [2] to generate 2000 patterns of similar distribution. The synthesis strategy of rain drops, snow flakes, and lens dirt patterns are pixel-wise addition formulated as $I^{adv} = I + \gamma * P^{I}$. For rain streaks patterns, we performed depth-aware synthesis according to Hu *et al.* [31]. Subsequently, we individually train four PQGs, each of which can generate various attack images containing the required pattern (See Fig. 4). During the PQG-based Attack, each attack loop consists of 300 iterations with Eq. (2) by default. Detailed parameter settings are provided in the supplementary material.

ATK Methods	mAP % $\downarrow$	
	Fr	Mk
clean	40.3	36.4
DPatch [44]	8.8*	15.3
AdvPatch [40]	5.5*	9.6
UAP [40]	12.1*	12.2
RS (Ours)	**4.2***	**5.1**
LD (Ours)	5.3*	7.7
SF (Ours)	4.9*	4.8
RD (Ours)	5.8*	5.5

(a) Object Detection (Target Network: Faster RCNN)

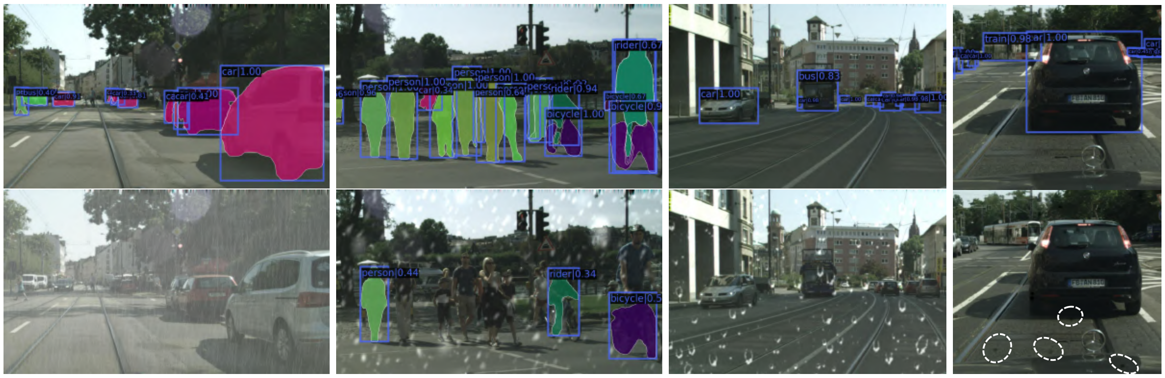
ATK Methods	mAP % $\downarrow$	
	Fr	Mk
clean	40.3	36.4
DPatch [44]	/	/
AdvPatch [40]	/	/
UAP [52]	13.5	6.3*
RS (Ours)	9.1	**2.1***
LD (Ours)	10.1	3.0*
SF (Ours)	**8.2**	2.4*
RD (Ours)	9.4	2.5*

(b) Instance Segmentation (Target Network: Mask RCNN)

Table 3. Compare the cross-task transferability with other attack methods. RS, LD, SF, and RD are the abbreviations of our attack methods using Rain Streak, Lens Dirt, Snow Flakes, and Rain Drops patterns, respectively. Fr and Mk stand for Faster-RCNN and Mask-RCNN, respectively. Note that DPatch and AdvPatch are designed for attacking object detection models only.

4.2. Transferability

Cross-model Transferability: In our experiment, five classifiers including ResNet-18 [27], ResNet-50 [27], VGG-19 [64], Inception-V3 [67], and MobileNet-V3 [29],



(a) Rain Streaks (b) Snow Flakes (c) Rain Drops (d) Lens Dirt

Figure 5. Four kinds of attack patterns generated by our approach. These patterns are scale-free, realistic, and misleading to instance segmentation models (the two columns on the left) and detectors (the two columns on the right).

Defense Attack	None	JPEG	HFC	HGD	APE	DEF-GAN
FGSM [24]	3.8	61.5	64.7	64.7	61.9	60.9
C&W [7]	**0.0**	64.0	65.5	65.5	64.8	61.0
IadvHaze [21]	21.4	45.5	46.3	35.7	41.4	35.7
RA-AVA [68]	4.2	40.2	41.6	43.9	48.9	42.2
ColorFool [61]	9.3	10.4	11.9	12.4	25.1	40.1
Shadow [78]	16.7	18.1	18.5	21.0	22.7	33.4
RS (Ours)	3.6	12.1	11.5	13.2	15.9	26.5
LD (Ours)	4.3	10.3	12.8	12.2	21.9	28.0
SF (Ours)	2.5	9.9	13.3	12.5	**14.5**	23.4
RD (Ours)	2.3	**9.7**	**10.7**	**12.1**	13.6	**25.8**

(a) Image Classification (ResNet-18).

Defense Attack	None	JPEG	HFC	HGD	APE	DEF-GAN
DPatch [44]	8.8	9.5	10.6	14.1	13.9	21.5
AdvPatch [40]	5.5	7.1	8.5	10.5	12.4	15.7
UAP [40]	12.1	14.7	13.5	15.2	13.0	13.2
RS (Ours)	**4.2**	5.0	5.1	**5.6**	7.2	**8.8**
LD (Ours)	5.3	**5.9**	**5.6**	6.7	8.3	9.1
SF (Ours)	5.8	7.6	8.1	7.6	**7.9**	9.9
RD (Ours)	4.9	6.4	7.1	8.2	8.9	8.8

(b) Object Detection (Faster-RCNN)

Defense Attack	None	JPEG	HFC	HGD	APE	DEF-GAN
UAP [52]	6.3	7.2	9.4	9.0	8.6	7.3
RS (Ours)	**2.1**	**2.9**	4.4	4.7	5.4	6.0
LD (Ours)	3.0	2.9	**4.3**	**4.2**	5.6	5.8
SF (Ours)	2.4	2.9	4.3	4.6	5.2	6.0
RD (Ours)	2.5	4.4	4.7	4.6	**4.7**	**5.2**

(c) Instance Segmentation (Mask-RCNN)

Table 4. Classification accuracy % (a) and mean average precision % (b,c) of the model on the adversarial examples generated by different attack methods (first columns) and the adversarial examples that different defense algorithms (first row) are applied.

are employed. We also additionally compare our method with two traditional attack methods: FGSM [24] and C&W [7], two unrestricted local attack methods: ColorFool [61] and Shadow Attack [78], as well as two unrestricted global attack methods: Adversarial Vignetting Attack (AVA) [68] and Adversarial Haze [21]. We employ the classification accuracy as the evaluation metric, where lower classification accuracy indicates better attack performance.

As shown in Tab. 2, our methods achieved the best performance among all compared attack models. Compared to existing approaches, our method provides at least 17.80% and 18.88% average performance drops for ResNet-18 and VGG-19 based classifiers. For example, when our approach attacks ResNet-18 with the Rain Drops attack pattern and transfer the adversarial examples to VGG-19, the classification accuracy is largely decreased from 68.3% to 49.1%, while the second best only decreasing the corresponding system to 54.3%. Fig. 3 also illustrates the adversarial examples produced from different methods on. It can be observed that our method can generate more photo-realistic attack patterns, but the other methods will destroy the structure (Shadow Atatck) or color (ColorFool) of the benign image.

Cross-task Transferability: To further demonstrate the effectiveness of our approach, we then evaluate the transferability of the generated adversarial examples under the cross-task and cross-model scenarios, *i.e.,* we generate the adversarial examples by the objection detection model (Faster RCNN [58]) and transfer to the instance segmentation model (Mask RCNN [26]), and vice versa. we compare our results with a task-agnostic attack methods: UAP [52] and two task-specific attack methods: DPatch [44] and AdvPatch [40]. We employ the mean average precision (mAP %) as the evaluation metric, where lower mAP indicates better attack performance.

As shown in Tab. 3 (a), we employ the Faster RCNN as the surrogate model to generate the adversarial examples and test in the Mask RCNN. The results indicate that our attack decreases the original mAP by a large margin, *i.e.*, 31.3% with the Rain Streaks attack pattern, 30.9% with the Rain Drops attack pattern, and so on. Compared with the current state-of-the-art method AdvPatch, we also achieve competitive results: the result of AdvPatch is 9.6%, and our best result is 5.1%, which is significantly lower.

In Tab. 3(b), we generate the adversarial patterns through Mask RCNN against the Fast RCNN. Our PQAttack still achieves competitive results. Using the Snow Flakes pattern, an improvement of 5.3% is achieved by PQAttack (8.2%) compared with UAP (13.5%).

4.3. Robustness

To evaluate the robustness, we apply three smoothing-based defense algorithms, JPEG Compression [13], High Frequency Suppression [77], HGD [41], and two GAN-based defense algorithms Ape-GAN [63], Defense-GAN [59] to the adversarial examples generated by our methods. Tab. 4 reports the experimental results on: (a) attacking image classification, (b) object detection, and (c) instance segmentation models. Smoothing-based defense algorithms (JPEG, HFC, HGD) are very effective in defending L_p-restricted attack methods (C&W, FGSM, and UAP) but less effective in defending unrestricted attack methods. To illustrate this, we also visualize the adversarial trajectories in Fig. 6.

Defense GAN [59] defense by learning the distribution of the benign images, which can identify and remove the disconformities between the distribution of the adversarial examples and benign images [59], therefore very effective in defending the local attack methods (ColorFool [61], Shadow Attack [78], DPatch [44], AdvPatch [40]). Different from the local attack methods, our method synthesizes semantic-aware adversarial patterns globally, which is very difficult to remove.

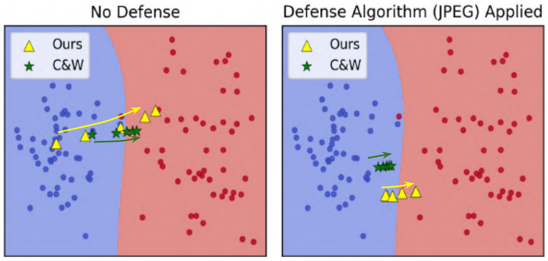


Figure 6. Adversarial trajectories visualized by the PCA [1] dimension reduction algorithm. The decision boundaries are plotted by applying SVM [28] to the reduced vectors. The adversarial example gradually departs from its original class (blue) and moves to the target class (red) through the adversarial attack iterations. The JPEG compression defense algorithm [13] successfully defense the restricted method C&W (green stars) by pulling the adversarial examples back from the decision boundary, but it fails to defend our method because our method has no pixel-value restriction, and our adversarial example (yellow triangles) move much further apart from the decision boundary.

4.4. Image Quality

For visualization, some typical adversarial examples of the Cityscapes and ImageNet are illustrated in Fig. 1, 3, and 5. We observe that our method can generate patterns that are noticeable but natural to human eyes, *i.e.*, it is hard for humans to identify the adversarial examples without referring to the original images. Meanwhile, the adversarial example can mislead the networks to give an incorrect output. It means that our generated semantic attack not only deceives the human visual system but also cheats the machine vision system.

ATK	BRISQUE ↓	NIQE ↓	PIQE ↓
Clean	22.4207	3.6792	32.1288
C&W [7]	32.1891	8.5136	36.1667
FGSM [24]	43.9118	17.8617	59.1998
IadvHaze [21]	41.6842	12.7820	72.2287
RA-AVA [68]	33.2389	9.3145	32.8531
ColorFool [61]	29.4896	5.1896	31.3078
Shadow [78]	30.7310	5.1603	30.9756
RS (Ours)	30.3656	5.6763	**30.5022**
LD (Ours)	**27.3893**	6.2135	31.8128
SF (Ours)	32.6419	**4.8950**	33.2251
RD (Ours)	31.2009	6.5481	32.9371

Table 5. Results of three non-reference image quality assessment metrics BRISQUE [49], NIQE [50] and PIQE [69] being evaluated on the adversarial examples generated by seven attack approaches. All adversarial examples are generated from the pre-selected 5000 images from the ImageNet dataset [15].

We further employ three reference-free image quality assessment metrics: BRISQUE [49], NIQE [50] and PIQE [69] to quantify the image quality. We compare the average scores of the adversarial examples generated by different attack methods. The results shown in Tab. 5 demonstrate that the adversarial examples produced by our approach have the best image quality among all adversarial examples.

5. Conclusion

This paper proposes a novel PQ-GAN for adversarial attack, which learns a set of generators called PQG. This is the first generator-based approach that can generate **photo-realistic patterns of any scale**, which can be used to **attack various computer vision tasks**. The results show that our PQAttack approach achieved state-of-the-art results in misleading various white-box and black-box computer vision models. Importantly, the adversarial examples produced by our approach are not only robust to various defense algorithms but also have high visual qualities.

Limitations and future works: The main limitation of our method is that our PQG can only generate limited types of attack patterns (e.g., rain and snow). However, it cannot generate patterns such as landscapes. we aim to address this in the future to generate more diversified scale-free attack patterns.

Acknowledgements The work was supported by Natural Science Foundation of China under grants no. 62276170, 91959108, the Science and Technology Project of Guangdong Province under grant no. 2020A1515010707, the Science and Technology Innovation Commission of Shenzhen under grant no. JCYJ20190808165203670. The work of Siyang Song is funded by the EPSRC/UKRI under grant ref. EP/R030782/1.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. 8

[2] Cecilia Aguerrebere, Yann Gousseau, and Guillaume Tartavel. Exemplar-based texture synthesis: the efros-leung algorithm. *Image Processing On Line*, 2013:223–241, 2013. 6

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 5

[4] Tao Bai, Jun Zhao, Jinlin Zhu, Shoudong Han, Jiefeng Chen, Bo Li, and Alex Kot. Ai-gan: Attack-inspired generation of adversarial examples. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2543–2547. IEEE, 2021. 2, 3

[5] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A Forsyth. Unrestricted adversarial examples via semantic manipulation. *arXiv preprint arXiv:1904.06347*, 2019. 2

[6] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 3

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 2, 3, 6, 7, 8

[8] Weimin Chen, Yuqing Ma, Xianglong Liu, and Yi Yuan. Hierarchical generative adversarial networks for single image super-resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 355–364, 2021. 3

[9] Wei-Ting Chen, Hao-Yu Fang, Cheng-Lin Hsieh, Cheng-Che Tsai, I Chen, Jian-Jiun Ding, Sy-Yen Kuo, et al. All snow removed: Single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4196–4205, 2021. 6

[10] Yuefeng Chen, Xiaofeng Mao, Yuan He, Hui Xue, Chao Li, Yinpeng Dong, Qi-An Fu, Xiao Yang, Wenzhao Xiang, Tianyu Pang, et al. Unrestricted adversarial attacks on imagenet competition. *arXiv preprint arXiv:2110.09903*, 2021. 2

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6, 13

[12] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017. 2

[13] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–204, 2018. 3, 8

[14] Debayan Deb, Jianbang Zhang, and Anil K Jain. Advfaces: Adversarial face synthesis. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2020. 2

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 6, 8

[16] Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. Greedyfool: Distortion-aware sparse adversarial attack. *arXiv preprint arXiv:2010.13773*, 2020. 3

[17] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 2, 3

[18] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1000–1008, 2020. 2, 3

[19] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018. 3

[20] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017. 2

[21] Ruijun Gao, Qing Guo, Felix Juefei-Xu, Hongkai Yu, and Wei Feng. Advhaze: Adversarial haze attack. *arXiv preprint arXiv:2104.13673*, 2021. 2, 3, 6, 7, 8

[22] Kshitiz Garg and Shree K Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007. 6

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2, 3

[24] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2, 3, 6, 7, 8

[25] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017. 13

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 7, 12

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[28] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998. 8

[29] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 6

[30] Shengshan Hu, Xiaogeng Liu, Yechao Zhang, Minghui Li, Leo Yu Zhang, Hai Jin, and Libing Wu. Protecting facial privacy: Generating adversarial identity masks via style-robust makeup transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15014–15023, 2022. 2, 3

[31] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8022–8031, 2019. 4, 6, 13

[32] Zhanhao Hu, Siyuan Huang, Xiaopei Zhu, Fuchun Sun, Bo Zhang, and Xiaolin Hu. Adversarial texture for fooling person detectors in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13307–13316, 2022. 2

[33] Surgan Jandial, Puneet Mangla, Sakshi Varshney, and Vineeth Balasubramanian. Advgan++: Harnessing latent layers for adversary generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[34] Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7799–7808, 2020. 3

[35] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 3

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 13

[37] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020. 12, 13

[38] Stepan Komkov and Aleksandr Petiushko. Advhat: Real-world adversarial attack on arcface face id system. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 819–826. IEEE, 2021. 2, 3

[39] Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14254–14263, 2020. 2

[40] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019. 3, 6, 7, 8, 12

[41] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1778–1787, 2018. 3, 8

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 13

[43] Xuanqing Liu and Cho-Jui Hsieh. Rob-gan: Generator, discriminator, and adversarial attacker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11234–11243, 2019. 2, 3

[44] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018. 3, 6, 7, 8, 12

[45] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 13

[46] Cheng Luo, Qinliang Lin, Weicheng Xie, Bizhu Wu, Jinheng Xie, and Linlin Shen. Frequency-driven imperceptible adversarial attack on semantic similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15315–15324, 2022. 3

[47] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2

[48] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017. 3

[49] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012. 8

[50] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012. 8

[51] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 2

[52] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2452–2465, 2018. 2, 6, 7, 12

[53] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. On generating transferable targeted perturbations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2021. 2

[54] Dantong Niu, Ruohao Guo, and Yisen Wang. Morié attack (ma): A new potential risk of screen photos. *Advances in Neural Information Processing Systems*, 34:26117–26129, 2021. 2, 3

[55] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. 2

[56] Harish Guruprasad Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 983–991, 2020. 12

[57] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 12, 13

[58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 7, 12

[59] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 3, 8

[60] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4570–4580, 2019. 3

[61] Ali Shahin Shamsabadi, Ricardo Sanchez-Matilla, and Andrea Cavallaro. Colorfool: Semantic adversarial colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1151–1160, 2020. 2, 6, 7, 8

[62] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016. 2, 3

[63] Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. *arXiv preprint arXiv:1707.05474*, 2017. 3, 8

[64] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[65] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. 2, 3, 12

[66] Chenghao Sun, Yonggang Zhang, Wan Chaoqun, Qizhou Wang, Ya Li, Tongliang Liu, Bo Han, and Xinmei Tian. Towards lightweight black-box attacks against deep neural networks. *arXiv preprint arXiv:2209.14826*, 2022. 2

[67] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 6

[68] Binyu Tian, Felix Juefei-Xu, Qing Guo, Xiaofei Xie, Xiaohong Li, and Yang Liu. Ava: Adversarial vignetting attack against visual recognition. *arXiv preprint arXiv:2105.05558*, 2021. 2, 3, 6, 7, 8

[69] N Venkatanath, D Praneeth, Maruthi Chandrasekhar Bh, Sumohana S Channappayya, and Swarup S Medasani. Blind image quality evaluation using perception based features. In *2015 Twenty First National Conference on Communications (NCC)*, pages 1–6. IEEE, 2015. 8

[70] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *European Conference on Computer Vision*, pages 649–665. Springer, 2020. 12, 13

[71] Zhibo Wang, Mengkai Song, Siyan Zheng, Zhifei Zhang, Yang Song, and Qian Wang. Invisible adversarial attack against deep neural networks: An adaptive penalization approach. *IEEE Transactions on Dependable and Secure Computing*, 18(3):1474–1488, 2019. 3

[72] Rey Wiyatno and Anqi Xu. Maximal jacobian-based saliency map attack. *arXiv preprint arXiv:1808.07945*, 2018. 2

[73] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018. 2

[74] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. Patchattack: A black-box texture-based attack with reinforcement learning. In *European Conference on Computer Vision*, pages 681–698. Springer, 2020. 3

[75] Bangjie Yin, Wenxuan Wang, Taiping Yao, Junfeng Guo, Zelun Kong, Shouhong Ding, Jilin Li, and Cong Liu. Adv-makeup: A new imperceptible and transferable attack on face recognition. *arXiv preprint arXiv:2105.03162*, 2021. 2, 3

[76] Yanghao Zhang, Wenjie Ruan, Fu Wang, and Xiaowei Huang. Generalizing universal adversarial attacks beyond additive perturbations. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1412–1417. IEEE, 2020. 2

[77] Zhendong Zhang, Cheolkon Jung, and Xiaolong Liang. Adversarial defense by suppressing high-frequency components. *arXiv preprint arXiv:1908.06566*, 2019. 2, 3, 8

[78] Yiqi Zhong, Xianming Liu, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15345–15354, 2022. 2, 3, 6, 7, 8

[79] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 12, 13

A. Results of Cross-model Transferability

We additionally evaluate the cross-model transferability of our attack method on the Object Detection and Instance Segmentation task. For Object Detection, we choose Faster-RCNN [58] to be the target networks and transfer the adversarial examples to YOLOv3 [57] and Deformable DETR [79]. For Instance Segmentation, we choose Mask-RCNN [26] to be the target networks and transfer the adversarial examples to PointRend [37] and SOLO [70]. The results are shown on Tab. S1.

Attack Methods	mAP %↓		
	FR-RCNN [58]	YOLO [57]	DETR [79]
clean	40.3	32.8	46.6
DPatch [44]	8.8*	12.3	20.2
AdvPatch [40]	5.5*	12.5	15.4
UAP [52]	12.1*	11.5	13.5
Rain Streaks (Ours)	**4.2***	7.8	**9.2**
Lens Dirt (Ours)	5.3*	10.2	12.4
Snow Flakes (Ours)	4.9*	8.3	9.7
Rain Drops (Ours)	5.8*	**7.7**	11.0

(a) Object Detection (Target Network: Faster-RCNN)

Attack Methods	mAP %↓		
	Mk-RCNN [26]	PtRend [37]	SOLO [70]
clean	36.4	37.1	34.9
UAP [52]	6.3*	9.7	8.5
Rain Streaks (Ours)	**2.1***	7.8	**7.3**
Lens Dirt (Ours)	3.0*	10.1	9.4
Snow Flakes (Ours)	2.4*	**7.5**	8.3
Rain Drops (Ours)	2.5*	8.5	8.8

(b) Instance Segmentation (Target Network: Mask-RCNN)

Table S1. The performances of cross-model transferability on the object detection and instance segmentation task with other attack methods. The white box attack results are marked with $*$.

B. Explainability and Sensitivity Analysis

B.1. PQ-GAN Architecture

To assist further analysis, we provide the details of the proposed PQ-GAN architecture in Fig. S2. In *Eq.* 5 and 6, we introduce the input and output of the generators $\mathrm{G_{HS}}$ and $\mathrm{G_{VS}}$. To better extract the spacial relation between patches, we concatenate $p^{\text{raw}}_{2a\pm1,2b+1}$ and a zero patch of the same size to get $P^{\text{input}}_{\text{HS}}$ of scale $h \times 3w$ to be the input of $\mathrm{G_{HS}}$, where h, w is the pre-defined patch size. Similarly, we concatenate $p^{\text{raw}}_{2a\pm1,2b\pm1}$, $p^{\text{raw}}_{2a,2b\pm1}$ and three zero patches to get $P^{\text{input}}_{\text{VS}}$ of scale $3h \times 3w$ to be the input of $\mathrm{G_{VS}}$, by Examples of $P^{\text{input}}_{\text{HS}}$ and $P^{\text{input}}_{\text{VS}}$ are depicted in Fig. S1.

B.2. Explainability of the Patch-wise Smoothness

$\mathrm{G_{HS}}$ and $\mathrm{G_{VS}}$ generate horizontal smoothness patches and vertical smoothness patches by considering their neighbor patches, *i.e.*, each $p^{\text{raw}}_{2a-1,2b} \in P_{\text{HS}}$ is generated conditioned to $P^{\text{input}}_{\text{HS}}$ which contains $p^{\text{raw}}_{2a\pm1,2b+1}$, while each $p^{\text{raw}}_{2a,2} \in P_{\text{VS}}$ is conditioned to $P^{\text{input}}_{\text{VS}}$ which contains $p^{\text{raw}}_{2a\pm1,2b\pm1}$ and $p^{\text{raw}}_{2a,2b\pm1}$. We draw the attention map of $P^{\text{input}}_{\text{HS}}$ and $P^{\text{input}}_{\text{VS}}$ to show how $\mathrm{G_{HS}}$ and $\mathrm{G_{VS}}$ pay attention to the neighbor patches, which is shown in Fig. S1.

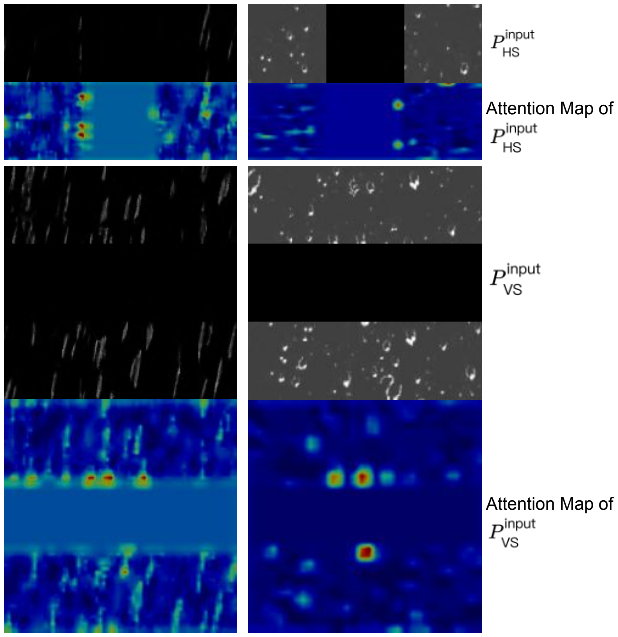

$P^{\text{input}}_{\text{HS}}$
Attention Map of $P^{\text{input}}_{\text{HS}}$
$P^{\text{input}}_{\text{VS}}$
Attention Map of $P^{\text{input}}_{\text{VS}}$

Figure S1. Visualization of $P^{\text{input}}_{\text{HS}}$ and $P^{\text{input}}_{\text{VS}}$ with their attention map obtained by applying Ramaswamy *et al.* [56]. $\mathrm{G_{HS}}$ and $\mathrm{G_{VS}}$ pay the most attention to the edges between generated and empty patches, which results in the smoothness between neighbor patches.

B.3. Influence of the Latent Vector's Dimensions

The adversarial attack strength is usually highly affected by the degree of freedom. For example, in traditional noise-based adversarial attack algorithms, tighter pixel-wise l_p constraint usually leads to weaker attack strength. An extreme case is that one-pixel attack algorithm [65] has much lower attack strength than global attack algorithms. Instead of modifying the image pixel-wisely, our method modifies the latent embedding $\mathcal{Z}$ of PQG. We want to see how the dimension k of the latent embedding $\mathcal{Z}$ affects the attack strength. We train PQG using Rain Streak samples with four different dimensions $k = \{8, 32, 128, 512\}$ and use it to evaluate the white-box attack performance and black-box transferability on the ImageNet classification task. We can see from Fig. S3 that the classification accuracy decrease as the k increases. It is especially influential under the white box scenario and when k is small. The classification accuracy tends to be steady as k goes above 128.

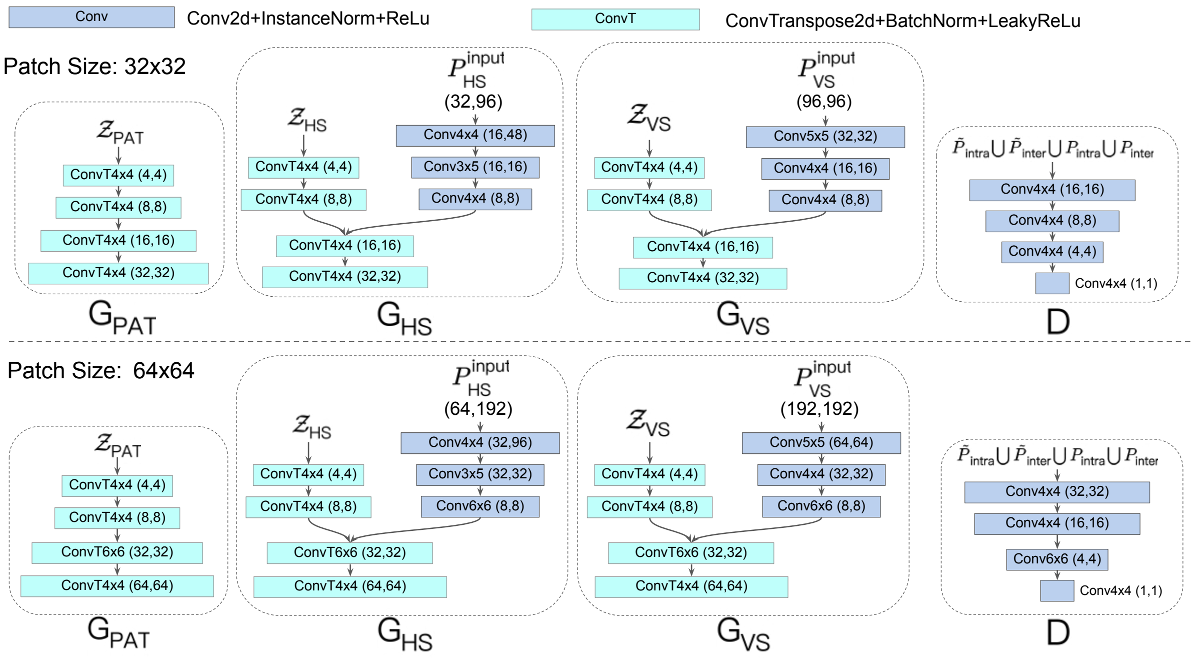


Figure S2. Two PQ-GAN architecture of patch size (32x32) and (64x64).

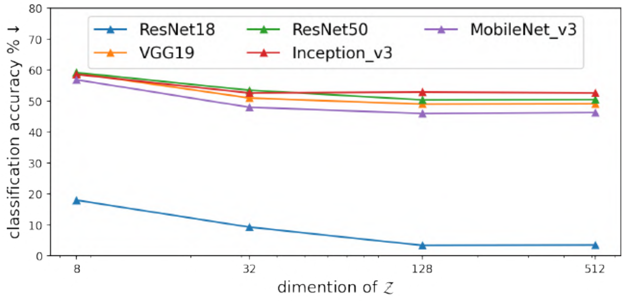


Figure S3. The white-box and black-box attack performance evaluation on the PQG *w.r.t* to the different dimension (8, 32, 128 and 512) of the latent embeddings.

C. Hyper-parameters Setups

PQ-GAN: We conduct two different patch sizes, 32 × 32 and 64 × 64. We use patch size of 64 × 64 for training the PQGAN of the rain streaks and snow flakes patterns and 32 × 32 for training the PQGAN of the rain drops and lens dirt patterns. The dimension k of each latent vector in $\mathcal{Z}$ is 128. During PQG training time, PQG generates images of size 256 × 256 with batch size set to be 4. The whole dataset (2000 images) is iterated 8 times. For the Inter-Patch Smoothness Loss, we set $M_{\text{inter}} = 32$. Following the training setup of the Wasstarian GAN with gradient penalty [25], we use $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0, \beta_2 = 0.9$, where λ is the coefficient of the gradient penalty; n_{critic} is the ratio of generator updates to each discriminator updates; α is the learning rate; β_1 and β_2 are the hyperparameters of the Adam [36] optimizer. For more information, please refer to Gulrajani *et al.* [25].

PQAttack: The synthesis strategy of rain drops, snow flakes, and lens dirt patterns are pixel-wise addition are formulated as $I^{adv} = I + \gamma * P^I$, where $\gamma = 0.3$ is used for rain drops and snow flakes patterns, $\gamma = -0.3$ is used for lens dirt patterns. For rain streaks patterns, we performed depth-aware synthesis according to Hu *et al.* [31], where we use $\alpha = 0.03$, $\beta = 0.04$. During the PQG-based Attack, each attack loop consists 300 iterations by default, where Adam [36] optimizer is employed with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a starting learning rate $lr = 0.03$. We also use cosine annealing scheduler [45] for learning rate decay.

Pre-trained Target Networks: All image classification pre-trained weights are obtained from the PyTorch model library. Pretrained weights of Mask-RCNN and Faster-RCNN are obtained from the OpenMMLab. YOLOv3 [57], Deformable BETR [79], PointRend [37], and SOLO [70] are trained using Cityscapes [11] standard training set with pre-trained weights on COCO dataset [42]. Deformable BETR, PointRend, and SOLO are trained with batch size 8 for 64 epochs, while YOLOv3 are trained with batch size 32 for 273 epochs.

Attack Methods: We follow the standard hyper-parameter setup for all attack methods.

D. Patterns generated by PQG

Our proposed PQG can be used to generate patterns of any size without retraining the network. We display patterns of size 896× 896 (A), 512× 512 (B), 1024 × 256 (C), 128 × 1000 (D), 384 × 512 (E), 128 × 408 (F) in Fig. S4 and S5.

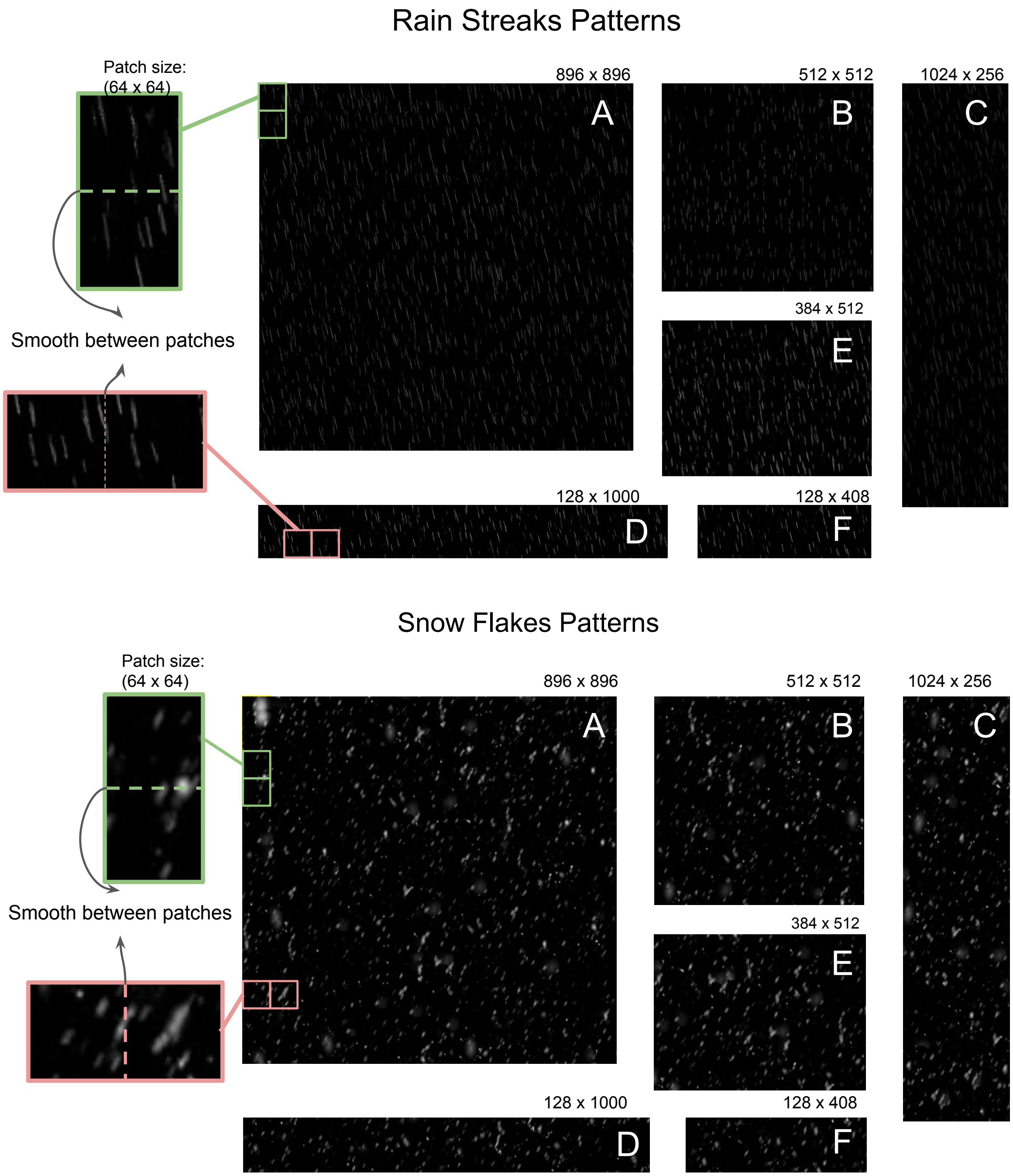


Figure S4. Rain streaks and snow flakes patterns generated by our proposed PQG with patch size equals 64. Neighbor patches (marked in yellow and pink rectangles) are connected smoothly.

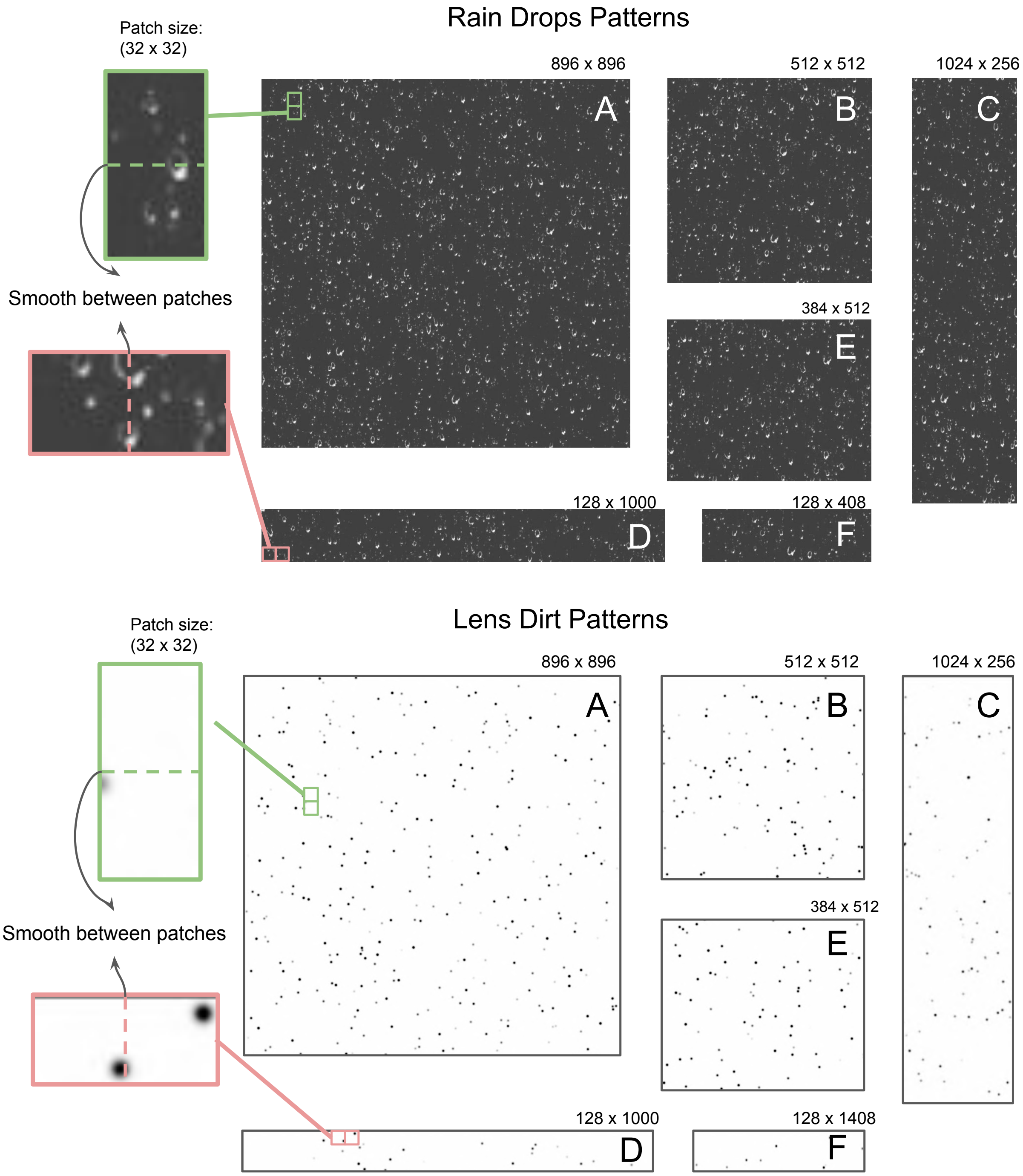


Figure S5. Snow Flakes and lens dirt patterns generated by our proposed PQG with patch size equals 32. Neighbor patches (marked in yellow and pink rectangles) are connected smoothly.

E. Visualization of Our Adversarial Examples

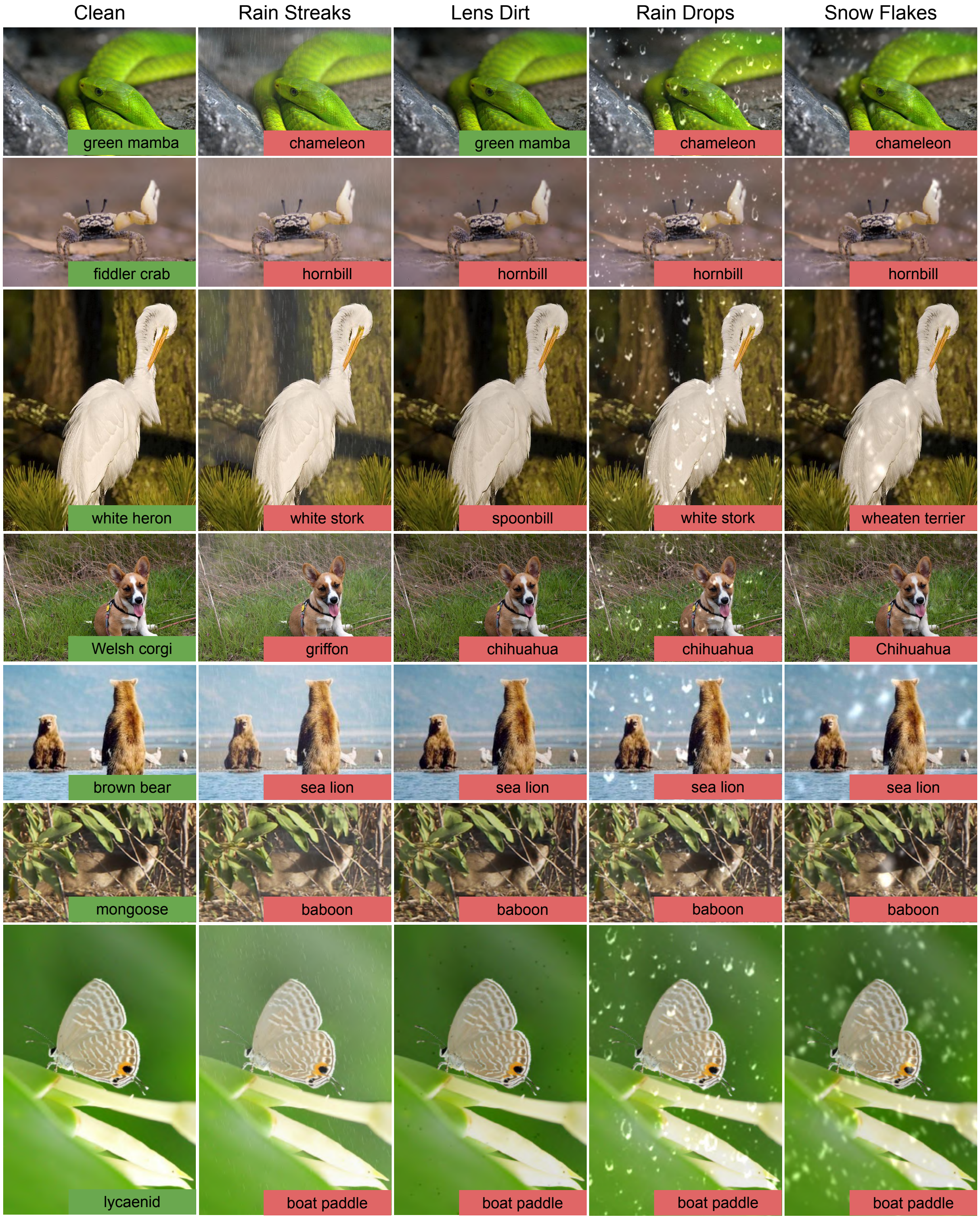


Figure S6. Adversarial Examples of ImageNet dataset. The classification results are shown in the bottom right of each image.

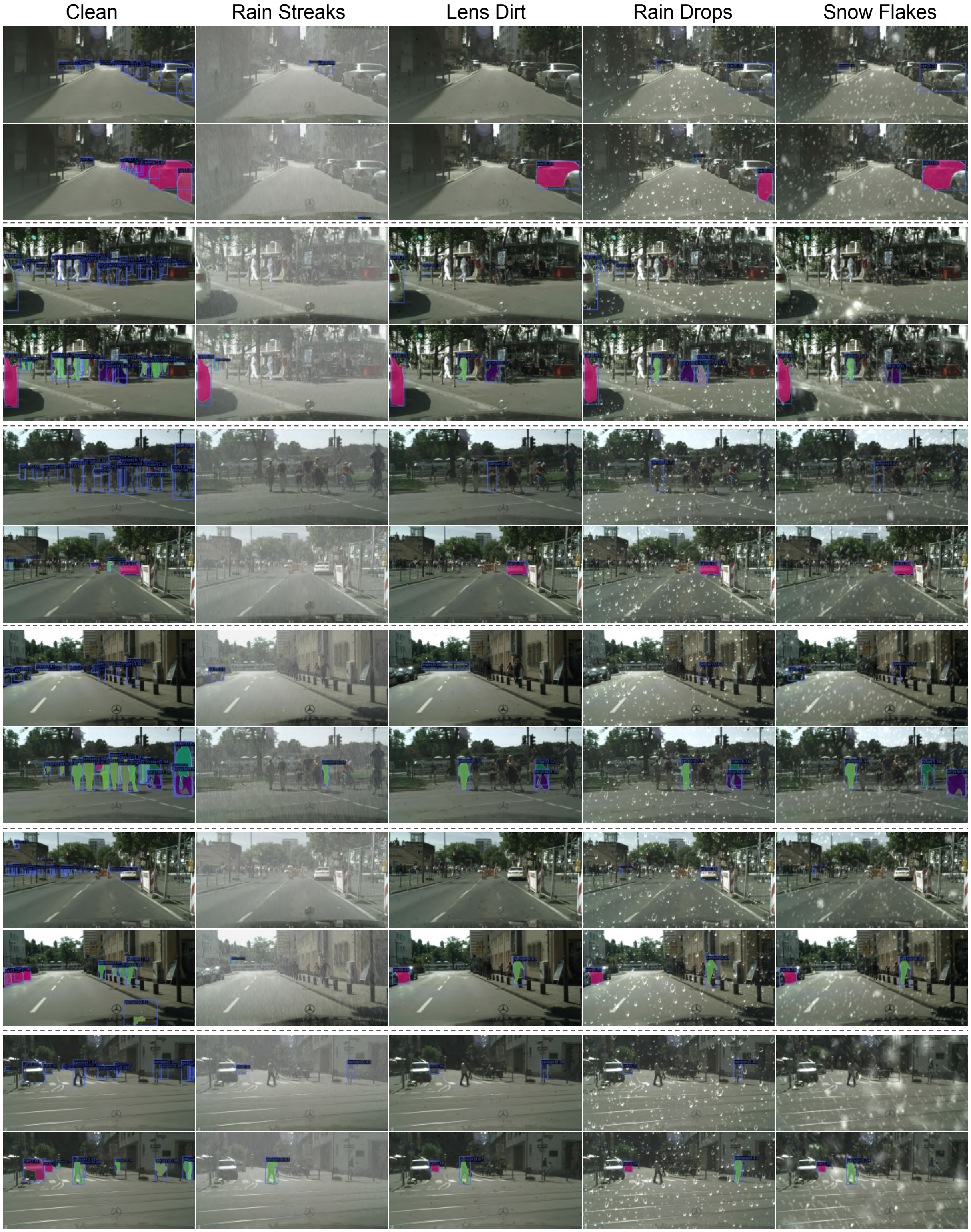


Figure S7. Adversarial Examples of CityScapes dataset. The Object Detection and Instance Segmentation results are shown in the odd and even rows, respectively.