Dynamic Generative Targeted Attacks with Pattern Injection

Weiwei Feng[1,*], Nanqing Xu[1,*], Tianzhu Zhang[1,2,†], Yongdong Zhang[1]
[1] University of Science and Technology of China, [2] Deep Space Exploration Lab
fengww@mail.ustc.edu.cn, xnq@mail.ustc.edu.cn, {tzzhang, zhyd73}@ustc.edu.cn

Abstract

Adversarial attacks can evaluate model robustness and have been of great concern in recent years. Among various attacks, targeted attacks aim at misleading victim models to output adversary-desired predictions, which are more challenging and threatening than untargeted ones. Existing targeted attacks can be roughly divided into instance-specific and instance-agnostic attacks. Instance-specific attacks craft adversarial examples via iterative gradient updating on the specific instance. In contrast, instance-agnostic attacks learn a universal perturbation or a generative model on the global dataset to perform attacks. However, they rely too much on the classification boundary of substitute models, ignoring the realistic distribution of the target class, which may result in limited targeted attack performance. And there is no attempt to simultaneously combine the information of the specific instance and the global dataset. To deal with these limitations, we first conduct an analysis via a causal graph and propose to craft transferable targeted adversarial examples by injecting target patterns. Based on this analysis, we introduce a generative attack model composed of a cross-attention guided convolution module and a pattern injection module. Concretely, the former adopts a dynamic convolution kernel and a static convolution kernel for the specific instance and the global dataset, respectively, which can inherit the advantages of both instance-specific and instance-agnostic attacks. And the pattern injection module utilizes a pattern prototype to encode target patterns, which can guide the generation of targeted adversarial examples. Besides, we also provide rigorous theoretical analysis to guarantee the effectiveness of our method. Extensive experiments demonstrate that our method shows superior performance than 10 existing adversarial attacks against 13 models.

1. Introduction

With the encouraging progress of deep neural networks (DNNs) in various fields [6, 19, 16, 44, 42], recent studies [36, 29, 28, 38] have corroborated that adversarial examples generated with small-magnitude perturbations can mislead the DNNs to make incorrect predictions. Due to the vulnerability of DNNs, adversarial attacks expose a security threat to real application systems based on deep neural networks, especially in some sensitive fields such as autonomous driving [27], face verification and financial systems [43], to name a few. Therefore, adversarial attacks have become a research hotspot over the past decade [2, 48, 35, 3, 33], which are significant in demonstrating the adversarial robustness and stability of deep learning models.

To further understand adversarial examples, there are tremendous works [1, 48, 33, 4, 36, 38] focusing on adversarial attacks. Recently, it has been found that adversarial examples possess an intriguing property of transferability, which indicates that adversarial examples generated for a white-box model can also mislead another black-box model. Due to this inherent property, black-box attacks become workable in real-world scenarios where attackers cannot access the attacked model. Extensive methods, such as

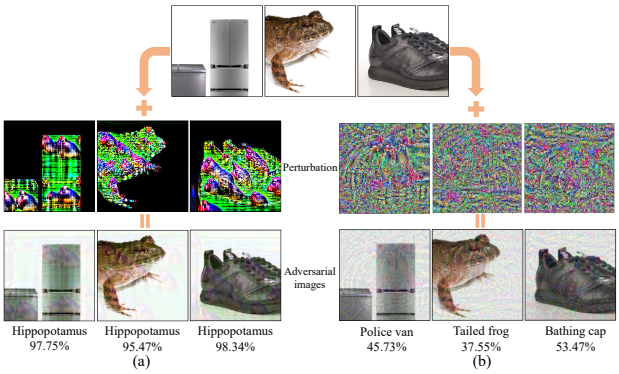

Figure 1. Visualization comparison between adversarial examples generated by our method (a) and the instance-specific method MIM [9] (b). Our perturbations (a) not only show an underlying dependency with the input instance, but also have strong semantic patterns or styles of the target class ("Hippopotamus"). In contrast, the perturbations generated by MIM perform like random noises. The adversarial examples are generated against ResNet-152, and labels are predicted by another unknown model (VGG-16).

* Equal Contribution
† Corresponding Author

MIM [9], DIM [53] and CD-AP [40], have made an impressive performance of boosting the transferability for untargeted attacks, which mislead the model to make an incorrect classification without specifying a target class. However, targeted attacks are more challenging compared with untargeted attacks, making the model output the adversary-desired target class. It is claimed in recent works [14, 55] that transferable targeted attacks are more worthy of study because attackers can directly control the unknown model to output the desired prediction, which can expose huge threats to data privacy and security. Therefore, it is of great significance to develop transferable targeted attacks.

Existing methods of transferable targeted attacks can be roughly categorized into instance-specific [9, 14, 13, 54, 50, 34, 31] and instance-agnostic [52, 35, 27, 40, 39] attacks. Specifically, almost all instance-specific attacks craft adversarial examples via iterative gradient updating, where attackers can only take advantage of the specific input instance, the white-box model and the target class label. Instance-specific attacks rely on optimizing the classification score of the adversary-desired class label to perturb the specific instance, which ignore the global data distribution. As a result, they inevitably lead to adversarial examples over-fitting the white-box model and result in modest transferability of targeted attacks. On the other hand, via learning a universal perturbation [37] or a generative attack model [52, 55], instance-agnostic attacks optimize adversarial perturbations on the global data distribution rather than the specific instance. To a certain extent, they can alleviate the problem of data-specific over-fitting and lead to more transferable targeted adversarial examples. However, taking the generative attack methods as an example, they suffer from two limitations. **(1)** Most generative attacks [52, 40, 35] still rely on the target label and the classification boundary information of white-box models rather than the realistic data distribution of the target class. Consequently, it is claimed that most generative attacks still have the possibility of over-fitting the white-box model, which may result in limited performance of transferable targeted attacks. **(2)** Another limitation is that existing generative attacks [39, 55, 27] apply the same network weights to every input instance in the test dataset. Nevertheless, it is considered that the shared network weights cannot stimulate the best attack performance of generative models [39, 55, 55]. Thus these aforementioned limitations have become the bottleneck of developing transferable targeted attacks, to a certain extent.

To address the aforementioned limitations, in this paper we construct a causal graph to formulate the prediction process of classifiers, and analyze the origin of adversarial examples. Based on this analysis, we propose to generate targeted adversarial examples via *injecting the specific pattern or style of the target class*. To this end, we introduce a generative attack model, which can not only inject pattern or style information of the target class to improve transferable targeted attacks, but also learn specialized convolutional kernels for each input instance. Specifically, we designed a cross-attention guided convolution module and a pattern injection module in the proposed generative attack model. **(1)** The cross-attention guided convolution module consists of a static convolutional kernel and a dynamic convolutional kernel that is computed according to the input instance. Consequently, this static and dynamic mixup module can not only encode the global information of the dataset, but also learn specialized convolutional kernels for each input instance. This paradigm makes our generative model inherit the advantages of both instance-specific and instance-agnostic attacks. **(2)** The pattern injection module is designed to model the pattern or style information of the target class and guide the generation of targeted adversarial examples. Concretely, we propose a pattern prototype to learn a global pattern representation over images from the target class, and use the prototype to guide the generation of more transferable targeted adversarial examples. And the generated adversarial images of our method are presented in Figure 1. It is observed that our generated perturbations (as shown in Figure 1(a)) pose strong semantic patterns or styles of the target class and show an underlying dependency on the input instance. In contrast, the perturbations (as presented in Figure 1(b)) generated by MIM [9] perform like random noises. Finally, to further understand our method, we provide rigorous theoretical analysis to guarantee the effectiveness of our method, as shown in Section 3.4, where we derive a concise conclusion based on the problem of Gaussian binary classification.

In summary, the main contributions of this paper are three-fold: **(1)** We propose a dynamic generative model to craft transferable targeted adversarial examples, which can not only inject pattern or style information of the target class to improve transferable targeted attacks, but also learn specialized convolutional kernels for each input instance. Besides, our method inherits the advantages of both instance-specific and instance-agnostic attacks, and to the best of our knowledge, we are the first to bridge them. **(2)** We state that *injecting the specific pattern or style of the target class can improve the transferability of targeted adversarial examples*, and we provide a comprehensive theoretical analysis to verify the rationality of this statement. **(3)** Extensive experimental results demonstrate that our method significantly boosts the transferability of targeted adversarial examples over 10 existing methods against 13 models.

2. Related Work

Instance-specific Attacks. As the pioneering work [48] exposes the vulnerability of neural networks, many recent methods [54, 17, 3, 11, 29] utilize gradient-based optimiza-

tion to generate input-dependent adversarial examples. To further boost the transferability of the adversarial example, several works have been proposed by various strategies to optimize the gradient update process. MIM [9] introduces a momentum term during the iterative gradient updating. DIM [53] improves the transferability via integrating diverse input patterns for optimizing, and TIM [10] performs a convolution operation on the gradient that is applicable to any gradient-based attack methods. Besides, some recent works [5, 18, 20] propose to ensemble multiple pre-trained substitute models to craft more transferable adversarial examples. However, these methods pose modest transferability under the targeted attacks setting, because they rely too much on the target label and the classification boundary information of white-box models. Meanwhile, among these methods, they all face the problem of data-specific over-fitting, because of ignoring the global data distribution. To overcome these limitations, in this paper, we propose a dynamic generative model to craft more transferable targeted adversarial examples, which can not only encode the global dataset, but also adapt to the specific instance.

Instance-agnostic Attacks. Distinguished from instance-specific attacks, instance-agnostic attacks learn a universal perturbation [37] or a generative function [35, 27, 51, 41, 55, 40, 39] to craft adversarial examples. Compared with the universal perturbation, the latter is more flexible and has drawn more attention in recent years. Xiao *et al.* [52] first explore generating adversarial examples with GANs [16], which can consider the whole dataset instead of the specific input instance. CD-AP [40] and TTP [39] exploit generative models to craft more transferable adversarial examples. Recently, Yang *et al.* [55] boost the transferability of targeted adversarial examples with hierarchical generative networks. To a certain extent, these methods lead to more general and transferable adversarial examples because of alleviating the data-specific over-fitting problem. However, they also suffer from some limitations. First, existing generative attack methods [52, 57, 55, 40, 39] apply the same network weights to each input instance, which may limit the transferability of adversarial examples. And the second is that most of them also rely too much on the target label and the classification boundary of white-box models, ignoring the realistic data distribution of the target class. Motivated by these, in this paper, we propose a dynamic generative attack model, which can not only inject pattern or style information of the target class to improve transferable targeted attacks, but also learn specialized convolutional kernels for each input instance.

3. Method

In this section, we present the overall scheme of our proposed method, including some preliminaries, motivations, network architectures and theoretical analyses.

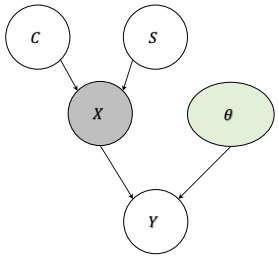

Figure 2. The casual graph of model inference. Each node is a random variable, where $C, S, \boldsymbol{x}, \boldsymbol{y}$ and θ represent content, style or pattern, the input image, the prediction label, and model parameters, respectively. In order to achieve good performance, the model will learn the statistical correlation between the style S and the label $\boldsymbol{y}$.

3.1. Preliminaries and Motivation

We denote the clean image as $\boldsymbol{x} \in \mathcal{X}$, and the attacked model parameterized by θ is denoted as $f : \mathcal{X} \to \mathcal{Y}$. The goal of target attacks is to generate an adversarial perturbation $\boldsymbol{\delta}$ that can make the model misclassify the input $\boldsymbol{x}_{adv} = \boldsymbol{x} + \boldsymbol{\delta}$ to the predefined class $\boldsymbol{y}_t$, which can be formulated as $f(\boldsymbol{x}_{adv}) = \boldsymbol{y}_t \neq f(\boldsymbol{x})$. To gain insights into the origin of targeted adversarial examples, we construct a casual graph $\mathcal{G}$ to formulate the inference of deep learning models, as shown in Figure 2. For the input images $\boldsymbol{x}$, we propose to group the whole causes of $\boldsymbol{x}$ into two categories, content-related cause C and content-independent cause S that can be dubbed as style or pattern cause. This indicates that $C \to \boldsymbol{x} \leftarrow S$, and $C \perp S$. According to human visual intuition, only the content variable C is relevant to the prediction class $\boldsymbol{y}$. Consequently, we can use the Law of total probability to expand the prediction $P_\theta(\boldsymbol{y} \mid \boldsymbol{x})$ as:

$$P_\theta(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{s \in \mathbb{S}} P_\theta(s \mid \boldsymbol{x}) P_\theta(\boldsymbol{y} \mid \boldsymbol{x}, s). \tag{1}$$

But several recent works [56] and Equation (1) imply that deep learning models can learn not only the dependencies between the content C and the label $\boldsymbol{y}$, but also the statistical correlation between the style S and the label $\boldsymbol{y}$ (*i.e.*, $P_\theta(\boldsymbol{y} \mid \boldsymbol{x}, s)$). This phenomenon indicates that deep learning models can capture some detailed texture patterns and style features to achieve good performance, and the output prediction $P_\theta(\boldsymbol{y} \mid \boldsymbol{x})$ will change with the statistical correlation between $\boldsymbol{y}$ and S (*i.e.*, $P_\theta(\boldsymbol{y} \mid \boldsymbol{x}, s)$). And this property certainly offers attackers an opportunity to generate adversarial examples for misleading the model. Therefore, the inference of targeted adversarial examples can be formulated as:

$$P_\theta(\boldsymbol{y}_t \mid \boldsymbol{x}_{adv}) = \sum_{s \in \mathbb{S}} P_\theta(s \mid \boldsymbol{x}_{adv}) P_\theta(\boldsymbol{y}_t \mid \boldsymbol{x}_{adv}, s). \tag{2}$$

Inspired by Figure 2 and the aforementioned Equation (2), we first propose to exploit the statistical correlation between $\boldsymbol{y}_t$ and S (*i.e.*, $P_\theta(\boldsymbol{y}_t \mid \boldsymbol{x}_{adv}, s)$) [1], via *injecting the specific*

[1]The statistical correlation between $\boldsymbol{y}_t$ and S (*i.e.*, $P_\theta(\boldsymbol{y}_t \mid \boldsymbol{x}_{adv}, s)$ reflects that the model learns some non-robust features. According to the analysis of [24] and [55], the transferability of adversarial examples may arise from non-robust features, such as texture patterns or styles.

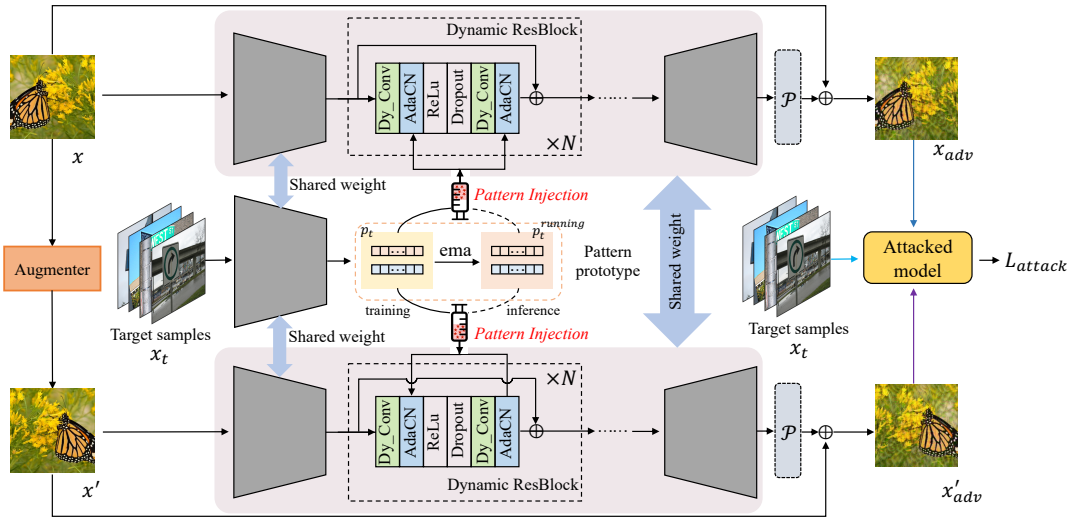

Figure 3. The architecture of our dynamic generative attack model, which is composed of a cross-attention guided convolution module (denoted as "Dy_Conv") and a pattern injection module. The cross-attention guided convolution module consists of a static convolutional kernel and a dynamic convolutional kernel that is computed according to the input instance. For the pattern injection, the pattern prototype $\boldsymbol{p}_t = \{\gamma_t, \beta_t\}$ encodes the patterns or styles of the target class, and the "AdaCN" layer performs an affine operation to inject $\boldsymbol{p}_t$ into the generation of adversarial examples. Similar to [25], we introduce $\boldsymbol{p}_t^{\text{running}}$ via the EMA updating based on $\boldsymbol{p}_t$.

style or pattern of images from the given target class $\boldsymbol{y}_t$ to generate targeted transferable adversarial examples.

To this end, we propose a generative model to craft targeted adversarial examples by injecting the pattern of the target class. Besides, we design a cross-attention guided dynamic convolution module in the generator, which makes our generative model inherit the advantages of both instance-specific and instance-agnostic attacks. Therefore, the formulation of our method can be denoted as

$$\boldsymbol{x}_{adv} = \text{clip}\left\{ \text{Proj}\left(\mathcal{W} * G_{\theta(\boldsymbol{x})}(\boldsymbol{x}, p_t), -\varepsilon, \varepsilon\right) + \boldsymbol{x} \right\}, \tag{3}$$

where ε is the perturbation budget, $\mathcal{W}$ is a smoothing operator with fixed weights, p_t represents the semantic pattern or style of the target class, and Proj is a projection operation.

3.2. Network Architecture

Specifically, our generative model is illustrated in Figure 3, which is composed of two modules: ① a cross attention-guided dynamic convolution module (corresponding to "Dy_Conv" in the ResBlock of Figure 3), ② a pattern injection module. And the detailed pipeline of our approach is presented in Algorithm 1. In the next section, we mainly describe the two designed modules in detail.

Cross-attention guided dynamic convolution module. Considering that not only the global dataset needs to be modeled, but also specialized convolution kernels need to be learned for each input, we design a cross-attention guided convolution module. To achieve this goal, as shown in Figure 4, we propose a static and a dynamic convolution kernel parameterized with W and ΔW, respectively[2]. Thus

[2]Note that we ignore bias terms for the sake of brevity

Algorithm 1: Dynamic Generative Targeted Attack

Input : A white-box model f, training data $\mathcal{X}$, target samples $\mathcal{X}_t$, perturbation budget ϵ, training epoch T.

Output: Generative attack model G_θ.

for $t \leftarrow 0$ *to* T **do**
- Sample mini-batches $\boldsymbol{x} \sim \mathcal{X}$ and $\boldsymbol{x}_t \sim \mathcal{X}_t$.
- Get the augmented mini-batches $\boldsymbol{x}'$ from $\boldsymbol{x}$.
- Feed $\boldsymbol{x}$, $\boldsymbol{x}'$ and $\boldsymbol{x}_t$ into the generator to get adversarial examples $\boldsymbol{x}_{adv}$ and $\boldsymbol{x}'_{adv}$, where $\boldsymbol{x}_t$ is used for extracting and injecting target patterns p_t.
- Update p_t^{running} based on p_t via EMA updating.
- Forward pass $\boldsymbol{x}_{adv}$, $\boldsymbol{x}'_{adv}$ and $\boldsymbol{x}_t$ through f.
- Compute losses L_{attack}, and backward for updating G_θ.

return Converged generator G_θ.

the designed module is formulated as:

$$\boldsymbol{X}_l = \text{conv}(\boldsymbol{X}_{l-1}; W + \Delta W), \tag{4}$$

where $\boldsymbol{X}_l$ is the output of layer l. To get dynamic convolution kernels, we design a series of learnable kernels $(N \times C_{in} \times C_{out} \times k^2)$ [3] and attention queries $\boldsymbol{q} \in R^{N \times C_{in}}$, which indicate the attention weight of each kernel. For the layer l, we compute attention weights via performing cross-attention between attention queries $\boldsymbol{q}$ and the input $\boldsymbol{X}_{l-1} \in R^{C_{in} \times H \times W}$, which can be indicated as:

$$att = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \tag{5}$$

[3]N indicates the number of learnable kernels, k represents the kernel size, C_{in} and C_{out} are the number of input and output channels.

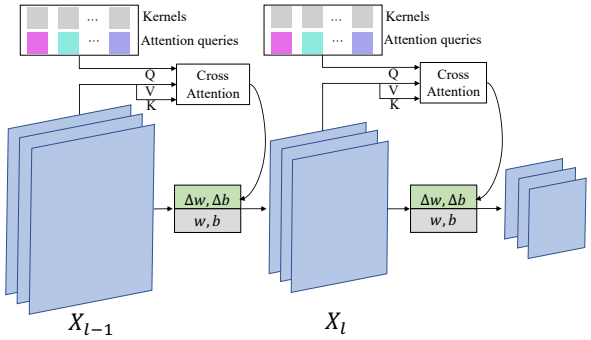

Figure 4. The illustration of the proposed cross-attention guided dynamic convolution module, which is composed of a static convolution kernel W and a dynamic convolution kernel ΔW.

where $att \in R^{N\times C_{in}}$, $Q = \boldsymbol{q}W^q$, $K = \boldsymbol{X}_{l-1}^{\top}W^k$ and $V = \boldsymbol{X}_{l-1}^{\top}W^v$. Then we can get the attention weight $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_N] \in R^{1\times N}$ by conducting the average pooling on att, and each element in $\boldsymbol{\alpha}$ indicates the weight of corresponding kernels. Therefore, the dynamic convolution kernels are implemented as $\Delta W = \alpha_1 * W_1 + \alpha_2 * W_2 + \cdots + \alpha_N * W_N$ [4]. Benefiting from this dynamic and static mixup convolution operation, our proposed generative attack model can inherit the advantages of both instance-specific and instance-agnostic attacks.

Pattern injection module. First, to encode target patterns and styles during the generation of adversarial examples, we design a pattern prototype $\boldsymbol{p}_t = \{\boldsymbol{\gamma}_t, \boldsymbol{\beta}_t\}$, which is extracted from the features of target samples, as shown in Figure 3. Following [25], we introduce $\boldsymbol{p}_t^{\text{running}}$ to model patterns of the global samples for the target class. Note that we update $\boldsymbol{p}_t^{\text{running}}$ during each training iteration, which can be indicated as $\boldsymbol{p}_t^{\text{running}} = \lambda \boldsymbol{p}_t + (1-\lambda)\boldsymbol{p}_t^{\text{running}}$. Then we propose an adaptive class normalization (corresponding to "AdaCN" in Figure 3) to inject $\boldsymbol{p}_t$ into the generation of target adversarial examples. Similar to [23], the AdaCN module exploits the learned $\boldsymbol{p}_t = \{\boldsymbol{\gamma}_t, \boldsymbol{\beta}_t\}$ to perform an affine operation for injecting target patterns or styles:

$$\text{AdaCN}(\boldsymbol{X}) = \boldsymbol{\gamma}_t \left(\frac{\boldsymbol{X} - \mu(\boldsymbol{X})}{\sigma(\boldsymbol{X})}\right) + \boldsymbol{\beta}_t, \tag{6}$$

where $\boldsymbol{p}_t = \{\boldsymbol{\gamma}_t, \boldsymbol{\beta}_t\}$ is the target pattern prototype. Note that Equation (6) represents pattern injection during training, and it is necessary to replace $\boldsymbol{p}_t = \{\boldsymbol{\gamma}_t, \boldsymbol{\beta}_t\}$ with $\boldsymbol{p}_t^{\text{running}} = \{\boldsymbol{\gamma}_t^{\text{running}}, \boldsymbol{\beta}_t^{\text{running}}\}$ during inference.

3.3. Objective Function

To lead the victim model to misclassify adversarial examples as the target class, it is necessary to make the output distribution of adversarial examples $f(\boldsymbol{x}_{adv})$,$f(\boldsymbol{x}'_{adv})$ and target samples $f(\boldsymbol{x}_t)$ consistent. Thus we define the loss function as follows:

$$\begin{aligned} \mathcal{L} &= D_{KL}\big(f(\boldsymbol{x}_{adv})\|f(\boldsymbol{x}_t)\big) + D_{KL}\big(f(\boldsymbol{x}_t)\|f(\boldsymbol{x}_{adv})\big) \\ \mathcal{L}_{aug} &= D_{KL}\big(f(\boldsymbol{x}'_{adv})\|f(\boldsymbol{x}_t)\big) + D_{KL}\big(f(\boldsymbol{x}_t)\|f(\boldsymbol{x}'_{adv})\big) \end{aligned} \tag{7}$$

Besides, similar to [39], we also introduce a local similarity loss. For a batch of generated adversarial examples $\{\boldsymbol{x}_{adv}^i\}_{i=1}^n$, $\{\boldsymbol{x}_{adv}^{\prime,i}\}_{i=1}^n$ and target samples $\{\boldsymbol{x}_t^i\}_{i=1}^n$, the similarity matrix can be computed as $\mathcal{S}_{i,j} = \frac{f(\boldsymbol{x}_{adv}^i)\cdot f(\boldsymbol{x}_{adv}^{\prime,j})}{\|f(\boldsymbol{x}_{adv}^i)\|\|f(\boldsymbol{x}_{adv}^{\prime,j})\|}$ and $\mathcal{S}_{i,j}^t = \frac{f(\boldsymbol{x}_t^i)\cdot f(\boldsymbol{x}_t^j)}{\|f(\boldsymbol{x}_t^i)\|\|f(\boldsymbol{x}_t^j)\|}$. Hence, we can get the local similarity loss as:

$$\mathcal{L}_{sim} = \sum_{i,j} \overline{\mathcal{S}}_{i,j}^t \log \frac{\overline{\mathcal{S}}_{i,j}^t}{\overline{\mathcal{S}}_{i,j}} + \sum_{i,j} \overline{\mathcal{S}}_{i,j} \log \frac{\overline{\mathcal{S}}_{i,j}}{\overline{\mathcal{S}}_{i,j}^t}, \tag{8}$$

where $\overline{\mathcal{S}}_{i,j} = \frac{\exp(\mathcal{S}_{i,j})}{\sum_k \exp(\mathcal{S}_{i,k})}$ and $\overline{\mathcal{S}}_{i,j}^t = \frac{\exp(\mathcal{S}_{i,j}^t)}{\sum_k \exp(\mathcal{S}_{i,k}^t)}$. Finally, the total objective function can be formulated as:

$$\mathcal{L}_{attack} = \mathcal{L} + \mathcal{L}_{aug} + \mathcal{L}_{sim}. \tag{9}$$

3.4. Theoretical Analyses

In order to further explain the insights of our method, we consider a concrete setting that allows us to theoretically investigate why our method is effective for boosting target attacks. Please refer to the **Supplementary Material** for comprehensive proofs of the following theorem.

Setup. We consider a simple problem of maximum likelihood classification, similar to that of [24], between two Gaussian distributions.

$$\mathcal{X}_s \sim N\left(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s\right), \quad \mathcal{X}_t \sim N\left(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\right), \tag{10}$$

where $\boldsymbol{\Sigma}_s, \boldsymbol{\Sigma}_t = \text{diag}\left\{\sigma_{s_1}^2, \cdots \sigma_{s_n}^2\right\}, \text{diag}\left\{\sigma_{t_1}^2, \cdots \sigma_{t_n}^2\right\}$, respectively. To simplify the following derivation, we introduce a mapping and a translation operation to transform the model into a standard Gaussian binary classification model, which can be denoted as:

$$\begin{aligned} \mathcal{F}_s &= \boldsymbol{A}\mathcal{X}_s - \tfrac{\boldsymbol{A}\boldsymbol{\mu}_s + \boldsymbol{\mu}_t}{2} \sim N\left(-\boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \\ \mathcal{F}_t &= \boldsymbol{E}\mathcal{X}_t - \tfrac{\boldsymbol{A}\boldsymbol{\mu}_s + \boldsymbol{\mu}_t}{2} \sim N\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \end{aligned} \tag{11}$$

where the mapping matrix $\boldsymbol{A} = \text{diag}\left\{\frac{\sigma_{t_1}}{\sigma_{s_1}}, \frac{\sigma_{t_2}}{\sigma_{s_2}} \cdots, \frac{\sigma_{t_n}}{\sigma_{s_n}}\right\}$, $\boldsymbol{E}$ is the identity matrix and $\boldsymbol{\mu} = \frac{-\boldsymbol{A}\boldsymbol{\mu}_s + \boldsymbol{\mu}_t}{2}, \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_t$. To perform targeted attacks against this model with a given sample $\boldsymbol{x}_s \in \mathcal{X}_s$ and the target label t, where $\boldsymbol{A}\boldsymbol{x}_s - \frac{\boldsymbol{A}\boldsymbol{\mu}_s + \boldsymbol{\mu}_t}{2} = \boldsymbol{f}_s \in \mathcal{F}_s$ and $\boldsymbol{f}_s + \boldsymbol{\delta}^\star = \boldsymbol{A}(\boldsymbol{x}_s + \boldsymbol{\delta}) - \frac{\boldsymbol{A}\boldsymbol{\mu}_s + \boldsymbol{\mu}_t}{2}$, we aim to solve the following optimization problem:

$$\boldsymbol{\delta} = \boldsymbol{A}^{-1}\boldsymbol{\delta}^\star, \quad \boldsymbol{\delta}^\star = \underset{\|\boldsymbol{\delta}'\|_2^2 \leqslant \varepsilon^2}{\text{argmax}} \left(\boldsymbol{f}_s + \boldsymbol{\delta}'\right)^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}. \tag{12}$$

[4] $W_1, W_2 \cdots W_N$ are the learnable kernels.

Table 1. The attack success rates against normally trained models on ImageNet NeurIPS validation set, and the perturbation budget $\ell_\infty \leq 16/255$. Note that the results are averaged on 8 different target classes.

Substite Model	Method	Inc-v3	Inc-v4	Inc-Res-v2	Res152	Densenet-121	GoogleNet	Vgg-16
Inc-v3	MIM	**99.90**	0.80	1.00	0.40	0.20	0.20	0.30
	TI-MIM	98.50	0.50	0.50	0.30	0.20	0.40	0.40
	SI-MIM	99.80	1.50	2.00	0.80	0.70	0.70	0.50
	DIM	95.60	2.70	0.50	0.80	1.10	0.40	0.80
	TI-DIM	96.00	1.10	1.20	0.50	0.50	0.50	0.80
	SI-DIM	90.20	3.80	4.40	2.00	2.20	1.70	1.40
	CD-AP	94.20	57.60	60.10	37.10	41.60	32.30	41.70
	TTP	91.37	46.04	39.37	16.40	33.47	25.80	25.73
	C-GSP	93.40	66.90	**66.60**	41.60	46.40	40.00	45.00
	GAP	86.90	45.06	34.48	34.48	41.74	26.89	34.34
	Ours	94.63	**67.95**	55.03	**50.50**	**47.38**	**47.67**	**48.11**
Res152	MIM	0.50	0.40	0.60	**99.70**	0.30	0.30	0.20
	TI-MIM	0.30	0.30	0.30	96.50	0.30	0.40	0.30
	SI-MIM	1.30	1.20	1.60	99.50	1.00	1.40	0.70
	DIM	2.30	2.20	3.00	92.30	0.20	0.80	0.70
	TI-DIM	0.80	0.70	1.00	90.60	0.60	0.80	0.50
	SI-DIM	4.20	4.80	5.40	90.50	4.20	3.60	2.00
	CD-AP	33.30	43.70	42.70	96.60	53.80	36.60	34.10
	TTP	62.03	49.20	38.70	95.12	82.96	65.09	62.82
	C-GSP	37.70	47.60	45.10	93.20	64.20	41.70	45.90
	GAP	30.99	31.43	20.48	84.86	58.35	29.89	39.70
	Ours	**66.83**	**53.62**	**47.61**	96.48	**86.61**	**68.29**	**69.58**
Vgg-16	MIM	0.26	0.47	0.20	0.35	0.40	0.34	90.24
	TI-MIM	0.43	0.63	0.34	0.55	1.45	0.64	89.13
	SI-MIM	0.35	0.57	0.42	0.31	0.56	0.52	90.89
	DIM	0.75	1.30	0.55	1.00	1.88	1.03	**97.70**
	TI-DIM	0.23	0.38	0.17	0.29	0.48	0.35	93.71
	SI-DIM	0.87	1.12	0.70	0.95	1.89	1.55	91.42
	CD-AP	5.32	8.94	4.87	9.33	14.02	3.19	96.82
	TTP	22.51	17.14	**9.68**	22.68	40.87	22.41	97.59
	C-GSP	9.42	9.60	3.01	11.76	32.28	13.33	96.81
	GAP	3.11	5.26	1.50	5.08	11.23	2.70	93.00
	Ours	**28.18**	**21.78**	9.56	**25.27**	**46.55**	**23.70**	93.00

Taking advantage of the method of Lagrange multipliers, we can easily get the optimal solution as follows:

$$\begin{aligned} \boldsymbol{\delta}^\star &= \tfrac{1}{\lambda}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \\ \boldsymbol{\delta} &= \tfrac{\boldsymbol{\Sigma}^{-1}}{2\lambda}\left[\boldsymbol{A}^{-1}\big(\boldsymbol{A}\left(\boldsymbol{x}_s-\boldsymbol{\mu}_s\right)+\boldsymbol{\mu}_t\big)-\boldsymbol{x}_s\right]. \end{aligned} \tag{13}$$

Going a step further, we rewrite the solution of $\boldsymbol{\delta}$ into a more concise formula as:

$$\boldsymbol{\delta} = C_1\left[\begin{pmatrix} \frac{\sigma_{t_1}}{\sigma_{s_1}} & & \\ & \ddots & \\ & & \frac{\sigma_{t_n}}{\sigma_{s_n}} \end{pmatrix}\left(\boldsymbol{x}_s-\boldsymbol{\mu}_s\right)+\boldsymbol{\mu}_t\right]-C_2\boldsymbol{x}_s, \tag{14}$$

where $C_1 = \frac{\boldsymbol{\Sigma}^{-1}\boldsymbol{A}^{-1}}{2\lambda}$ and $C_2 = \frac{\boldsymbol{\Sigma}^{-1}}{2\lambda}$. In fact, note that the item of [redacted] represents target pattern or style injection, which is consistent with the previous works [23, 26]. Therefore, the formulation of Equation (14) shows a close underlying correlation between the optimal targeted adversarial perturbation and the embedding of target pattern or style, which also theoretically guarantees the effectiveness of our proposed generative model for targeted attacks.

4. Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of our method for targeted attacks under various settings. Please feel free to get more experimental results in our **Supplementary Material**.

4.1. Experimental Setup

Dataset. Following [55], we perform training on ImageNet [7] training set, and evaluate on ImageNet-NeurIPS (1k) proposed by NeurIPS 2017 adversarial competition [30].

Victim Models. We consider 13 attacked models in our experiments. The normally trained models includes Inception-v3 (Inc-v3) [47], Inception-v4 (Inc-v4) [45], DenseNet-121 [22], GoogleNet [46], ResNet-152 (Res-152) [19], and VGG16 (Vgg-16) [44] and Inception-ResNet (Inc-Res-v2) [45]. Besides, we also consider several models with robust training mechanisms, including adv-Inception-v3 (Adv-Inc-v3) [17], ens-adv-Inception-ResNet-v2 (Ens-Adv-IncRes-v2) [18] and ResNet-50 trained with various robust training tricks [21, 15].

Baseline Attacks. To illustrate the effectiveness of our method, we compare it with instance-specific attacks and instance-agnostic attacks. Instance-specific attacks mainly include MIM [9], DIM [53], SIM [32] and TIM [10], while

Table 2. The attack success rates against robust models on ImageNet NeurIPS validation set. The perturbation budget $\ell_\infty \leq 16/255$. Adv-Inc-v3 [29] and Ens-Adv-IncRes-v2 [49] are trained with the adversarial training mechanism. Res50_SIN (stylized ImageNet), Res50_SIN_IN (mixture of stylized ImageNet and Nature ImageNet) and Res50_SIN_fine_IN (mixture of stylized ImageNet and Nature ImageNet with finetuning tricks) are trained with auxiliary dataset [15], and Res50_Augmix [21] is trained with the state-of-the-art data augmentation approach. Note that the results are averaged on 8 different target classes.

Substite Model	Method	Adv-Inc-v3	Ens-IncRes-v2	Res50_SIN	Res50_SIN_IN	Res50_SIN_fine_IN	Res50_AugMix
Res152	MIM	0.19	0.15	0.28	1.58	2.75	0.78
	TI-MIM	0.61	0.73	0.50	2.51	4.75	1.76
	SI-MIM	0.24	0.24	0.39	0.66	0.84	0.36
	DIM	0.63	0.37	0.94	8.50	14.22	3.77
	TI-DIM	0.23	0.30	0.28	0.76	1.49	0.49
	SI-DIM	0.71	0.71	0.75	2.73	3.89	1.37
	CD-AP	3.77	6.48	7.09	63.72	76.79	49.67
	TTP	27.99	26.08	24.61	72.47	74.51	70.96
	GAP	5.72	4.51	7.33	71.04	**83.64**	52.07
	Ours	**31.10**	**30.07**	**27.70**	**77.13**	80.55	**76.78**
VGG16	MIM	0.14	0.15	0.16	0.40	0.34	0.19
	TI-MIM	0.26	0.24	0.20	0.45	0.57	0.28
	SI-MIM	0.28	0.20	0.21	0.49	0.25	0.14
	DIM	0.22	0.16	0.27	0.93	0.99	0.49
	TI-DIM	0.14	0.19	0.21	0.35	0.34	0.21
	SI-DIM	0.50	0.36	0.33	0.80	0.69	0.26
	CD-AP	0.36	0.34	0.35	4.63	10.20	3.60
	TTP	3.75	3.20	**2.66**	27.80	32.70	16.57
	GAP	0.30	0.52	0.42	4.52	8.92	3.35
	Ours	**4.14**	**3.22**	**2.66**	**30.16**	**38.10**	**17.95**

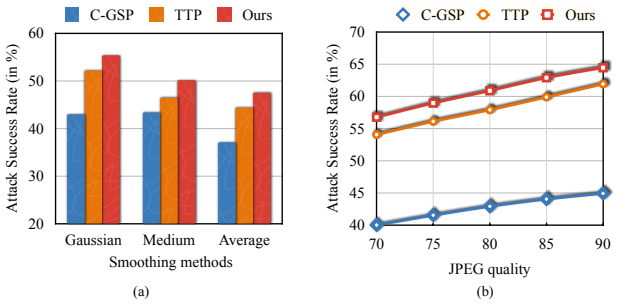

Figure 5. Attack success rates of targeted adversarial examples generated by various generative attack methods against different input process defense methods. Figure (a) displays the results against various input smooth methods (including gaussian, medium and average smoothing). Figure (b) shows the results against JPEG compression (the JPEG quality factor varies from 70 to 90). Here, the substitute model is Res-152 and the target model is Vgg-16.

instance-agnostic attacks include CD-AP [40], TTP [39], GAP [41] and C-GSP [55]. In our experiments, the perturbation size $\varepsilon = 16/255$, the decay factor μ_0 is 1 in MIM, the transform probability is 0.7 in DIM, and the kernel size is 15 in TIM. And other hyper-parameters follow the default settings provided in their original works.

4.2. Comparison with State-of-the-art Methods

Attack Normally Trained Models. To evaluate the performance of our method, we first perform targeted adversarial attacks to compare the transferability across normally trained models of adversarial examples generated by various methods, including 6 iterative instance-specific attack methods and 5 instance-agnostic attack methods. As shown in Table 1, our dynamic generative attack method reaches the best transferability on 16 out of 18 various black-box cases. Furthermore, the attack success rate of our method against normally trained models is 65.42% on average with the substitute model as Res-152, which outperforms the best of baselines [39] by 5.28% on average. Another interesting discovery is that iterative instance-specific attack methods can reach better performance under the setting of white-box attacks, while generative attack methods show more transferability when adapting the adversarial examples to a black-box model. Among these generative methods, our method can still generate targeted adversarial examples with higher transferability than others.

Attack Models with Robust Training Mechanisms. To comprehensively verify the effectiveness of our method, we compare the transferability of our approach with baselines against several robust models, which are trained with various robust training strategies. As presented in Table 2, although the attack success rate against these robust models is relatively low, our method is still able to outperform baseline methods. Our generative attack method achieves the best performance of targeted transferable adversarial attacks on 11 over 12 cases. And regarding Res-152 as the substitute model, our method outperforms the best of baselines [39] by 4.45% on average. Compared with normally trained models, models with adversarial or robust training strategies are difficult to attack successfully, which indicates they are more robust than normal models.

Attack Models with Input Process Defense. Another widely used and simple defense method is *Input Process*, which leverages image processing techniques to remove

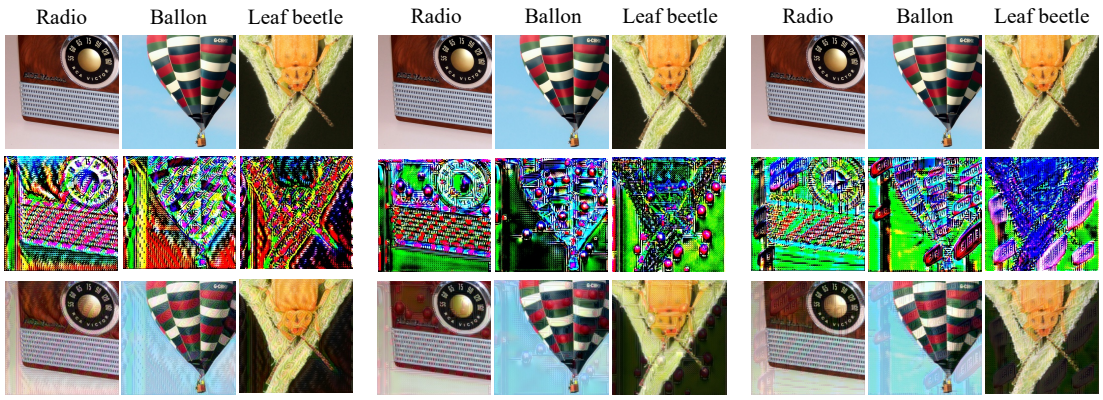

Figure 6. Visualization results of adversarial examples generated by our proposed method. The 1st, 2nd and 3rd rows show clean images, adversarial perturbations and adversarial examples, respectively. And (a), (b) and (c) display the results of different target classes (French bulldog, Fire engine and Street sign), which indicate that different target classes lead to different patterns or texture styles of the generated perturbations and adversarial images. Please refer to our **Supplementary Material** for more qualitative examples.

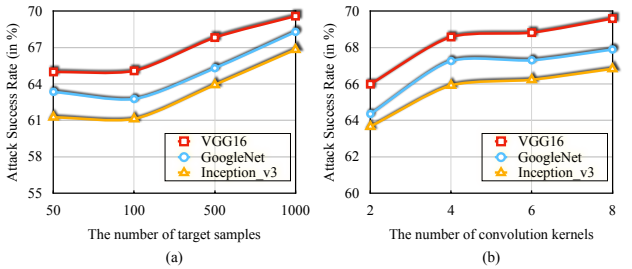

Figure 7. Ablation studies. (a) Effect of the number of target samples. (b) Effect of the number of learnable convolution kernels. Note that the substitute model is Res-152, while victim models include Vgg-16, GoogleNet and Inc-v3.

the adversary perturbation before feeding into target models. To further understand our method, we also evaluate the targeted adversarial examples crafted by various generative attacks against several image processing based defense methods, namely, JPEG compression [12] and Smooth [8]. As presented in Figure 5, our method consistently outperforms the two existing generative attacks against both JPEG compression and various smoothing defense methods. For example, our method achieves an attack success rate of 54.77%, when the JPEG quality is 70, while TTP [39] and C-GSP [55] just reach 54.06% and 40.04%, respectively.

4.3. Visualization and Ablation Studies

Visualization. To vividly demonstrate the working mechanism of our method, we visualize several adversarial examples and perturbations with different target classes. As shown in Figure 6, there is an underlying dependency between the generated perturbations and input instances, where the perturbations are mainly concentrated on the semantical part of the input images. Besides, for different target classes, our method tends to generate perturbations with different texture patterns, which also verifies the effectiveness of our designed pattern injection module.

Ablation Studies. To look deeper into our proposed method, in this section, we present a series of ablation studies. As shown in Figure 7, we vary the number of target samples and convolution kernels to verify the effectiveness of our method. As presented in Figure 7(a), the performance of our method continues to grow until the number of target samples reaches 1000. Besides, in Figure 7(b), when the number of kernels is greater than 4, our performance reaches a plateau, which means that it is sufficient to reach an impressive attack success rate by learning 4 kernels in each designed dynamic convolution module.

5. Conclusion

In this paper, we first construct a causal graph to expose the origin of targeted adversarial examples, which motivates us to inject target patterns or styles for generating transferable target adversarial examples. Then we introduce a dynamic generative attack model composed of a cross-attention guided convolution module and a pattern injection module. Our generative attack model can not only inject pattern or style information of the target class to improve transferable targeted attacks, but also learn specialized convolutional kernels for each input instance, which inherits the advantages of both instance-specific and instance-agnostic attacks. Moreover, we also provide rigorous theoretical analysis to guarantee the effectiveness of our method, and extensive experiments demonstrate that our method performs better than state-of-the-art targeted attack methods.

6. Acknowledgement

This work was partially supported by the National Nature Science Foundation of China (Grant 62022078, Grant 62021002), and National Defense Basic Scientific Research Program (Grant JCKY2022911B002).

References

[1] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018. 1

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020. 1

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International Conference on Machine Learning*, pages 284–293. PMLR, 2018. 1, 2

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017. 1

[5] Zhaohui Che, Ali Borji, Guangtao Zhai, Suiyi Ling, Jing Li, Xiongkuo Min, Guodong Guo, and Patrick Le Callet. Smgea: A new ensemble adversarial attack powered by long-term gradient memories. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. 3

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 6

[8] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019. 8

[9] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018. 1, 2, 3, 6

[10] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 3, 6

[11] Jiawei Du, Hu Zhang, Joey Tianyi Zhou, Yi Yang, and Jiashi Feng. Query-efficient meta attack to deep neural networks. In *International Conference on Learning Representations*, 2020. 2

[12] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. A study of the effect of jpg compression on adversarial images, 2016. 8

[13] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 2

[14] Lianli Gao, Yaya Cheng, Qilong Zhang, Xing Xu, and Jingkuan Song. Feature space targeted attacks by statistic alignment. *arXiv preprint arXiv:2105.11645*, 2021. 2

[15] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018. 6, 7

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1, 3

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

[18] Jie Hang, Keji Han, Hui Chen, and Yun Li. Ensemble adversarial black-box attacks against deep learning systems. *Pattern Recognition*, 101:107184, 2020. 3, 6

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 6

[20] Ziwen He, Wei Wang, Jing Dong, and Tieniu Tan. Revisiting ensemble adversarial attack. *Signal Processing: Image Communication*, page 116747, 2022. 3

[21] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2019. 6, 7

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 6

[23] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 5, 6

[24] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 3, 5

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 4, 5

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 6

[27] 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. 1, 2, 3

[28] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1

[29] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 1, 2, 7

[30] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, Alan Yuille, Sangxia Huang, Yao Zhao, Yuzhe Zhao, Zhonglin Han, Junjiajia Long, Yerkebulan Berdibekov, Takuya Akiba, Seiya Tokui, and Motoki Abe. Adversarial attacks and defences competition, 2018. 6

[31] Qizhang Li, Yiwen Guo, and Hao Chen. Yet another intermediate-level attack. In *European Conference on Computer Vision*, pages 241–257. Springer, 2020. 2

[32] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2019. 6

[33] Aishan Liu, Jiakai Wang, Xianglong Liu, Bowen Cao, Chongzhi Zhang, and Hang Yu. Bias-based universal adversarial patch attack for automatic check-out, 2020. 1

[34] Yantao Lu, Yunhan Jia, Jianyu Wang, Bai Li, Weiheng Chai, Lawrence Carin, and Senem Velipasalar. Enhancing cross-task black-box transferability of adversarial examples with dispersion reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 940–949, 2020. 2

[35] Jinqi Luo, Tao Bai, Jun Zhao, and Bo Li. Generating adversarial yet inconspicuous patches with a single image. 2020. 1, 2, 3

[36] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. 1

[37] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017. 2, 3

[38] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016. 1

[39] 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, 3, 5, 7, 8

[40] Muhammad Muzammal Naseer, Salman H Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. *Advances in Neural Information Processing Systems*, 32:12905–12915, 2019. 2, 3, 7

[41] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. 3, 7

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 1

[43] Suproteem K. Sarkar, Kojin Oshiba, Daniel Giebisch, and Yaron Singer. Robust classification of financial risk, 2018. 1

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 1, 6

[45] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI Conference on Artificial Intelligence*, 2017. 6

[46] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 6

[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 6

[48] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *Computer Science*, 2013. 1, 2

[49] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 7

[50] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. *arXiv preprint arXiv:2103.15571*, 2021. 2

[51] Xiaosen Wang, Kun He, and John E Hopcroft. At-gan: A generative attack model for adversarial transferring on generative adversarial nets. *arXiv preprint arXiv:1904.07793*, 3(4), 2019. 3

[52] 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, 3

[53] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 2, 3, 6

[54] Yifeng Xiong, Jiadong Lin, Min Zhang, John E Hopcroft, and Kun He. Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14983–14992, 2022. 2

[55] Xiao Yang, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Boosting transferability of targeted adversarial examples via hierarchical generative networks. *arXiv preprint arXiv:2107.01809*, 2021. 2, 3, 6, 7, 8

[56] Yonggang Zhang, Mingming Gong, Tongliang Liu, Gang Niu, Xinmei Tian, Bo Han, Bernhard Schölkopf, and Kun Zhang. Adversarial robustness through the lens of causality. In *International Conference on Learning Representations*, 2021. 3

[57] Zheng-An Zhu, Yun-Zhong Lu, and Chen-Kuo Chiang. Generating adversarial examples by makeup attacks on face recognition. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2516–2520. IEEE, 2019. 3