Robustness and Generalization via Generative Adversarial Training

Omid Poursaeed[1,2] Tianxing Jiang[1] Harry Yang[3]

Serge Belongie[1,2] Ser-Nam Lim[3]

[1]Cornell University [2]Cornell Tech [3]Facebook AI

Abstract

While deep neural networks have achieved remarkable success in various computer vision tasks, they often fail to generalize to new domains and subtle variations of input images. Several defenses have been proposed to improve the robustness against these variations. However, current defenses can only withstand the specific attack used in training, and the models often remain vulnerable to other input variations. Moreover, these methods often degrade performance of the model on clean images and do not generalize to out-of-domain samples. In this paper we present Generative Adversarial Training, an approach to simultaneously improve the model's generalization to the test set and out-of-domain samples as well as its robustness to unseen adversarial attacks. Instead of altering a low-level pre-defined aspect of images, we generate a spectrum of low-level, mid-level and high-level changes using generative models with a disentangled latent space. Adversarial training with these examples enable the model to withstand a wide range of attacks by observing a variety of input alterations during training. We show that our approach not only improves performance of the model on clean images and out-of-domain samples but also makes it robust against unforeseen attacks and outperforms prior work. We validate effectiveness of our method by demonstrating results on various tasks such as classification, segmentation and object detection.

1. Introduction

Deep neural networks have shown promising generalization to in-domain samples. However, they are vulnerable to slight alterations of input images and have limited generalization to new domains. Several defenses have been proposed for improving the models' robustness against input variations. However, these models only provide robustness against a narrow range of threat models used in training, and they have poor generalization to unseen attacks. We hypothesize that this is due to the fact that they only consider a small subset of realistic examples on or near the manifold of natural images. For instance, additive perturbations leave high-level semantic aspects of images intact. Therefore, models trained against these examples do not provide robustness to high-level input variations. In order for a model to become robust to all realistic variations of inputs, it needs to see a diverse set of samples during training. However, most of the existing works only alter a low-level aspect of images such as color [18, 24, 4], spatial [13, 36, 1], pose [2, 42], and others. Even if we consider the union of several types of attacks, the model can still be vulnerable to other input variations that are not contained in any of the constituent threat models. To provide robustness against unforeseen attacks, we propose adversarial training against a range of low-level to high-level variations of inputs. We leverage generative models with disentangled latent representations to systematically build diverse and realistic examples without leaving the manifold of natural images. We show that our approach improves generalization and robustness of the model to unseen variations of input images *without* training against any of them.

We build upon state-of-the-art generative models which disentangle factors of variation in images. We create fine and coarse-grained adversarial changes by manipulating various latent variables at different resolutions. Loss of the target network is used to guide the generation process. The pre-trained generative model constrains the search space for our adversarial examples to realistic images, thereby revealing the target model's vulnerability in the natural image space. We verify that we do not deviate from the space of realistic images with a user study as well as a t-SNE plot comparing distributions of real and adversarial images. As a result, we observe that including these examples in training the model enhances its accuracy on clean images as well as out-of-domain samples. Moreover, since the model has seen a variety of low-level and high-level alterations of images, it becomes robust to a wide range of adversarial examples including recoloring, spatial transformations, perceptual and additive perturbations.

Our contributions can be summarized as follows:

- We present *Generative Adversarial Training (GAT)* to simultaneously improve the model's generalization to clean and out-of-domain samples and its robustness to *unforeseen* attacks. Our approach is based on adversarial training with fine-grained unrestricted adversarial examples in which the attacker controls which aspects of the image to manipulate, resulting in a diverse set of realistic, on-the-manifold examples.

- We evaluate GAT against a diverse set of adversarial attacks: recoloring, spatial transformations, perceptual and additive perturbations, and demonstrate that it achieves state-of-the-art robustness against these attacks *without* training against any of them.

- We extend our approach to semantic segmentation and object detection tasks, and propose the first method for generating unrestricted adversarial examples for segmentation and detection. Training with our examples improves both robustness and generalization of the model.

2. Related Work

2.1. Adversarial Examples

Most of the existing works on adversarial attacks focus on norm-constrained adversarial examples: for a given classifier $F : \mathbb{R}^n \rightarrow \{1, \ldots, K\}$ and an image $x \in \mathbb{R}^n$, the adversarial image $x' \in \mathbb{R}^n$ is created such that $\|x - x'\|_p < \epsilon$ and $F(x) \neq F(x')$. Common values for p are $0, 2, \infty$, and ϵ is chosen small enough so that the perturbation is imperceptible. Various algorithms have been proposed for creating x' from x. Optimization-based methods solve a surrogate optimization problem based on the classifier's loss and the perturbation norm [34, 14, 8]. Gradient-based methods use gradient of the classifier's loss with respect to the input image [15, 27, 11, 23].

Another line of work creates unrestricted adversarial examples that are not bounded by a norm threshold. One way to achieve this is by applying subtle geometric transformations such as spatial transformations [36, 1], translations and rotations [13] or pose changes [2] to the inputs. Other works consider recoloring [18, 24, 4], intermediate features [12, 25, 40] and inserting new objects or patches in the image [6]. A challenge for creating unrestricted adversarial examples and defending against them is introduced in [5] using the simple task of classifying between birds and bicycles. Recent works consider using generative models to create adversarial examples. Song et al. [31] search in the latent (z) space of AC-GAN [28] to find generated images that can fool a target classifier but yield correct predictions on AC-GAN's auxiliary classifier. They constrain the search region of z so that it is close to a randomly sampled noise vector, and show results on MNIST, SVHN and CelebA datasets. This approach does not alter any aspects of images and merely generates a set of generated images misclassified by the model. As we show in the supplementary material, training with these adversarial examples hurts the classifier's performance on clean images. One reason for this accuracy drop is that requiring two classifiers to have inconsistent predictions degrades sample quality of the model. To further illustrate difference of our approach with [31], we plot t-SNE embeddings of real CelebA-HQ images and adversarial examples from our method and [31] in the supplementary material, and show that our adversarial images stay closer to the manifold of real images. The recent work by [16] shows that adversarial training with examples generated by StyleGAN can improve performance of the model on clean images. Their approach requires precomputing a mapping from the image space to the latent space for the whole dataset, which is computationally prohibitive for large datasets. It is also constrained to fine changes using only a subset of the latent variables. They argue that coarse changes might be label-dependent. This statement is not true since coarse stylistic changes do not alter the label (e.g. gender) and merely modify high-level aspects of images. Moreover, [16] only considers the classification task on low-resolution datasets such as ColorMNIST (28x28) and CelebA (64x64). While their approach uses the StyleGAN model, they do not show any results on high-resolution datasets that StyleGAN is originally trained on (e.g. LSUN and CelebA-HQ (1024x1024)), which makes it hard to ascertain that their adversarial training will be effective on high-resolution datasets. Even on low-resolution datasets such as Color-MNIST, their adversarial training can perform worse than random sampling on unbiased datasets as shown in Table 2 of [16]. On the other hand, our approach directly samples and manipulates latent variables without requiring the mapping step. We demonstrate that by limiting the number of iterations we can use both coarse and fine changes and both contribute to improvements in performance. While [16] only enhances the accuracy on clean images, our approach also improves generalization to out-of-domains samples and unseen adversarial examples. In addition, we propose the first method for unrestricted adversarial attacks on semantic segmentation and object detection, and demonstrate that adversarial training improves segmentation and detection results on clean images.

2.2. Adversarial Robustness

Several methods have been proposed for defending against adversarial attacks. Many defenses attempt to combat adversaries using a form of input pre-processing or by manipulating intermediate features or gradients [17, 38, 30]. Few approaches have been able to scale up to high-resolution datasets such as ImageNet [26, 39, 20]. Most of the proposed heuristic defenses were later broken by stronger adversaries [8, 35, 3]. One of the most successful defenses is *adversarial training* [15, 23, 27, 37, 32, 16] which augments training data with adversarial examples generated as the training progresses. This approach is able to withstand strong attacks. Adversarial training with perturbation-based examples degrades performance of the model on clean images. [37] proposes to use separate batch norm layers for clean and adversarial images to avoid the accuracy drop. Our approach improves the model's accuracy on clean images without modifying the architecture. Moreover, unlike prior work which make the classifier robust only against the specific attack used in training, our method provides generalizable robustness across a range of attacks.

3. Approach

We present *Generative Adversarial Training*, a method for improving generalization and robustness of models to unseen attacks. Most of the existing works on adversarial attacks modify a low-level aspect of images, thereby a model adversarially trained with these samples remains susceptible to various other input alterations. We demonstrate that by encouraging diversity and realism in the generated adversarial examples, we can improve both performance of the model on clean and out-of-domain images and its robustness to a wide range of adversarial attacks. Our approach creates a spectrum of low-level, mid-level and high-level changes for which the target network fails to generalize. The adversarially trained model observes a variety of examples on or near the manifold of natural images. This allows the model to generalize better to test samples and various alterations of images. We leverage disentangled latent representations for generating the adversarial examples. We build upon state-of-the-art generative models and use Style-GAN [22] for the classification task and SPADE [29] for semantic segmentation and object detection.

3.1. Classification

Style-GAN [22] is a state-of-the-art generative model which disentangles high-level attributes and stochastic variations in an unsupervised manner. Stylistic variations are represented by *style* variables and stochastic details are captured by *noise* variables. Changing the noise only affects low-level details, leaving the overall composition and high-level aspects intact. This allows us to manipulate the noise variables such that variations are barely noticeable by the human eye. The style variables affect higher level aspects of image generation. For instance, when the model is trained on bedrooms, style variables from the top layers control viewpoint of the camera, middle layers select the particular furniture, and bottom layers deal with colors and details of materials [22]. This allows us to manipulate images in a controlled manner, providing an avenue for fine-grained unrestricted attacks.

Formally, we can represent Style-GAN with a mapping function f and a synthesis network g. As illustrated in Figure 1, the mapping function is an 8-layer MLP which takes a latent code $\mathbf{z}$, and produces an intermediate latent vector $\mathbf{w} = f(\mathbf{z})$. This vector is then specialized by learned affine transformations A to style variables $\mathbf{y}$, which control adaptive instance normalization operations after each convolutional layer of the synthesis network g. Noise inputs are single-channel images consisting of un-correlated Gaussian noise that are fed to each layer of the synthesis network. Learned per-feature scaling factors B are used to generate noise variables $\boldsymbol{\eta}$ which are added to the output of convolutional layers. The synthesis network takes style $\mathbf{y}$ and noise $\boldsymbol{\eta}$ as input, and generates an image $\mathbf{x} = g(\mathbf{y}, \boldsymbol{\eta})$. We pass the generated image to a pre-trained classifier F. We seek to slightly modify $\mathbf{x}$ so that F can no longer classify it correctly. We achieve this through perturbing the style and noise tensors. We initialize adversarial style and noise variables as $\mathbf{y}^{(\mathbf{0})}_{\mathbf{adv}} = \mathbf{y}$ and $\boldsymbol{\eta}^{(\mathbf{0})}_{\mathbf{adv}} = \boldsymbol{\eta}$, and iteratively update them in order to fool the classifier. Loss of the classifier determines the update rule, which in turn depends on the type of attack. As common in the literature, we consider two types of attacks: non-targeted and targeted.

In order to generate non-targeted adversarial examples, we need to change the model's original prediction. Starting from initial values $\mathbf{y}^{(\mathbf{0})}_{\mathbf{adv}} = \mathbf{y}$ and $\boldsymbol{\eta}^{(\mathbf{0})}_{\mathbf{adv}} = \boldsymbol{\eta}$, we can iteratively perform gradient ascent in the style and noise spaces of the generator to find values that maximize the classifier's loss. Alternatively, as proposed by [23], we can use the least-likely predicted class $ll_{\mathbf{x}} = \arg\min(F(\mathbf{x}))$ as our target. We found this approach more effective in practice. At time step t, the update rule for style and noise variables is:

$$\mathbf{y}^{(\mathbf{t+1})}_{\mathbf{adv}} = \mathbf{y}^{(\mathbf{t})}_{\mathbf{adv}} - \epsilon \cdot \mathrm{sign}(\nabla_{\mathbf{y}^{(\mathbf{t})}_{\mathbf{adv}}} J(F(g(\mathbf{y}^{(\mathbf{t})}_{\mathbf{adv}}, \boldsymbol{\eta}^{(\mathbf{t})}_{\mathbf{adv}})), ll_{\mathbf{x}})) \tag{1}$$

$$\boldsymbol{\eta}^{(\mathbf{t+1})}_{\mathbf{adv}} = \boldsymbol{\eta}^{(\mathbf{t})}_{\mathbf{adv}} - \delta \cdot \mathrm{sign}(\nabla_{\boldsymbol{\eta}^{(\mathbf{t})}_{\mathbf{adv}}} J(F(g(\mathbf{y}^{(\mathbf{t})}_{\mathbf{adv}}, \boldsymbol{\eta}^{(\mathbf{t})}_{\mathbf{adv}})), ll_{\mathbf{x}})) \tag{2}$$

in which $J(\cdot, \cdot)$ is the classifier's loss function, $F(\cdot)$ gives the probability distribution over classes, $\mathbf{x} = g(\mathbf{y}, \boldsymbol{\eta})$, and $\epsilon, \delta \in \mathbb{R}$ are step sizes. We use $(\epsilon, \delta) = (0.004, 0.2)$ and $(0.004, 0.1)$ for LSUN and CelebA-HQ respectively. We perform multiple steps of gradient descent (usually 2 to 10) until the classifier is fooled. We report the average number of iterations required to fool the classifier in the supplementary material.

Generating targeted adversarial examples is more challenging as we need to change the prediction to a specific class T. In this case, we perform gradient descent to mini-

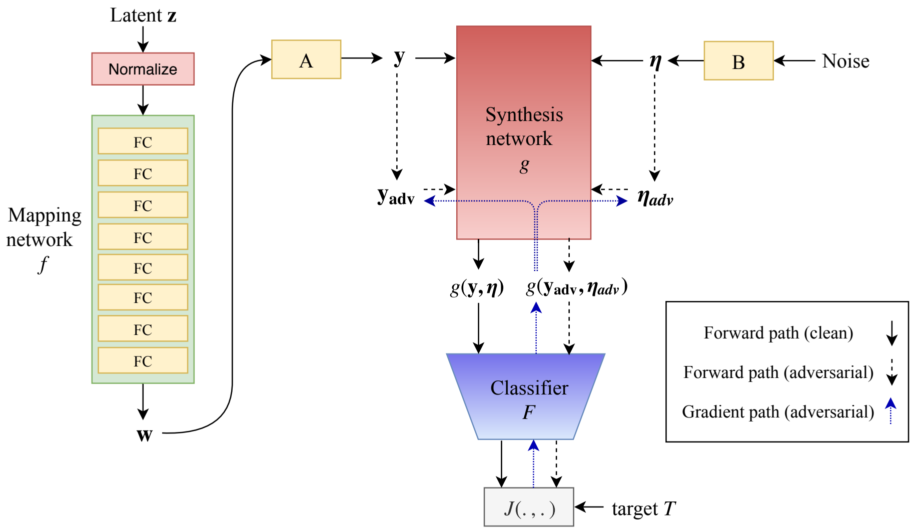


Figure 1: Classification architecture. Style $(\mathbf{y})$ and noise $(\boldsymbol{\eta})$ variables are used to generate images $g(\mathbf{y}, \boldsymbol{\eta})$ which are fed to the classifier F. Adversarial style and noise tensors are initialized with $\mathbf{y}$ and $\boldsymbol{\eta}$ and iteratively updated using gradients of the loss function J. The classifier F is adversarially trained with clean and adversarial samples.

mize the classifier's loss with respect to the target T:

$$\mathbf{y}_{\mathbf{adv}}^{(\mathbf{t+1})} = \mathbf{y}_{\mathbf{adv}}^{(\mathbf{t})} - \epsilon \cdot \mathrm{sign}(\nabla_{\mathbf{y}_{\mathbf{adv}}^{(\mathbf{t})}} J(F(g(\mathbf{y}_{\mathbf{adv}}^{(\mathbf{t})}, \boldsymbol{\eta}_{\mathbf{adv}}^{(\mathbf{t})})), T)) \tag{3}$$

$$\boldsymbol{\eta}_{\mathbf{adv}}^{(\mathbf{t+1})} = \boldsymbol{\eta}_{\mathbf{adv}}^{(\mathbf{t})} - \delta \cdot \mathrm{sign}(\nabla_{\boldsymbol{\eta}_{\mathbf{adv}}^{(\mathbf{t})}} J(F(g(\mathbf{y}_{\mathbf{adv}}^{(\mathbf{t})}, \boldsymbol{\eta}_{\mathbf{adv}}^{(\mathbf{t})})), T)) \tag{4}$$

We use $(\epsilon, \delta) = (0.005, 0.2)$ and $(0.004, 0.1)$ in the experiments on LSUN and CelebA-HQ respectively. Note that we only control deviation from the initial latent variables, and do not impose any norm constraint on generated images.

The classifier F is adversarially trained with equal number of clean and adversarial images. To maximize diversity of generated samples, we manipulate groups of consecutive style and noise layers separately (e.g. layers 1-2, 3-4, etc.) for experiments on adversarial training. As the training progresses, the classifier observes a variety of examples and becomes more robust to input variations. Hence, the generator needs to explore new areas of the natural image manifold that correspond to generalization errors of the classifier. Since our model allows both coarse and fine changes, it results in superior generalization performance compared to existing works on adversarial training that only manipulate a low-level aspect of images.

3.1.1 Input-conditioned Generation

Generation can also be conditioned on real input images by embedding them into the latent space of Style-GAN. We first synthesize images similar to the given input image I by optimizing values of $\mathbf{y}$ and $\boldsymbol{\eta}$ such that $g(\mathbf{y}, \boldsymbol{\eta})$ is close to I. More specifically, we minimize the perceptual distance [19] between $g(\mathbf{y}, \boldsymbol{\eta})$ and I. We can then proceed similar to equations 1–4 to perturb these tensors and generate the adversarial image. Realism of synthesized images depends on inference properties of the generative model. In practice, generated images resemble input images with high fidelity especially for CelebA-HQ images.

3.2. Semantic Segmentation and Object Detection

We also consider the task of semantic segmentation and leverage the generative model proposed by [29]. The model is conditioned on input semantic layouts and uses SPatially-Adaptive (DE)normalization (SPADE) modules to better preserve semantic information against common normalization layers. The layout is first projected onto an embedding space and then convolved to produce the modulation parameters γ and β. We adversarially modify these parameters with the goal of fooling a segmentation model. We consider non-targeted attacks using per-pixel predictions and compute gradient of the loss function with respect to the modulation parameters with an update rule similar to equations 1 and 2. Figure 2 illustrates the architecture. We consider a similar architecture for the object detection task except that we pass the generated image to the detection model and try to increase its loss. Results for this task are shown in the supplementary material.

4. Results and Discussion

We provide qualitative and quantitative results using experiments on LSUN [41] and CelebA-HQ [21]. LSUN contains 10 scene categories and 20 object categories. We use all the scene classes as well as two object classes: *cars* and

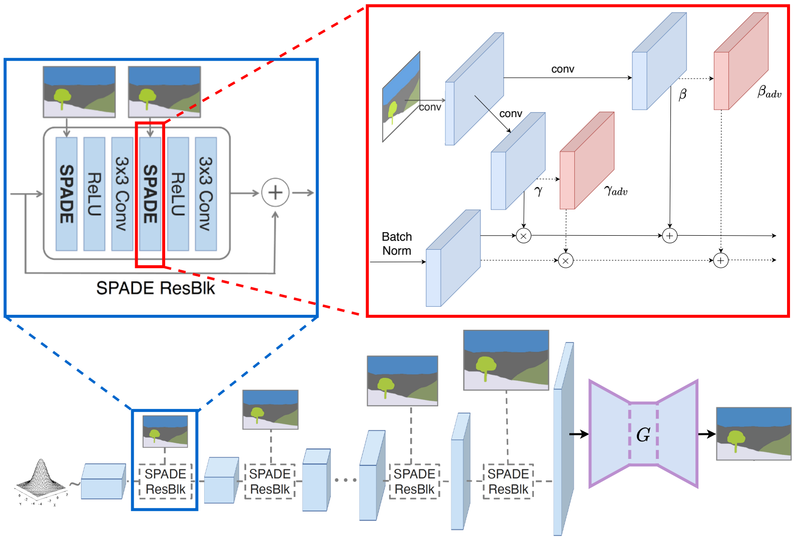


Figure 2: Semantic segmentation and object detection architecture. Adversarial parameters γ_{adv} and β_{adv} are initialized with γ and β, and iteratively updated to fool the segmentation or detection model G. The model G is adversarially trained with clean and adversarial examples.

cats. We consider this dataset since it is used in Style-GAN, and is well suited for a classification task. For the scene categories, a 10-way classifier is trained based on Inception-v3 [33] which achieves an accuracy of 88.9% on LSUN's test set. The two object classes also appear in ImageNet [10], a richer dataset containing 1000 categories. Therefore, for experiments on cars and cats we use an Inception-v3 model trained on ImageNet. This allows us to explore a broader set of categories in our attacks, and is particularly helpful for targeted adversarial examples. CelebA-HQ consists of 30,000 face images at 1024×1024 resolution. We consider the gender classification task, and use the classifier provided by [22]. This is a binary task for which targeted and non-targeted attacks are similar.

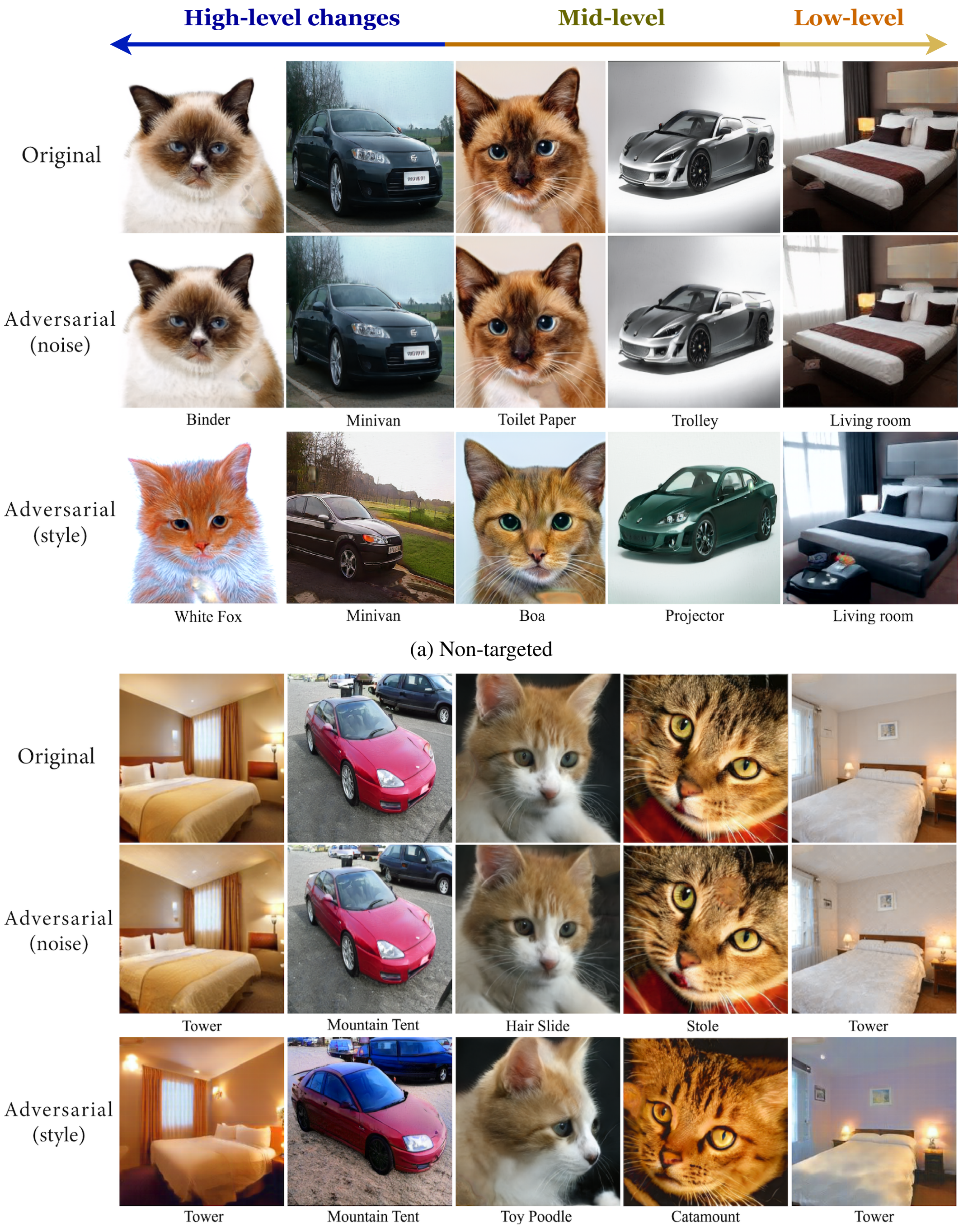
In order to synthesize a variety of adversarial examples, we use different random seeds in Style-GAN to obtain various values for $\mathbf{z}, \mathbf{w}, \mathbf{y}$ and $\boldsymbol{\eta}$. Style-based adversarial examples are generated by initializing $\mathbf{y_{adv}}$ with the value of $\mathbf{y}$, and iteratively updating it as in equation 1 (or 3) until the resulting image $g(\mathbf{y_{adv}}, \boldsymbol{\eta})$ fools the classifier F. Noise-based adversarial examples are created similarly using $\boldsymbol{\eta}_{\mathbf{adv}}$ and the update rule in equation 2 (or 4). While using different step sizes makes a fair comparison difficult, we generally found it easier to fool the model by manipulating the noise variables. We can also combine the effect of style and noise by simultaneously updating $\mathbf{y_{adv}}$ and $\boldsymbol{\eta}_{\mathbf{adv}}$ in each iteration, and feeding $g(\mathbf{y_{adv}}, \boldsymbol{\eta}_{\mathbf{adv}})$ to the classifier. In this case, the effect of style usually dominates since it creates coarser changes.

Figure 3 illustrates generated adversarial examples on LSUN. Original images $g(\mathbf{y}, \boldsymbol{\eta})$, noise-based images $g(\mathbf{y}, \boldsymbol{\eta}_{\mathbf{adv}})$ and style-based images $g(\mathbf{y_{adv}}, \boldsymbol{\eta})$ are shown. Adversarial images look almost indistinguishable from natural images. Manipulating the noise variable results in subtle, imperceptible changes. Varying the style leads to coarser changes such as different colorization, pose changes, and even removing or inserting objects in the scene. We can also control granularity of changes by selecting specific layers of the model. Manipulating top layers, corresponding to coarse spatial resolutions, results in high-level changes. Lower layers, on the other hand, modify finer details. In the first two columns of Figure 3, we only modify top 6 layers (out of 18) to generate adversarial images. The middle two columns change layers 7 to 12, and the last column uses the bottom 6 layers. We also show results on CelebA-HQ gender classification in the supplementary material. Figure 4 illustrates adversarial examples conditioned on real input images using the procedure described in Section 3.1.1. Synthesized images resemble inputs with high fidelity, and set the initial values in our optimization process.

We also show results on semantic segmentation in Figure 5 in which we consider non-targeted attacks on DeepLab-v2 [9] with a generator trained on the COCO-stuff dataset [7]. We iteratively modify modulation parameters at all layers, using a step size of 0.001, to maximize the segmentation loss with respect to the given label map. As we observe, subtle modifications to images lead to large drops in accuracy. Object detection results and additional examples are provided in the supplementary material.

4.1. Adversarial Training

Adversarial training increases robustness of models by injecting adversarial examples into training data. Adversar-



(a) Non-targeted

(b) Targeted

Figure 3: Unrestricted adversarial examples on LSUN for a) non-targeted and b) targeted attacks. Predicted classes are shown under each image.

ial training with norm-bounded examples degrades performance of the classifier on clean images as they have different underlying distributions. We show that adversarial training with our unrestricted examples *improves* the model's accuracy on clean test images as well as out-of-domain samples. To ensure that the model maximally benefits from these additional samples, we need to avoid unrealistic examples which do not resemble natural images. Therefore, we only include samples that can fool the model in less than a specific number of iterations. We use a threshold of 10 as the maximum number of iterations, and demonstrate results on classification, semantic segmentation and

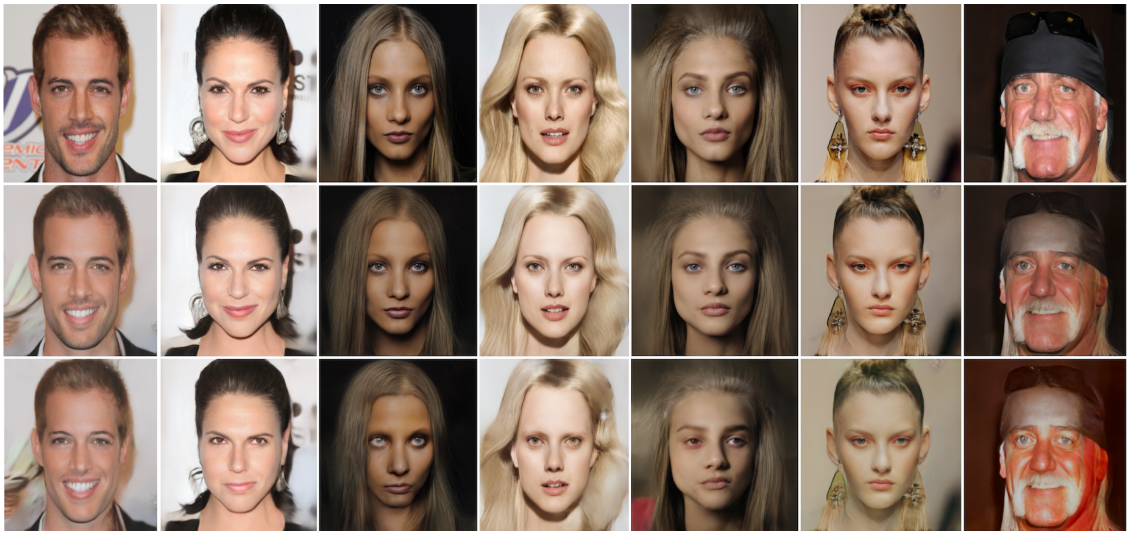


Figure 4: Input-conditioned adversarial examples on CelebA-HQ gender classification. From top to bottom: input, generated and style-based images. Males are classified as females and vice versa.

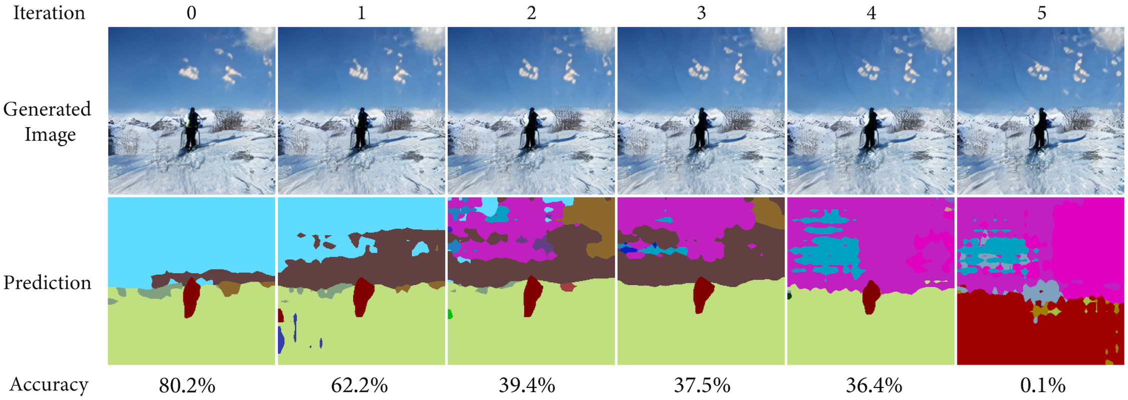


Figure 5: Unrestricted adversarial examples for semantic segmentation. Generated images, corresponding predictions and their accuracy (ratio of correctly predicted pixels) are shown for different number of iterations.

object detection[1]. We use the first 10 generated examples for each starting image in the segmentation and detection tasks. Table 1 shows accuracy of the strengthened and original classifiers on clean and adversarial test images. For the segmentation and detection tasks we report the mean accuracy and average precision of adversarial images at iteration 10. Similar to norm-constrained perturbations, adversarial training is an effective defense against our unrestricted attacks. Note that accuracy of the model on clean test images is improved after adversarial training. This is in contrast to training with norm-bounded adversarial inputs which hurts the classifier's performance on clean images, and it is due to the fact that unlike perturbation-based inputs, our generated images live on the manifold of realistic images as constrained by the generative model.

We also evaluate the adversarially trained model against various unforeseen attacks to demonstrate generalizable robustness of the model. We consider several attacks including recoloring [18, 24], spatial transformations [36], perceptual [25] and additive perturbations [27]. Results are shown in Table 2, and are compared against other defense methods such as Adversarial Training with PGD (AT PGD) [27], AT Spatial [36], AT Recolor [24], PAT [25] and AT AdvProp [37]. We observe that our adversarially trained model achieves superior robustness to these attacks. Unlike other methods which create a low-level change for each image, our approach generates a spectrum of low-level, mid-level and high-level changes around each sample, resulting in superior generalization to a variety of attacks without being trained against them. We also evaluate our attack against a certified defense in the supplementary material.

[1] In the supplementary material we show that limiting the number of iterations for norm-bounded perturbations is not effective for avoiding the accuracy drop on clean images.

	Classification (LSUN)		Classification (CelebA-HQ)		Segmentation		Detection	
	Clean	Adversarial	Clean	Adversarial	Clean	Adversarial	Clean	Adversarial
Adv. Trained	**89.5%**	78.4%	**96.2%**	83.6%	**69.1%**	60.2%	**40.2%**	33.7%
Original	88.9%	0.0%	95.7%	0.0%	67.9%	2.7%	39.0%	2.0%

Table 1: Performance of adversarially trained and original models on clean and adversarial test images. Accuracy is shown for classification and segmentation, and Average Precision is shown for object detection.

Model	Attack						Mean
	Clean	GAT	PGD	Spatial	Recolor	Perceptual	
GAT (Ours)	**89.5%**	78.4%	39.4%	47.8%	52.3%	28.9%	**42.1%**
AT PGD [27]	81.2%	6.3%	56.7%	5.1%	37.9%	2.8%	13.0%
AT AdvProp [37]	89.4%	7.8%	57.6%	6.0%	38.5%	3.5%	22.7%
AT Spatial [36]	76.3%	5.4%	3.1%	66.0%	4.1%	2.2%	3.7%
AT Recolor [24]	88.6%	4.7%	7.3%	0.4%	60.7%	1.7%	3.5%
PAT [25]	72.4%	18.3%	40.1%	46.3%	42.5%	30.1%	36.5%

Table 2: Accuracy of adversarially trained models against various attacks on the LSUN dataset. The mean accuracy of models on unseen attacks is shown in the last column.

	LSUN	ImageNet
Adv. Trained	**94.7%**	**92.0%**
Original	94.2%	91.4%

Table 3: Generalization of the models to in-domain (LSUN) and out-of-domain (ImageNet) samples.

Finally, we examine performance of the model on out-of-domain samples. We train a binary classifier on the two object classes of LSUN, i.e. cars and cats. We then evaluate the model on test samples from LSUN and ImageNet. Since several ImageNet classes represent cars and cats, we group all the relevant categories. Table 3 demonstrates the results for regular and adversarially trained models. We observe that adversarial training with our examples improves generalization power of the model to LSUN test images as well as ImageNet out-of-domain samples.

4.2. User Study

Norm-constrained attacks provide visual realism by L_p proximity to a real input. To verify that our unrestricted adversarial examples are realistic and correctly classified by an oracle, we perform human evaluation using Amazon Mechanical Turk. In the first experiment, each adversarial image is assigned to three workers, and their majority vote is considered as the label. The user interface for each worker contains nine images, and shows possible labels to choose from. We use 2400 noise-based and 2400 style-based adversarial images from the LSUN dataset, containing 200 samples from each class (10 scene classes and 2 object classes). The results indicate that 99.2% of workers' majority votes match the ground-truth labels. This number is 98.7% for style-based adversarial examples and 99.7% for noise-based ones. As we observe in Figure 3, noise-based examples do not deviate much from the original image, resulting in easier prediction by a human observer. On the other hand, style-based images show coarser changes, which in a few cases result in unrecognizable images or false predictions by the workers.

We use a similar setup in the second experiment but for classifying real versus fake (generated). We also include 2400 real images as well as 2400 unperturbed images generated by Style-GAN. 74.7% of unperturbed images are labeled by workers as real. This number is 74.3% for noise-based adversarial examples and 70.8% for style-based ones, indicating less than 4% drop compared with unperturbed images generated by Style-GAN.

5. Conclusion and Future Work

Existing works on adversarial defense assume a known threat model in advance. Therefore, adversaries can easily circumvent these defenses by using different types of attacks. This raises the need for defenses that are robust against unforeseen threat models. To this end, we incorporate diversity and realism in the examples used in adversarial training to bridge the distribution gap between real and adversarial examples. We leverage state-of-the-art generative models with disentangled representations which enable a range of low-level to high-level adversarial changes without leaving the manifold of natural images. We demonstrate results on classification, segmentation and object detection tasks. We consider extending the model to other tasks and evaluating it against new threat models in the future.

References

[1] Rima Alaifari, Giovanni S Alberti, and Tandri Gauksson. Adef: An iterative algorithm to construct adversarial deformations. *arXiv preprint arXiv:1804.07729*, 2018. 1, 2

[2] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. *arXiv preprint arXiv:1811.11553*, 2018. 1, 2

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018. 3

[4] 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. 1, 2

[5] Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018. 2

[6] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 2

[7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018. 5

[8] 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

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 5

[10] 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 conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[11] 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

[12] Isaac Dunn, Laura Hanu, Hadrien Pouget, Daniel Kroening, and Tom Melham. Evaluating robustness to context-sensitive feature perturbations of different granularities. *arXiv preprint arXiv:2001.11055*, 2020. 2

[13] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *arXiv preprint arXiv:1712.02779*, 2017. 1, 2

[14] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013. 2

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

[16] Sven Gowal, Chongli Qin, Po-Sen Huang, Taylan Cemgil, Krishnamurthy Dvijotham, Timothy Mann, and Pushmeet Kohli. Achieving robustness in the wild via adversarial mixing with disentangled representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1211–1220, 2020. 2, 3

[17] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017. 3

[18] Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1614–1619, 2018. 1, 2, 7

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 4

[20] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018. 3

[21] 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. 4

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018. 3, 5

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 2, 3

[24] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. *arXiv preprint arXiv:1906.00001*, 2019. 1, 2, 7, 8

[25] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020. 2, 7, 8

[26] 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

[27] 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, 3, 7, 8

[28] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017. 2

[29] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 3, 4

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

[31] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*, pages 8312–8323, 2018. 2

[32] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019. 3

[33] 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. 5

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[35] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018. 3

[36] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018. 1, 2, 7, 8

[37] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020. 3, 7, 8

[38] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017. 3

[39] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *arXiv preprint arXiv:1812.03411*, 2018. 3

[40] Qiuling Xu, Guanhong Tao, Siyuan Cheng, Lin Tan, and Xiangyu Zhang. Towards feature space adversarial attack. *arXiv preprint arXiv:2004.12385*, 2020. 2

[41] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 4

[42] Xiaohui Zeng, Chenxi Liu, Yu-Siang Wang, Weichao Qiu, Lingxi Xie, Yu-Wing Tai, Chi-Keung Tang, and Alan L Yuille. Adversarial attacks beyond the image space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4302–4311, 2019. 1