FROM ADVERSARIAL TRAINING TO GENERATIVE ADVERSARIAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we are interested in two seemingly different concepts: *adversarial training* and *generative adversarial networks (GANs)*. Particularly, how these techniques help to improve each other. To this end, we analyze the limitation of adversarial training as a defense method, starting from questioning how well the robustness of a model can generalize. Then, we successfully improve the generalizability via data augmentation by the "fake" images sampled from generative adversarial network. After that, we are surprised to see that the resulting robust classifier leads to a better generator, for free. We intuitively explain this interesting phenomenon and leave the theoretical analysis for future work. Motivated by these observations, we propose a system that combines generator, discriminator, and adversarial attacker together in a single network. After end-to-end training and fine tuning, our method can simultaneously improve the robustness of classifiers, measured by accuracy under strong adversarial attacks, and the quality of generators, evaluated both aesthetically and quantitatively. In terms of the classifier, we achieve better robustness than the state-of-the-art adversarial training algorithm proposed in (Madry *et al.*, 2017), while our generator achieves competitive performance compared with SN-GAN (Miyato and Koyama, 2018). Source code is publicly available online at `https://github.com/anonymous`.

1 INTRODUCTION

Deep neural networks have been very successful in modeling images, texts, and audios. Nonetheless, their characters have not yet been fully understood (Szegedy et al., 2013), leaving a big hole for malicious attack algorithms. In this paper, we start from adversarial attacks and defense but try to find the connection with Generative Adversarial Network (GAN) (Goodfellow et al., 2014a). Superficially, the difference between them is that the adversarial attack is the algorithm that finds a highly resembled image to cheat the classifier, whereas the GAN algorithm at its core is a generative model where the generator learns to convert white noise to images that look like authentic to the discriminator. We show in this paper that they are indeed closely related and can be used to strengthen each other: to accelerate and stabilize the GAN training cycle, the discriminator is expected to stay robust to adversarial examples; at the same time, a well trained generator provides a continuous support in probability space and thus improves the generalization ability of discriminator, even under adversarial attacks. That is the starting point of our idea to associate generative networks with robust classifiers.

Contributions: We find a novel way to make a connection between GAN and adversarial training. More importantly, we develop a system called AdvGAN to combine generator, discriminator, and adversarial attacker in the same network. Through the proposed "co-training" and "fine-tunning" steps, we are able to simultaneously improve the quality of generated images and the accuracy of discriminator under strong adversarial attacks. For example, when applying state-of-the-art adversarial training technique (Madry et al., 2017), the accuracy of ResNet18(+CIFAR10) drops from 81.5% to 29.6%; whereas the accuracy of our discriminator network drops from 81.1% to 36.4% (keeping all the hyperparameters and network structure unchanged). For the generator side, we are able to match or even beat the inception score of state-of-the-art method (Miyato & Koyama, 2018) on medium scale datasets (see Sec. 4 for details), with significantly fewer iterations. Lastly, we modify the loss of AC-GAN and our experiments confirm the superiority over the original one.

Notations Throughout this paper, we denote the (image, label) pair as (x_i, y_i), i is the index of data point; The classifier parameterized by weights w is $f(x; w)$, this function includes the final `Softmax` layer so the output is probabilities. We also define $D(x)$ and $G(z)$ as the discriminator and generator networks respectively. The adversarial example x_{adv} is crafted by perturbing the original input, i.e. $x_{\text{adv}} = x + \delta$, where $\|\delta\| \leq \delta_{\max}$. For convenience, we consider ℓ_∞-norm in our experiments. The real and fake images are denoted as $x_{\text{real/fake}}$, readers should differentiate the "fake" images with "adversarial" images[1]. The training set is denoted as $\mathcal{P}_{\text{real}}$, this is the empirical distribution. Given the training set $\mathcal{P}_{\text{real}}$, we define empirical loss function $\frac{1}{N_{\text{tr}}}\sum_{i=1}^{N_{\text{tr}}} \ell(f(x_i; w), y_i) = \mathbb{E}_{(x,y)\sim\mathcal{P}_{\text{real}}}\, \ell(f(x; w), y)$.

2 Related works

Generative adversarial network. This is a kind of algorithm that learns to model distribution either with or without supervision (Goodfellow et al., 2014a), which is often considered as a hard task especially for high dimensional data (images, texts, audios, etc.). In recent years, GANs keep to be intensively studied, toghther with other competitive generative models such as variational autoencoder or VAE, which learns the latent representation of data via prior knowledge (Kingma & Welling, 2013), and auto-regressive model that models the conditional distribution given previous states (e.g. PixelCNN (van den Oord et al., 2016)). One advantage of GANs over other methods is that they are able to generate high quality images directly from certain distributions, whereas the other methods are either slow in generation, or yield blurry images.

A GAN has two competing networks with different objectives: in the training phase, the generator $G(z)$ and the discriminator $D(x)$ are evolved in a minimax game, which can be denoted as a unified loss:

$$\min_G \max_D \Big\{ \mathbb{E}_{x\sim\mathcal{P}_{\text{real}}}\big[\log D(x)\big] + \mathbb{E}_{z\sim\mathcal{P}_z}\big[\log(1 - D(G(z))\big]\Big\}. \tag{1}$$

Unlike traditional machine learning problems where we typically minimize the loss, (1) is hard to optimize and that is the focus of recent literature. Among them, a guideline for the architectures of G and D is summarized in (Radford et al., 2015). Other training techniques, including feature matching (similar to MMD-GAN (Li et al., 2015; 2017)) and mini-batch discrimination are proposed in (Gulrajani et al., 2017a) to improve the stability and quality of networks. For high resolution and photo-realistic image generation, currently the standard way is to first learn to generate low resolution images as the intermediate products, and then learn to refine them progressively (Denton et al., 2015; Karras et al., 2017), this turns out to be more stable than directly generate high resolution images through a gigantic network. To reach the equilibrium efficiently, alternative loss metrics (Arjovsky & Bottou, 2017; Arjovsky et al., 2017; Berthelot et al., 2017; Gulrajani et al., 2017b; Unterthiner et al., 2017) are applied and proven to be effective. Among them, (Arjovsky & Bottou, 2017) theoretically explains why training the DCGAN is highly unstable — since the image manifold is highly concentrated towards a low dimensional manifold, and if two distributions $\mathbb{P}_{\text{real}}$ and $\mathbb{P}_{\text{fake}}$ are supported on two low dimensional manifolds that do not perfectly align, then there exists an "optimal discriminator $D(x)$" that tells apart two distributions with probability one. Moreover, under that situation, the gradient of discriminator $\nabla D(x)$ closes to zero and thus the training process is halted. Closely following that theorem, (Arjovsky et al., 2017) proposes to use Wasserstein-1 distance to measure the distance between real and fake data distribution. The resulting network, namely "Wasserstein-GAN", largely improves the stability of GAN training. Another noteworthy work inspired by WGAN/WGAN-GP is spectral normalization (Miyato et al., 2018), the idea is to estimate the operator norm $\sigma_{\max}(W)$ of weights W inside layers (convolution, linear, etc.), and then normalize these weights to have 1-operator norm through dividing weight tensors by operator norm: $\tilde{W} = W/\sigma_{\max}(W)$. Because ReLU non-linearity is already 1-Lipschitz, if we stack these layers together the network as a whole would still be 1-Lipschitz, that is exactly the prerequisite to apply Kantorovich-Rubinstein duality to estimate Wasserstein distance.

Despite the success of aforementioned works, we want to address one missing part of these models: **to the best of our knowledge, none of them consider the robustness of discrimination network** $D(x)$. This overlooked aspect can be problematic especially for high resolution images and large networks, this will be one of the central points of this paper.

[1] The fake images are generated by generator, while adversarial images are made by perturbing the natural images.

Adversarial attacks and defenses: Apart from GAN, another key ingredient of our method is adversarial examples, originated in (Szegedy et al., 2013) and further studied in (Goodfellow et al., 2014b). They found that machine learning models can be easily "fooled" by slightly modified images if we design a tiny perturbation according to some "attack" algorithms. In this paper we apply a simple yet efficient algorithm, namely PGD-attack (Madry et al., 2017), to generate adversarial examples. Given an example x with ground truth label y, PGD computes adversarial perturbation δ by solving the following optimization with Projected Gradient Descent:

$$\delta := \underset{\|\delta\|\le\delta_{\max}}{\arg\max}\,\ell\big(f(x+\delta;w),y\big), \tag{2}$$

where $f(\cdot;w)$ is the network parameterized by weights w, $\ell(\cdot,\cdot)$ is the loss function and for convenience we choose $\|\cdot\|$ to be the ℓ_∞-norm in accordance with (Madry et al., 2017; Athalye et al., 2018), but note that other norms are also applicable. Intuitively, the idea of (2) is to find the point $x_{\text{adv}} := x+\delta$ within an ℓ_∞-ball such that the loss value of x_{adv} is maximized, so that point is most likely to be an adversarial example. In fact, most optimization based attacking algorithms (e.g. FGSM (Goodfellow et al., 2014b), C&W (Carlini & Wagner, 2017)) shares the same idea as PGD attack.

Opposite to the adversarial attacks, the adversarial defenses are techniques that make models resistant to adversarial examples. It is worth noting that defense is a much harder task compared with attacks, especially for high dimensional data combined with complex models. Despite that huge amount of defense methods are proposed (Papernot et al., 2016; Madry et al., 2017; Buckman et al., 2018; Ma et al., 2018; Guo et al., 2018; Dhillon et al., 2018; Xie et al., 2018; Song et al., 2018; Samangouei et al., 2018), many of them rely on gradient masking or obfuscation, which provide an "illusion" of safety (Athalye et al., 2018; Athalye & Carlini, 2018). They claimed that the most effective defense algorithm is adversarial training (Madry et al., 2017), formulated as

$$\min_w \rho(w), \quad \text{where } \rho(w) := \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_{\text{real}}}\Big[\max_{\|\delta\|\le\delta_{\max}} \ell\big(f(x+\delta;w),y\big)\Big], \tag{3}$$

where $(x,y)\sim\mathcal{P}_{\text{real}}$ is the (image, label) joint distribution of real data, $f(x;w)$ is the network parameterized by w, $\ell\big(f(x;w),y\big)$ is the loss function of network (such as the cross-entropy loss). We remark that the data distribution $\mathcal{P}_{\text{real}}$ is often not available in practice, which will be replaced by the empirical distribution.

3 Proposed Approach

3.1 Motivation I: The generalization gap of adversarial training — how can GAN help?

In Sec. 2 we listed some of the published works on adversarial defense, and pointed out that adversarial training is the most effective method to date. However, until now this method has only been tested on small dataset like MNIST and CIFAR10 and it is an open problem as to whether it scales to large dataset such as ImageNet. To our knowledge, there are two significant drawbacks of this method that restrict its application. First and most obviously, the overhead to find adversarial examples in each iteration is about 10x of the normal process (this can be inferred by #Iterations in each PGD attack[2]). Moreover, we noticed that the trained model performs significantly worse on the test set than training set, i.e. the generalization gap is large under adversarial attacks (Fig. 1 (*Left*)). This indicates it is hard to find an adversarial example near the training data, yet much easier to find one close to testing data. We discuss the reason behind this huge generalization gap, and later we will alleviate this problem using GAN. From statistical learning theory, it is known that the generalization ability of model relies on the convergence of empirical risk to population risk, formally:

$$\sup_{h\in\mathcal{F}}\Big|\frac{1}{n}\sum_{i=1}^{n} h(X_i) - \mathbb{E}_X[h(X)]\Big| \xrightarrow{a.s.} 0, \text{ when } n\to\infty, \tag{4}$$

where $\mathcal{F}$ is the set of functions that are L-Lipschitz continuous, X can be any sub-Gaussian random variabl. Apart from that, to make our model robust to adversarial distortion, it is desirable to enforce a small local Lipschitz value (LLV) around data manifold $\mathcal{P}_{\text{real}}$. This idea includes many of the defense

[2] We refer readers to the source code in https://github.com/MadryLab/cifar10_challenge.

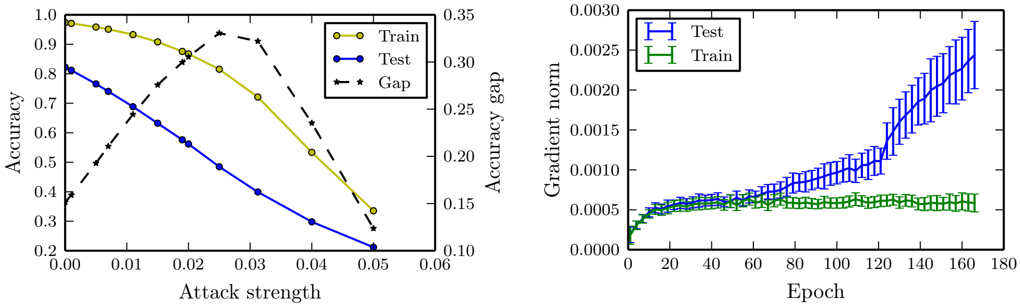

Figure 1: *Left*: Accuracy under different levels of attack. The model (VGG16) is obtained by adversarial training on CIFAR-10, we set $\delta_{\max} = 0.03125$ in (3). The horizontal axis is the attack strength δ which is equivalent to $\delta_{\max}$ in (2). Note that $\delta_{\max}$ in (2) and (3) have different meanings — one is for attack and the other is for defense. Notice the increasing accuracy gap when $\delta < 0.03125$. *Right*: The local Lipschitz value (LLV) measured by gradient norm $\|\frac{\partial}{\partial x_i}\ell\big(f(x_i; w), y_i\big)\|_2$, data pairs (x_i, y_i) are chosen from the training and testing set respectively. During the training process, LLV on the training set stabilizes at a low level, whereas LLV on the test set keeps growing.

methods such as (Cisse et al., 2017). In essence, restricting the LLV can be formulated as a composite loss minimization problem:

$$\min_w \mathbb{E}_{(x,y)\sim\mathcal{P}_{\text{real}}}\Big[\ell\big(f(x; w), y\big) + \lambda \cdot \big\|\frac{\partial}{\partial x}\ell\big(f(x; w), y\big)\big\|_2\Big]. \tag{5}$$

Notice that (5) can be regarded as the "one-step approximation" of (3). In practice we need to change the expectation over $\mathcal{P}_{\text{real}}$ to empirical distribution of finite data,

$$\min_w \frac{1}{n}\sum_{i=1}^{n}\Big[\ell\big(f(x_i; w), y_i\big) + \lambda \cdot \big\|\frac{\partial}{\partial x_i}\ell\big(f(x_i; w), y_i\big)\big\|_2\Big], \tag{6}$$

where $\{(x_i, y_i)\}_{i=1}^{n}$ are feature-label pairs constitute the training set. Ideally, if we have enough data and model size is moderate then the objective function in (6) still converges to (5). However in practice when taking adversarial examples into account, we have one more problem to worry about: *Does small LLV in training set imply small LLV in test set?* The enlarged accuracy gap shown in Fig. 1 (*Left*) tends to give a negative answer. To verify this phenomenon directly, we calculate the LLV on images sampled from training and testing set respectively (Fig. 1 (*Right*)), we observe that in parallel with accuracy gap, the LLV gap between training and testing set is equally significant. Thus we conclude that *although adversarial training controls LLV around training set effectively, this property does not generalize to test set.* Notice that our empirical finding does not contradict the certified robustness of adversarial training using generalization theory (e.g. (Sinha et al., 2017)), which only explains weak attack situation.

The generalization gap can be potentially reduced if we have a better understanding of $\mathcal{P}_{\text{real}}$ instead of approximating it by training set. This leads to our first motivation: *can we use GAN to learn $\mathcal{P}_{real}$ and plug it into adversarial training algorithm to improve robustness on test set?* We will give a possible solution in Sec. 3.3.

3.2 MOTIVATION II: MORE EFFECTIVE GAN TRAINING BY ROBUST DISCRIMINATOR

GANs are notoriously hard to train. To our knowledge, there are two major symptoms of a failure trial — gradient vanishing and mode collapse. The theoretical explanation of gradient vanishing problem is discussed in (Arjovsky & Bottou, 2017) by assuming the images lie in a low dimensional manifold. Following this idea, (Arjovsky et al., 2017; Gulrajani et al., 2017a) propose to use 1-Wasserstein distance in place of the KL-divergence. The central character of WGAN and improved WGAN is that they require the set of discriminators $\{D(x; w)|\forall w \in \mathbb{R}^d\}$ equals to the set of all 1-Lipschitz functions w.r.t input x. Practically, we can either clip the weight of discriminator w (Arjovsky et al., 2017), or add a gradient norm regularizer (Gulrajani et al., 2017a). Recently, another regularization technique called "spectral normalization" (Miyato et al., 2018; Miyato & Koyama, 2018) is proposed to enforce 1-Lipschitz discriminator and for the first time, GAN learns to generate high quality images from full ImageNet data with only one generator-discriminator pair. In contrast, AC-GAN (Odena et al., 2017) — the supervised version of DCGAN — divides 1000 classes into 100 groups so each network-pair only learns 10 classes.

Despite the success along this line of research, we wonder if a weaker assumption to the discriminator is possible. Concretely, instead of a strict **one**-Lipschitz function, we require a **small local** Lipschitz value on image manifold. Indeed, we find a connection between robustness of discriminator and the learning efficiency of generator, as illustrated in Fig. 2.

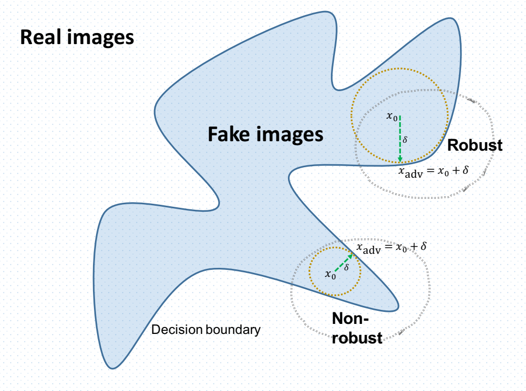

Figure 2: Comparing robust and non-robust discriminators, for simplicity, we put them together into one graph. Conceptually, the non-robust discriminator tends to make all images close to the decision boundary, so even a tiny distortion δ can make a fake image x_0 to be classified as a real image $x_{\text{adv}} = x_0 + \delta$. In contrast, such δ is expected to be much larger for robust discriminators.

As one can see in Fig. 2, if a discriminator $D(x)$ has small LLV (or $|D'(x)|$), then we know $D(x + \delta) \approx D(x) + D'(x) \cdot \delta \approx D(x)$ for a "reasonably" large δ. In other words, for robust discriminator, the perturbed fake image $x_{\text{adv}} = x_0 + \delta$ is unlikely to be mistakenly classified as real image, unless δ is large. Compared with adversarial attacks (2), the attacker is now a generator $G(z; w)$ parameterized by $w \in \mathbb{R}^d$ instead of the gradient ascend algorithm. For making x_0 "looks like" a real image (x_{adv}), we must update generator $G(z; w)$ to $G(z; w')$ and by assuming the Lipschitz continuity of G,

$$\|\delta\| = \|x_{\text{adv}} - x_0\| = \|G(z; w') - G(z; w)\| \leq L_G \|w - w'\|. \tag{7}$$

This indicates the movement of generator weights $\|w' - w\|$ is lower bounded by the distance of a fake image x_0 to the decision boundary, specifically we have $\|w' - w\| \geq \|\delta\|/L_G$. Furthermore, recall that a robust discriminator $D(x)$ implies a larger $\|\delta\|$, putting them together we know that improving the robustness of discriminator will lead to larger updates of the generator. In Sec. 4 we experimentally show that adversarial training not only speeds up the convergence to the equilibrium, but also obtains an excellent generator. But we leave the rigorous analysis for future works.

3.3 AdvGAN: Adversarial training on learned image manifold

Motivated by Sec. 3.1 and 3.2, we propose a system that combines generator, discriminator, and adversarial attacker into a single network. Our system consists of two stages, the first stage is an end-to-end GAN training: the generator feeds fake images to the discriminator; meanwhile real images sampled from training set are processed by PGD attacking algorithm before sending to the discriminator. After that the discriminator is learned to minimize both discrimination loss and classification loss (introduced below). In the next stage, the discriminator is refined by combining the fake and real images. The network structure is illustrated in Fig. 3. In what follows, we give more details about each component:

Discriminator: The discriminator could have the standard architecture like AC-GAN. In each iteration, it discriminates real and fake images. When the ground truth labels are available, it also predicts the classes. In this paper, we only consider the label-conditioning GANs (Mirza & Osindero, 2014; Odena et al., 2017; Miyato & Koyama, 2018), whose architectural differences are briefly overviewed in Fig. 4. Among them we simply choose AC-GAN, despite that SN-GAN (a combination of spectral normalization (Miyato et al., 2018) and projection discriminator (Miyato & Koyama, 2018)) performs much better in their paper. The reason we choose the AC-GAN is that

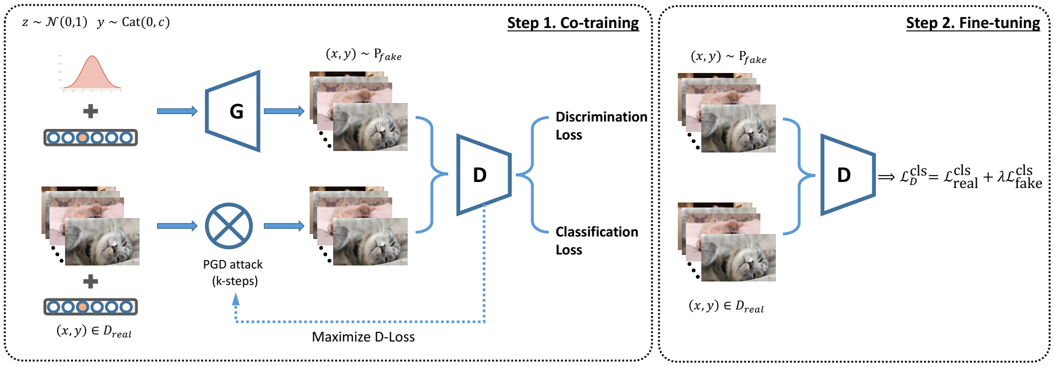
$z \sim \mathcal{N}(0,1)$ $y \sim \text{Cat}(0,c)$
Step 1. Co-training
$(x,y) \sim \mathbb{P}_{fake}$
G
D
Discrimination Loss
Classification Loss
PGD attack (k-steps)
$(x,y) \in D_{real}$
Maximize D-Loss
Step 2. Fine-tuning
$(x,y) \sim \mathbb{P}_{fake}$
D
$\Rightarrow \mathcal{L}_D^{\text{cls}} = \mathcal{L}_{\text{real}}^{\text{cls}} + \lambda \mathcal{L}_{\text{fake}}^{\text{cls}}$
$(x,y) \in D_{real}$

Figure 3: Illustration of the training process. Step-1 is the standard GAN training, i.e. alternatively updating the G and D networks. The only difference is that whenever feeding the real images to the D network, we first run 5 steps of PGD attack, so the discriminator is trained with adversarial examples. Step-2 is a refining technique, aiming at improving prediction accuracy on the test set.

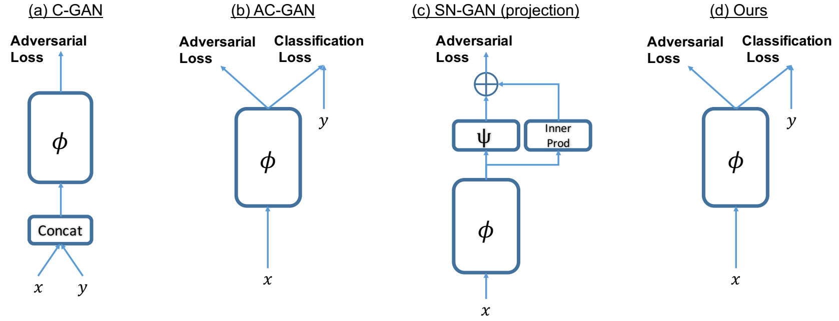

Figure 4: Comparing the architectures of discriminators. Our architecture is similar to AC-GAN (Odena et al., 2017), but they are different in loss functions, if one compares (8) with (9).

SN-GAN discriminator relies on the ground truth labels and their adversarial loss is not designed to encourage high classification accuracy. But surprisingly, even though AC-GAN is beaten by SN-GAN by a large margin, after inserting the adversarial training module, the performance of AC-GAN matches or even surpasses the SN-GAN, due to the reason discussed in Sec. 3.2. We also changed the loss objective of AC-GAN. Recall that the original loss in (Odena et al., 2017) defined by discrimination likelihood L_S and classification likelihood L_C:

$$\begin{aligned} L_S &= \mathbb{E}[\log \mathbb{P}(S = \text{real}|X_{\text{real}})] + \mathbb{E}[\log \mathbb{P}(S = \text{fake}|X_{\text{fake}})] \\ L_C &= \mathbb{E}[\log \mathbb{P}(C = c|X_{\text{real}})] + \mathbb{E}[\log \mathbb{P}(C = c|X_{\text{fake}})], \end{aligned} \tag{8}$$

where $X_{\text{real/fake}}$ are any real/fake images, S is the discriminator output, C is the classifier output. Based on (8), the goal of discriminator is to maximize $L_S + L_C$ while generator aims at maximizing $L_C - L_S$. According to this definition, both G and D are optimized to increase L_C: even if $G(z)$ produces unrecognizable images, $D(x)$ has to struggle to classify them (with high loss), in such case the corresponding gradient term ∇L_C can contribute uninformative direction to the discriminator. To resolve this issue, we split L_C as follows,

$$L_{C_1} = \mathbb{E}[\log \mathbb{P}(C = c|X_{\text{real}})], \quad L_{C_2} = \mathbb{E}[\log \mathbb{P}(C = c|X_{\text{fake}})], \tag{9}$$

then discriminator maximizes $L_S + L_{C_1}$ and generator maximizes $L_{C_2} - L_S$. The new objective functions ensure that discriminator only focuses on classifying the real images and discriminating real/fake images.

Generator: Similar to the traditional GAN training, the generator is updated on a regular basis to mimic the distribution of real data. This is the key ingredient to improve the robustness of discriminators: as shown in Sec. 3.1, adversarial training performs well on training set but is vulnerable on test set. Intuitively, this is because during adversarial training, the network only "sees" adversarial examples residing in the $\delta_{\max}$-ball of all training samples, whereas the rest images in the data manifold are undefended. Data augmentation is a natural way to resolve this issue, but traditional techniques (Krizhevsky et al., 2012; Halevy et al., 2009; Tokozume et al., 2017; Zhang et al., 2017; Inoue, 2018) rely largely on combinations of geometric transforms to the training images, in our case the support of the probability density function is still very small. Instead, our system uses images sampled from generator to provide a continuously supported p.d.f. for the adversarial training. Unlike traditional augmentation methods, if the equilibrium in (1) is reached, then we can show that one desirable solution of (1) would be $\mathcal{P}_{\text{fake}}(z) \overset{\text{dist.}}{=} \mathcal{P}_{\text{real}}$, and therefore the robust classifier can be trained on the learned distribution.

Fine-tuning the classifier: This step aims at improving the classification accuracy, based on the auxiliary classifier in the pretrained discriminator. This is crucial because in the GAN training stage (step 1 in Fig. 3), the discriminator is not trained to minimize the classification error, but a weighted loss of both **discrimination** and **classification**. But in step 2, we want to focus on the robust classification task

$$\begin{aligned} \mathcal{L}_D^{\text{cls}} &\triangleq \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_{\text{real}}} \ell(f(x_{\text{adv}}; w), y) + \lambda \cdot \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_{\text{fake}}} \ell(f(x_{\text{adv}}; w), y), \\ \text{where } x_{\text{adv}} &= \mathop{\arg\min}_{\|x'-x\|\le\delta_{\max}} \ell(f(x'; w), y). \end{aligned} \tag{10}$$

Here the function $f(x; w)$ is just the classifier part of network $D(x)$, recall that we are dealing with conditional GAN. As we can see, throughout the fine-tuning stage, we force the discriminator to focus on the classification task rather than the discrimination task. It turns out that the fine-tuning step boosts the accuracy by a large margin. Adversarial attacker is omitted in Fig. 3 due to width limit.

4 Experiment

We experiment on both CIFAR10 and a subset of ImageNet data. Specifically, we extract classes y_i such that $y_i \in$ `np.arange(151, 294, 1)` from the original ImageNet data: recall in total there are 1000 classes in ImageNet data and we sampled $294 - 151 = 143$ classes from them. We choose these datasets because 1) the current state-of-the-art GAN, SN-GAN (Miyato & Koyama, 2018), also worked on these datasets, and 2) the current state-of-the-art adversarial training method (Madry et al., 2017) only scales to CIFAR dataset. For fair comparison, we copy all the network architectures for generators and discriminators from SN-GAN, other important factors, such as learning rate, optimization algorithms, #discriminator updates in each cycle, etc. are also kept the same. The only modification is that we discarded the feature projection layer and applied the auxiliary classifier (see Fig. 4). Please refer to the appendix or source code for more implementation details.

Effect of fine-tuning In what follows, we check whether fine-tuning helps improving test set accuracy. To this end, we design a experiment that compares two set of models: in the first set, we directly extract the auxiliary classifiers from discriminators to classify images; in the next set, we apply fine-tuning strategy to the pretrained model as Fig. 3 illustrated. The results can be found in Fig. 5, which supports our argument that fine-tuning is indeed useful for better prediction accuracy.

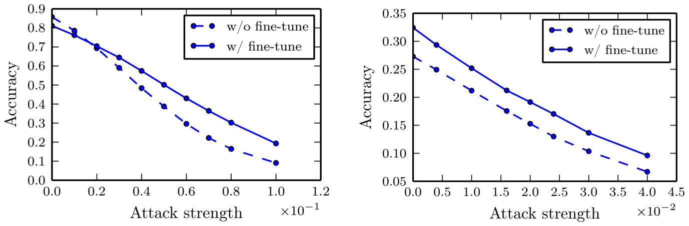

Figure 5: The effect of fine-tuning on prediction accuracy (*left:* CIFAR10, *right:* ImageNet-64px)

Robustness of discriminator: comparing robustness with/ without data augmentation In this experiment, we would like to compare the robustness of discriminator networks with or without data augmentation technique discussed in Sec. 3.3. The robustness is measured by the prediction accuracy under adversarial attack. For networks without data augmentation, that would be equal to the state-of-the-art Madry's algorithm (Madry et al., 2017). For attacking algorithm, we choose the widely used ℓ_∞ PGD attack (Madry et al., 2017), but other gradient based attacks are expected to

Dataset	$\sigma_{\max}$ of ℓ_∞ attacks			
	0	0.02	0.04	0.08
CIFAR10	81.1% (-0.35%)	70.41% $(+1.26\%)$	57.43% $(+3.69\%)$	30.25% $(+6.67\%)$
	0	0.01	0.02	0.03
ImageNet[†] (64px)	32.4% $(+6.35\%)$	25.2% $(+6.9\%)$	19.1% $(+6.58\%)$	13.7% $(+5.38\%)$

[†] Denotes the 143-class subset of ImageNet.

Table 1: Accuracy of our model under ℓ_∞ PGD-attack. Inside the parenthesis is the improvement over standard adversarial training defense (Madry et al., 2017).

yield the same results. We set the ℓ_∞ perturbation to $\sigma_{\max} \in$ `np.arange(0, 0.1, 0.01)` as defined in (2). Another minor detail is that we scale the images to $[-1, 1]$ rather than usual $[0, 1]$. This is because generators always have a $\tanh()$ output layer, so we need to do some adaptations accordingly. We exhibit the results in Tab. 1, showing our method can improve the robustness of state-of-the-art defensive algorithm.

Effect of split classification loss Here we show the effect of split classification loss described in (9), recall that if we apply the loss in (8) then the resulting model is AC-GAN. It is known that AC-GAN can easily lose modes in practice, i.e. the generator simply ignores the noise input z and produces fixed images according to the label y. This defect is observed in many previous works (Huang et al., 2017; Mathieu et al., 2015; Isola et al., 2017). In this ablation experiment, we compare the generated images trained by two loss functions, the result is shown in Fig. 6.

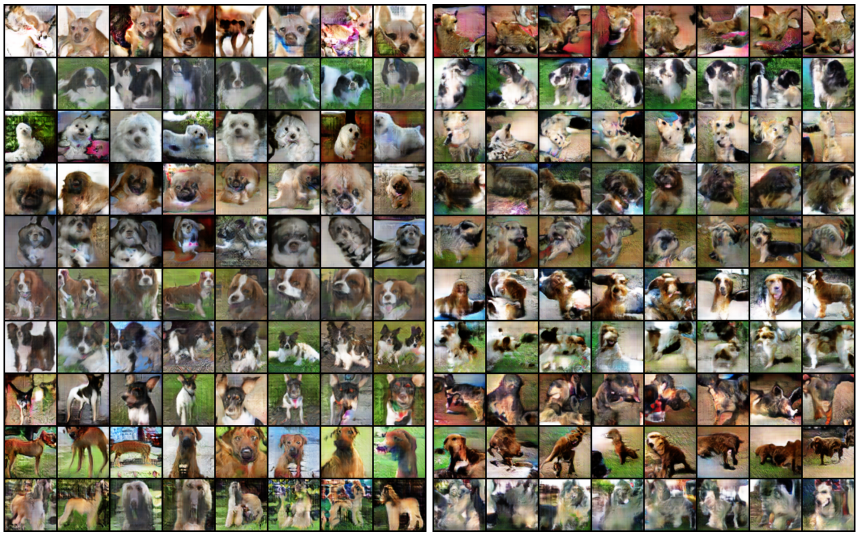

Figure 6: Comparing the generated images trained by our modified loss(*left*) with the original AC-GAN loss(*right*). For fair comparison, both networks are trained with adversarial real images. We can see images from AC-GAN are more distorted and harder to distinguish.

Quality of generator and convergence speed In the last experiment, we compare the quality of generators trained in three datasets: CIFAR10, ImageNet subset (64px) and ImageNet subset (128px). Our baseline model is the SN-GAN, considering that, as far as we know, SN-GAN is the best GAN model capable of learning hundreds of classes. Note that SN-GAN can also learn the conditional distribution of the entire ImageNet data (1000 classes), unfortunately, we are not able to match this experiment due to time and hardware limit. To show that the adversarial training technique indeed accelerates the convergence speed, we also tried to exclude adversarial training — this is basically an AC-GAN, except that an improved loss function discussed in (9) is applied to discriminator $D(x)$. The results are exhibited in Fig. 7, which shows that adversarial training can improve the performance of GAN, and our generator achieves better inception score than SNGAN. Another finding is that our new loss proposed in (9) works much better than the original AC-GAN loss. (8). Last of all, we check

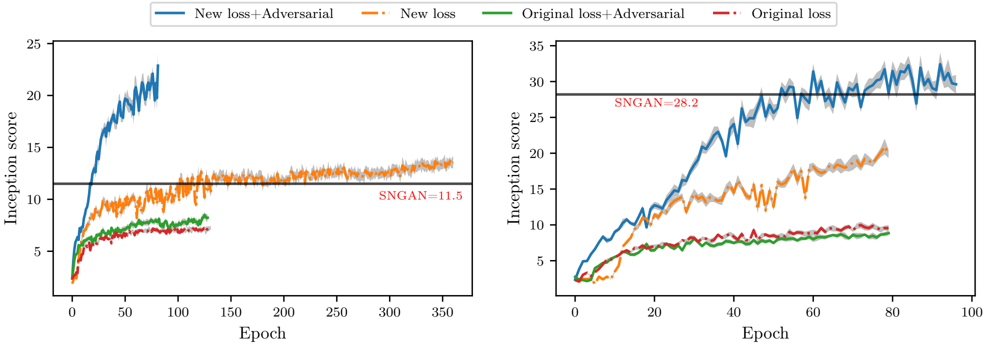

Figure 7: Results on subset of ImageNet, left: 64px, right: 128px. We compare the inception scores between our model and the SN-GAN. Clearly our method learns a high quality generator in a short time, specifically, in both datasets, AC-GAN with adversarial training surpasses SN-GAN in just 25 epochs (64px) or 50 epochs (128px). Another observation is that with adversarial training, the convergence is greatly accelerated.

whether adversarial training with fake data augmentation really shrinks the generalization gap. To this end, we draw the same figure as Fig. 1, except that now the classification model is the discriminator after fine-tuning step (shown in Fig. 3). We compare the accuracy gap in Fig. 8. Clearly the model trained with the adversarial *real+fake augmentation* strategy works extremely well: it improves the testing accuracy under PGD-attack and so the generalization gap between training/testing set does not increase that much.

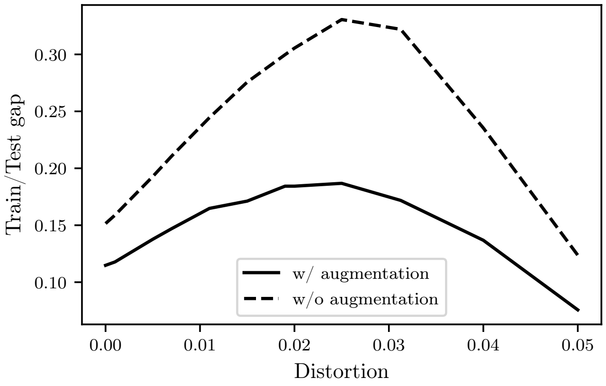

Figure 8: Comparing the accuracy gap between adversarial training model and GAN data augmentation model.

5 Discussion

In this paper, we draw a connection between adversarial training (Madry et al., 2017) and generative adversarial network (Goodfellow et al., 2014a). Our primary goal is to improve the generalization ability of adversarial training and this is achieved by data augmentation by the unlimited fake images. Independently, we see an improvement of both robustness and convergence speed in GAN training. While the theoretical principle in behind is still unclear to us, we gave an intuitive explanation. Apart from that, a minor contribution of our paper is the improved loss function of AC-GAN, showing a better result in image quality.

References

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Anish Athalye and Nicholas Carlini. On the robustness of the cvpr 2018 white-box adversarial example defenses. *arXiv preprint arXiv:1804.03286*, 2018.

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.

David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=S18Su--CW`.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pp. 854–863, 2017.

Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pp. 1486–1494, 2015.

Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=H1uR4GZRZ`.

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*, pp. 2672–2680, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5769–5779, 2017a.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5769–5779, 2017b.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SyJ7ClWCb`.

Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. 4, 2017.

Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

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.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pp. 2200–2210, 2017.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pp. 1718–1727, 2015.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Michael E. Houle, Dawn Song, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B1gJ1L2aW`.

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.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=ByS1VpgRZ`.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B1QRgziT-`.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*, pp. 2642–2651, 2017.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pp. 582–597. IEEE, 2016.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BkJ3ibb0-`.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJUYGxbCW`.

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.

Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Learning from between-class examples for deep sound recognition. *arXiv preprint arXiv:1711.10282*, 2017.

Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, Martin Heusel, Hubert Ramsauer, and Sepp Hochreiter. Coulomb gans: Provably optimal nash equilibria via potential fields. *arXiv preprint arXiv:1708.08819*, 2017.

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pp. 4790–4798, 2016.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Sk9yuql0Z`.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.