Crafting Adversarial Attacks with Adversarial Transformations

Kawin Ethayarajh
kawin@cs.toronto.edu

Avishek (Joey) Bose
joey.bose@mail.utoronto.ca

Romina Abachi
romina@psi.toronto.edu

Mohammad Firouzi
firouzi@cs.toronto.edu

Abstract

In adversarial attacks, small, often imperceptible, perturbations are added to inputs with the goal of getting a neural network to misclassifying them. Different adversarial attack strategies have been proposed, but the creation of adversarial examples that are fast to produce, potent, and transferable to many deep neural networks (DNNs) remains an open problem. In this paper, we propose a novel strategy to craft adversarial examples by training an adversarial generator network jointly with the model under attack. Our approach is fast and scalable, given that a perturbed image can be crafted from a forward pass through our trained generator network. We then show on CIFAR10 that using our trained generator, it is possible to craft adversarial attacks against other DNNs in a black-box manner. In a different experiment, also on CIFAR10, we demonstrate that adversarially training a classifier – where the classifier is tasked with correctly classifying both unperturbed and perturbed images – greatly improves robustness. Our attack is faster and assumes less than both the Fast Gradient Sign Method and Carlini Wagner attacks while being stronger and more transferable than the former on very deep neural networks.

1 Introduction

Given the application of deep neural networks to problems as varied as vehicle automation [20] and cancer detection [4], it is imperative to better understand the ways in which these models are vulnerable to attack. In the domain of image recognition, Szegedy et al. [24] found that small, often imperceptible, perturbations can be added to images to fool a network into misclassifying them. Such perturbed images are called *adversarial examples*. These adversarial examples can then be used in conducting *adversarial attacks* on networks. There are several known methods for crafting adversarial examples, and they vary greatly with respect to complexity, computational cost, and the level of access they require to the attacked model.

A baseline approach is the Fast Gradient Sign Method (FGSM) [6], where an attack is crafted based on the gradient of the input image, x, with respect to the classifier loss. FGSM is a white-box approach, meaning it requires access to the internals of the classifier being attacked. It is also limited to *untargeted* attacks, where the aim is simply to get the classifier to misclassify an input; in targeted attacks, the goal is to fool the classifier into predicting a particular class. Though FGSM is only worst-case for linear models, it is moderately effective against neural networks, with more linear models being more vulnerable to attack [6]. Adversarial examples generated using FGSM were found to be highly transferable, which Goodfellow et al. [6] ascribe to different classifiers all learning similar mapping functions and decision boundaries.

There are several strong adversarial attacks for attacking deep neural networks, such as L-BFGS [24], Jacobian-based Saliency Map Attack (JSMA) [19], DeepFool [17], and Carlini-Wagner [3]. However,

CSC2541 Fall 2017

these methods all involve some complex optimization over the space of possible perturbations, making them slow and computationally expensive. Papernot et al. [18] have had some success in making these attacks more practical by using a combined white-box / black-box attack, whereby the adversarial image is generated using a model the attacker has full access to (white-box) and then used to attack a classifier the attacker has no access to (black-box). Still, the distortion of the perturbation typically needs to be much greater for transferable attacks to work well.

In this paper, we propose a baseline for untargeted attacks of deep neural networks (DNNs) in place of FGSM. Our goal is not to compete with the likes of Carlini-Wagner [3], but rather to design a fast and inexpensive method for transferable attacks on DNNs that is more effective than FGSM, which is only optimally worst-case for linear models [6]. We propose training a generator – a small neural network called the Generative Adversarial Transformation Network (GATN) – and a DNN classifier concurrently. Given an image, the generator produces a small perturbation that can be added to the image to fool the classifier. The classifier is trained only on unperturbed images and as such remains oblivious to the generator's presence. Over time, the generator learns to produce perturbations that can effectively fool the classifier it is trained with. Generating an adversarial example with a GATN is fast and inexpensive, even more so than for FGSM, since creating a perturbation for an input only requires a forward pass once the generator is sufficiently well-trained.

We test our approach using CIFAR-10 [10]. We find that when a GATN is used to attack the classifier it is trained with, it outperforms FGSM for very deep neural network architectures such as VGG [22] and ResNet [7], while slightly underperforming FGSM for simpler architectures such as LeNet [13]. We also find that if a GATN is trained using one DNN architecture, such as VGG, then the adversarial images it generates are often highly transferable to other DNNs, such as ResNet. This allows for black-box attacks in the style proposed by Papernot et al. [18]: give the GATN white-box access to some known DNN classifier, train the GATN, and then use it to conduct black-box attacks on unknown DNNs. In our experiments, GATN attacks were also almost twice as fast as FGSM attacks.

We then explore a variant of our approach in which the classifier is adversarially trained with the GATN, thus it is trained on both unperturbed and perturbed images. This is similar to a Generative Adversarial Network (GAN) [5], but the GATN in our case is conditioned on the image and we do not sample from a latent distribution. We find that adversarial training makes classifiers more robust to attacks from the GATN they are trained with. For very deep architectures like VGG, the adversarially trained classifier also becomes more robust to attacks from GATNs that are adversarially trained with other very deep architectures like Resnet. Most notably, transferable attacks are much more effective on a simpler classifier like LeNet if the attacking GATN has been adversarially trained. To explain some of our findings, we refer to the theories proposed by Goodfellow et al. [6] in their seminal work on adversarial attacks.

2 Related Work

Adversarial attacks can be grouped by the level of access they have to the attacked model and by their adversarial goal. *White-box attacks* have full access to the architecture and parameters of the model that they are attacking; *black-box* attacks only have access to the output of the attacked model [18]. Adversarial attacks can also be grouped into *targeted* and *untargeted* attacks. Given an input image x, class label y and a classifier $D(x) : x \rightarrow y$ to attack, the goal of an untargeted attack is to solve $\text{argmin}_{x'} L(x, x')$ such that $D(x) \neq D(x')$, where L is a distance function between the unperturbed and perturbed inputs [2]. The goal of a targeted attack is to solve $\text{argmin}_{x'} L(x, x')$ such that $D(x') = t'$, where t' is a target class chosen by the attacker. In this paper, we focus only on untargeted attacks. In this section, we look at some common attacks on the opposite ends of the spectrum with respect to complexity and effectiveness.

2.1 Fast Gradient Sign Method

Given an image x, the Fast Gradient Sign Method (FGSM) [6] returns a perturbed input x':

$$x' = x - \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

where J is the loss function for the attacked classifier and ϵ controls the extent of the perturbation, set to be sufficiently small that the perturbation is undetectable by eye. Intuitively, FGSM works by taking the gradient of the loss function to determine which direction a pixel's intensity should be

changed to minimize the loss function. Then it shifts the pixel in the other direction. When done for all pixels simultaneously, the classifier is more likely to misclassify x'. FGSM is intended for use in untargeted tasks, but Papernot et al. [18] used them in targeted attacks by trying different values of ϵ until one resulted in the classifier misclassifying x' as the targeted class x'. FGSM is only worst-case for linear models, but also works relatively well on neural networks, which Goodfellow et al. ascribe to the high level of linearity in neural networks [6].

2.2 Carlini-Wagner

The Carlini-Wagner method [3] is used for conducting both targeted and untargeted attacks. The adversarial goal is finding some minimal perturbation δ such that $D(x+\delta) = t'$, where D is the classifier, x is some input, t' is the target class, and δ is the perturbation. This is expressed as:

$$\text{argmin}_\delta \|\delta\|_p + c \cdot f(x+\delta) \text{ s.t. } x+\delta \in [0,1]^n$$

where f is an objective function such $D(x+\delta) = t' \Leftrightarrow f(x+\delta) \leq 0$. In our implementation of the Carlini-Wagner attack, we use the L_2 distance metric with the objective function $f(x') = \max(\max(\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa)$, where Z is the output of all layers of the classifier except the softmax layer and κ is a confidence hyperparameter. f was the best objective function found by the original authors [3]. It encourages the solver to find a perturbation such that the perturbed input will be classified as the target class t with high confidence, at least relative to the other possible classes. The Carlini-Wagner attack is very strong – achieving over 99.8% misclassification on CIFAR-10 – but also slow and computationally expensive [3].

2.3 Adversarial Transformative Networks

An Adversarial Transformative Network (ATN) is any neural network that, given an input image, returns an adversarial image to be used against a particular classifier(s). Baluja et al. provide a broad formulation [2]:

$$\text{argmin}_\theta \sum_{x_i \in \mathcal{X}} \beta \cdot L_{\mathcal{X}}(g_{f,\theta}(x_i), x_i) + L_{\mathcal{Y}}(f(g_{f,\theta}(x_i)), f(x_i))$$

where β is a scalar, $L_{\mathcal{X}}$ is a perceptual loss (e.g., the L_2 distance) between the original and perturbed inputs and $L_{\mathcal{Y}}$ is the loss between the classifier's predictions on the original inputs and the perturbed inputs. In the original paper, Baluja et al. [2] use $L_{\mathcal{Y}} = L_2(f(x'), r(f(x), t))$, where r is a re-ranking function meant to encourage better reconstruction. ATNs were less effective than strong attacks like Carlini-Wagner, and the adversarial images they generated were not found to be transferable for use in black-box attacks. One key advantage ATNs have is that they are fast and inexpensive to use: an adversarial image can be created with just a forward pass through the ATN.

2.4 Generative Adversarial Nets

Generative Adversarial Network (GAN) training involves jointly training two networks: a generator, tasked with producing samples from some distribution that ideally mimics examples from the true data distribution, and a discriminator, which attempts to differentiate between samples from the true data distribution and the one produced by the generator. In its most vanilla form the GAN objective can be written as a min max optimization problem with the following form:

$$\min_G \max_D < \log(D(x)) >_{P_r(x)} + < \log(1 - D(\tilde{x})) >_{P_g(\tilde{x})},$$

where $< . >$ denotes expectation, $\mathbb{P}_r$ is the true data distribution and $\mathbb{P}_g$ is the model distribution which is implicitly defined by the generator. A generated sample $\tilde{x} = G(z), z \sim P(z)$, where z is sampled from some noise distribution such as a Gaussian. It has been shown that a discriminator trained to optimality minimizes the Jensen Shannon Divergence (JSD). In practice it is common for the generator to maximize $< \log(D(\tilde{x})) >_{P_g(\tilde{x})}$ [5]. If both the discriminator and generator are conditioned on additional information, such as the label, then the quality of generated samples significantly improves. Other variants of GANs minimize other metrics such as the Wasserstein distance [1]. In this work, we only consider the JSD GAN and its conditional variant [16].

3 Approach

3.1 Threat Model

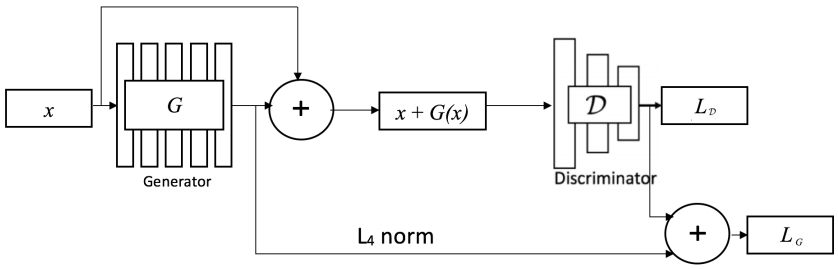


Figure 1: The proposed GATN architecture where a generator network G creates image conditional perturbations in order to fool a discriminator D. G's loss is based on its success in fooling D and the magnitude of the L_4 perturbation norm, while D's loss is the conventional cross-entropy loss.

Our model is most similar to that of an Adversarial Transformation Network (ATN) [2], a label that broadly applies to any generator network used to create adversarial attacks. However, it is significantly different from the specific type of ATN that was proposed and tested by Baluja et al. [2]. We train two networks – a discriminator D and a generator G – concurrently. G is the Generative Adversarial Transformation Network (GATN) and is conditioned on the same input image x as seen by the classifier D. Specifically, G produces a small perturbation that can be added to x to produce an adversarial image x'. D remains oblivious to the presence of G while G's loss depends on how well it can fool D into misclassifying x'. Over time, G produces perturbations that can effectively fool the discriminator D it is trained with. Once fully trained, G can be used to generate image-conditional perturbations with a simple feed-forward operation. The loss functions for D and GATN G are:

$$L_D(\theta_d, x, y) = -\log D(x; \theta_d)_y \tag{1}$$

$$L_G(x, y) = D(x + \epsilon \cdot G(x); \theta_d)_y + c_g \cdot \max\left(||G(x)||_p, 1^{-3}\right) \tag{2}$$

where ϵ is the degree of perturbation, c_g is a weight on the L_p regularization term, and p denotes the p-norm. D has a regular cross-entropy loss while G's loss is the likelihood of the perturbed images and a L_p regularization term with a lower bound of 1^{-3}. The loss functions of our discriminator and generator scarcely resemble those in Baluja et al. [2]. Specifically, the differences are:

- Instead of training a generator network on a fully trained discriminator, as in Baluja et al. [2], we concurrently train the discriminator and generator. We find that this is important in creating adversarial images that are transferable, which is a limitation of their ATN.
- Given an image, GATN produces a perturbation, unlike Baluja et al.'s ATN, which produces a complete perturbed image.
- For L_G, we use the likelihood of the perturbed images with respect to D's loss function. This means that we do not consider the predictions made by D on the unperturbed images, only the true labels for the unperturbed images.
- For the perceptual loss between the perturbed and unperturbed images, we use the p-norm of the perturbation instead of the L_2 norm. Our perceptual loss also has a lower bound of 1^{-3}. We find that this is necessary to generate a perturbation that is effective enough to fool D while still being imperceptible. The L_2 norm did not work well here; the perturbations tended to explode in size early on in training and remained that way. Empirically, we find that $p = 4, c_g = 0.005, \epsilon = 1.0$ work best. When p is larger, there is the risk of the opposite happening, of the perturbations vanishing early on in training and never recovering. Having a lower bound is a simple but effective way of mitigating this.

3.2 Black-Box Attacks

We are motivated by the transferability of adversarial attacks between DNNs, as explored by Papernot et al. [18]. However, in our black-box approach we do not need to synthetically create a dataset first to train a substitute model. Instead we simply use our trained G on a different D' that was not used to train G. Concretely, during training, the GATN G has white-box access to the discriminator D it is being trained with. Any subsequent adversarial attack by G on D is therefore a white-box attack. We can conduct black-box attacks on some unknown discriminator D' by training G with some known neural network classifier and then using G, once fully trained, to generate adversarial images for D'.

3.3 Adversarial Training

A variant of our basic approach in subsection 3.1 is to train the discriminator D adversarially, similar to a discriminator in a Conditional Generative Adversarial Network (CGAN) [5]. This encourages D to become more robust to perturbed images and encourages G to produce stronger perturbations that can fool the increasingly robust D. The loss function for the adversarially trained discriminator D is:

$$L_D(\theta_f, x, y) = -\log D(x + \epsilon \cdot G(x); \theta_d)_y - \log D(x; \theta_d)_y \tag{3}$$

where the discriminator's loss is the cross-entropy over the regular inputs and the cross-entropy over the perturbed inputs. The generator loss remains unchanged from our original formulation. It is important to note that our approach is strictly not a CGAN as we do not sample from a latent distribution for the generator. Adversarially trained GATNs can also be used in black-box attacks as described in subsection 3.2.

4 Results, Discussion, and Limitations

In this section, we use GATN for white-box and black-box attacks using images from CIFAR-10 [11]. We also compare our attacks to FGSM and Carlini-Wagner and show that our method achieves comparable success rates to the former in much lower computation time, as shown in table 1.

We use the subscript AT to indicate that a discriminator or generator has been trained adversarially. For example, D_{AT} signifies that the discriminator D has been trained adversarially with a GATN (i.e., it has been trained on both unperturbed images and images perturbed with a GATN). G_{AT} denotes the GATN in the other half of that training: it is the generator used to produce the perturbed images used to adversarially train D_{AT}. This means that there are four possible configurations for attacks: $\langle D, G\rangle, \langle D_{AT}, G_{AT}\rangle, \langle D_{AT}, G\rangle, \langle D, G_{AT}\rangle$.

White-box attacks are only possible with the first two configurations: since the discriminator and GATN must be trained together, either both are adversarially trained or neither are. Black-box attacks are possible with all four configurations. However, in a black-box attack, the discriminator and generator have not been trained together: for example, even in the $\langle D_{AT}, G_{AT}\rangle$ setting, D_{AT} must have been trained adversarially with some different GATN ${G'}_{AT}$ and G_{AT} must have been trained with some different discriminator ${D^*}_{AT}$.

Table 1: Comparison of computation time with state-of-the-art methods. Runtime is measured for generating 1000 samples using 1 Nvidia GPU GeForce GTX-1080 Ti GPU.

	FGSM	C-W	GATN
Runtime	2.21s	>6300s	1.21s

Implementation details: The GATN can take the form of any generator network; we use the simple architecture from Lee et al. [15]. We train the GATN with the following discriminator networks: VGG16 [22], ResNet18 [7], DenseNet121 [8], AlexNet [12], GoogleNet [23], and LeNet [14]. We then evaluate the trained GATN on a target classifier that has one of those architectures. In the non-adversarial setting, we train the discriminator with loss 1. In the adversarial setting, we train the discriminator with loss 3. In both settings, we train the GATN with loss 2. We tune the hyperparameters using 5,000 validation samples from the original 50,000 training samples. We use a

batch size of 128 and learning rate 0.001. Our optimizer is Adam [9]. Due to resource constraints, we could not train the classifiers as fully as would have been required to get state-of-the-art accuracy[1].

Some adversarial examples generated with a GATN are shown in Figure 2, where the GATN is trained non-adversarially with a VGG16 discriminator. Note that the GATN does not generate a complete image, like an ATN [2], but rather a perturbation that is added to the original image. It can be seen that the perturbations generated by the GATN are almost imperceptible.

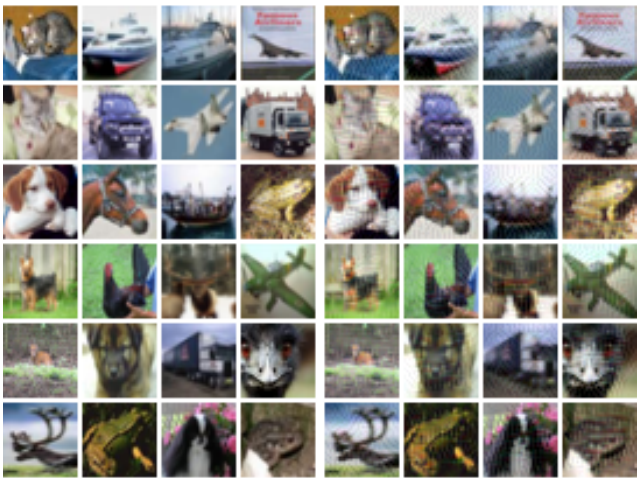


Figure 2: Adversarial images generated with a GATN trained with VGG16 using images from CIFAR-10. The first four columns contain the original images; the last four the perturbed images.

4.1 White-Box Attacks

In white-box attacks, a GATN G trained with some discriminator D is used to attack D once fully trained. Therefore the attacker has full access to the architecture and parameters of the classifier being attacked. Figure 3a contains the results of the attacks when the GATN G is non-adversarially trained with D. The GATN attack is quite effective at lowering the test accuracy of D and is a better attack than FGSM for 4/6 DNN architectures, namely VGG16, GoogleNet, ResNet18, and DenseNet121. The biggest discrepancy is with DenseNet121: an FGSM attack yields an accuracy of 25.0% but a GATN attack yields an accuracy of 15.2%.

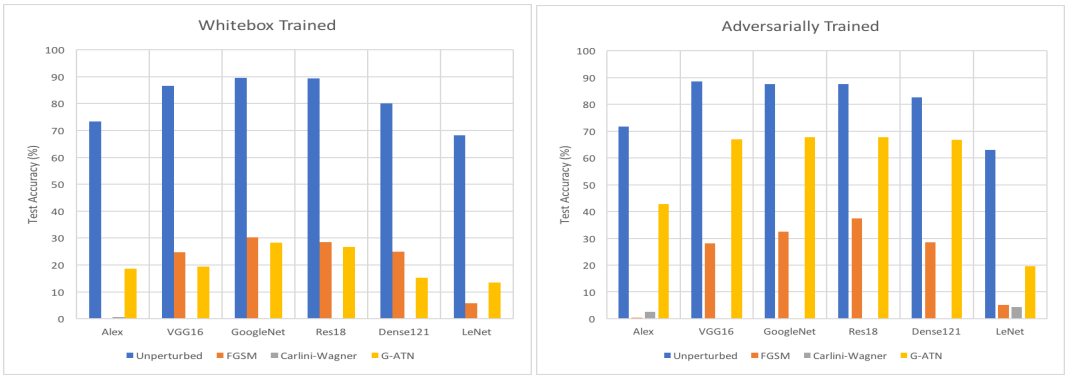
Figure 3b contains the results of the attacks when a GATN G_{AT} is adversarially trained with discriminator D_{AT}. Adversarial training makes the discriminators more robust to the GATN's attacks, as expected. The adversarially trained discriminators also become more robust to FGSM attacks, though to a far less extent than with the GATN attacks. Adversarially-trained AlexNet and LeNet are the only discriminators that show an increased robustness to the Carlini-Wagner attack.

The fact that FGSM outperforms the GATN attack on simpler models like LeNet is not surprising, and is one limitation of our approach. FGSM is only worst-case for a perfectly linear model, and Goodfellow et al. have hypothesized that the more linearity there is in a model, the more susceptible it is to FGSM attacks [6]. Deeper, more complex models such as VGG and ResNet are less susceptible to FGSM attacks, which explains why our GATN attack is more effective on them.

4.2 Black-Box Attacks

In this section, we demonstrate the transferability of GATN attacks by attacking a classifier D' that the GATN did not have access to during training. In black-box attacks, we train the GATN with some discriminator D and then used the trained GATN to attack some different classifier D'. These results for when the GATN is trained non-adversarially are available in Table 2. With the exception of AlexNet, which is fairly robust to black-box attacks from GATNs, our black-box attacks are as effective – and in some cases, more effective – than white-box attacks. For example, a GATN trained

[1] All code available online at https://github.com/kawine/atgan



(a) (b)

Figure 3: Test accuracy of various classifiers on original CIFAR-10 data and on images perturbed by different attacks: FGSM, Carlini-Wagner, and our GATN trained with full access to the architecture and parameters of the attacked model. 3a is for the whitebox threat model and 3b is for the adversarially-trained threat model

with DenseNet121 is more effective at attacking a ResNet18 classifier than a GATN trained with that classifier (21.9% accuracy vs. 26.8% accuracy, respectively). When the classifier D'_{AT} being attacked is adversarially trained with a different GATN G'_{AT}, it becomes more robust to attacks from the non-adversarially trained G. These results are given in table 3.

The transferability of GATN attacks is not surprising. Goodfellow et al. [6] have hypothesized that adversarial attacks are as transferable as they are because of different classifiers all learning similar mapping functions. However, in our experiments, this transferability only really exists between very deep neural networks (VGG, ResNet, GoogleNet, and DenseNet); AlexNet and LeNet are fairly robust to our black-box attacks. Building on Goodfellow et al.'s hypothesis, this suggests that the very deep neural networks all learn a very similar mapping function, but one that is ultimately different from that of simpler networks like LeNet. The lack of transferability of GATN attacks to simpler neural networks is another limitation of our approach.

Table 4 contains the results of the attacks when a GATN G_{AT} is adversarially trained with some discriminator D_{AT} before being used to attack a different classifier D' that was naively trained. Adversarially trained GATNs are slightly less effective in black-box attacks than naively trained GATNs on the four DNNs for which naively trained GATNs outperformed FGSM. However, they are much more effective in black-box attacks on AlexNet and LeNet than their naively trained counterparts. When the attacked classifier D'_{AT} is adversarially trained – with its own GATN, given that this is a black-box attack – it also becomes more robust to attacks from the adversarially trained GATNs; these results are given in Table 5.

Table 2: Test accuracy of various classifiers on CIFAR-10 when attacked by GATNs non-adversarially trained with different classifiers (i.e. GATN G trained with D is used to attack a classifier D'). The diagonal show the white-box attacks.

	Architecture of D in GATN					
Attacked Model	AlexNet	VGG16	GoogleNet	ResNet18	DenseNet121	LeNet
AlexNet	**18.6%**	64.0%	61.0%	61.0%	55.0%	43.5%
VGG16	70.8%	**19.4%**	22.3%	19.6%	23.2%	48.4%
GoogleNet	67.0%	24.4%	**28.2%**	26.5%	24.4%	53.7%
ResNet18	72.0%	24.2%	25.4%	**26.8%**	21.9%	56.9%
DenseNet121	58.6%	17.1%	20.9%	17.4%	**15.2%**	40.5%
LeNet	22.1%	31.8%	30.2%	31.0%	23.3%	**13.4%**

Table 3: Test accuracy of various adversarially-trained classifiers on CIFAR-10 when attacked by GATNs non-adversarially trained with different classifiers (i.e. GATN G trained with D is used to attack a classifier D'_{AT}). The diagonal show the white-box attacks.

	Architecture of D in GATN					
Model	AlexNet	VGG16	GoogleNet	ResNet18	DenseNet121	LeNet
AlexNet	**20.9**%	68.9%	64.7%	64.6%	62.9%	56.2%
VGG16	68.9%	**14.5**%	27.2%	18.75%	20.3%	56.2%
GoogleNet	72.7%	20.6%	**22.3**%	20.9%	24.2%	63.4%
Res18	79.2%	28.5%	35.6%	**19.5**%	28.9%	63.0%
DenseNet121	64.8%	22.4%	26.7%	19.3%	**14.18**%	82.6%
LeNet	37.4%	48.7%	47.7%	46.2%	44.6%	**27.0**%

Table 4: Test accuracy of various classifiers on CIFAR-10 when attacked by GATNs adversarially trained with different classifiers (i.e. GATN G_{AT} trained with D_{AT} is used to attack a naively trained classifier D'). The diagonal show the white-box attacks.

	Architecture of D in GATN					
Model	AlexNet	VGG16	GoogleNet	ResNet18	DenseNet121	LeNet
AlexNet	**17.1**%	18.6%	25.0%	19.5%	22.9%	12.2%
VGG16	28.0%	**30.4**%	26.5%	32.4%	29.4%	21.6%
GoogleNet	25.1%	29.0%	**30.6**%	27.3%	26.1%	21.1%
ResNet18	31.1%	29.2%	25.4%	**33.2**%	28.3%	23.5%
DenseNet121	27.1%	26.2%	28.3%	30.8%	**25.9**%	18.1%
LeNet	17.7%	20.7%	21.2%	21.6%	20.4%	**16.8**%

Table 5: Test accuracy of various adversarially-trained classifiers on CIFAR-10 when attacked by GATNs adversarially trained with different classifiers (i.e. GATN G_{AT} trained with D_{AT} is used to attack a different adversarially trained classifier D'_{AT}). The diagonal show the white-box attacks.

	Architecture of D in GATN					
Model	AlexNet	VGG16	GoogleNet	ResNet18	DenseNet121	LeNet
AlexNet	**36.9**%	20.6%	25.4%	22.5%	25.5%	12.7%
VGG16	63.4%	**62.6**%	37.9%	42.4%	53.4%	19.5%
GoogleNet	58.3%	57.2%	**56.2**%	59.5%	56.8%	25.3%
ResNet18	57.9%	59.5%	45.9%	**52.6**%	54.2%	22.4%
DenseNet121	67.1%	50.9%	38.5%	43.7%	**58.5**%	16.0%
LeNet	22.0%	21.1%	28.2%	24.1%	26.2%	**16.6**%

4.3 Resiliency of Attack under Defense

There are many different defense strategies in the literature but most are not robust to attacks [3]. One recently proposed defense strategy is APE-GAN [21], which needs no knowledge of the parameters of the target model and is used as a preprocessing step to remove adversarial perturbations from an image. This defense was found to be reasonably effective against Carlini-Wagner and FGSM attacks [21].

This defense is also quite effective against our attack[2], more so than against FGSM and Carlini-Wagner: a VGG16 classifier with APE-GAN defense achieves a test accuracy of 67.3% on CIFAR10 when subject to a white-box attack from a GATN, compared to an accuracy of 40.5% for an FGSM attack and 13.0% for a Carlini-Wagner attack. These are all strong improvements over an undefended VGG16 classifier (see Figure 3a), but the improvement is particularly acute for GATN attacks. We speculate that the perturbations generated by our attack are more easily detected by a defense that uses a similar architecture. This is a major limitation of the current work. Making the GATN attack resilient against the APE-GAN defense, as well as evaluating the attack against other common defenses, will be left for future work.

[2]We used the implementation of APE-GAN by [3] at https://github.com/carlini/APE-GAN

4.4 Limitations

We have discussed the limitations of our work in detail throughout this section. Here, we briefly re-iterate the three major limitations and the subsections in which they were discussed:

- FGSM outperforms the GATN attack on simpler classifiers like LeNet (4.1)
- black-box GATN attacks are not transferable to simpler neural networks (4.2)
- GATN attacks are more easily defended by APE-GAN than FGSM or Carlini-Wagner (4.3)

5 Conclusion

We have introduced a novel adversarial attack whose benefits are threefold: (1) it is more effective than FGSM for attacking very deep neural networks; (2) it is faster than both FGSM and Carlini-Wagner; (3) it is highly transferable to other DNNs via black-box attacks. Our attack is crafted through jointly training a discriminator D and a generator G that we call a Generative Adversarial Transformation Network (GATN). G is responsible for learning to create image-conditional adversarial perturbations that can fool D. Once trained, G can be used to conduct white-box attacks against D and black-box attacks against other classifiers D'. We found that black-box attacks on D' were as effective as white-box attacks when both D and D' had a very deep neural network architecture (e.g., ResNet). When the classifier being attacked has a simple architecture (e.g., LeNet), black-box attacks are not very effective and white-box attacks are slightly less effective than FGSM. We provided some theoretical explanations for this behavior, building on Goodfellow et al.'s hypotheses [6]. We also found that adversarially training D with G makes it more robust to attacks from G and, to a lesser extent, to attacks from FGSM and Carlini-Wagner. We leave comprehensive analysis for the resiliency of our attacks against defenses as future work. In short, we found that our approach is, in several respects, better than FGSM as a baseline untargeted attack for DNNs.

Contributions: Kawin: Introduction, Related Work (except 2.4), Approach, design of GATN attack, implementation of GATN, FGSM, and Carlini-Wagner attacks; **Romina**: Results, Discussion, and Limitations, experiments with attacks in white-box and black-box settings; tests against APE-GAN ; **Joey**: Abstract, GAN description (2.4), Conclusion, Approach implementation, architecture design of GATN, training and evaluation of GATN white-box + black-box attacks; **Mohammad:** experiments with model hyperparameters

References

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

[2] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017.

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

[4] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

[5] 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.

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

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

[8] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

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

[10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[11] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

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

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 2015.

[15] Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387*, 2017.

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

[17] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[18] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[19] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.

[20] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[21] Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan.

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

[23] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] 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.