Disrupting Image-Translation-Based DeepFake Algorithms with Adversarial Attacks

Yeh, Chin-Yuan Chen, Hsi-Wen Tsai, Shang-Lun Wang, Shang-De
National Taiwan University
r06921105, r06921045, r07921059, sdwang@ntu.edu.tw

Abstract

DeepNude, a deep generative software based on image-to-image translation algorithm, excelling in undressing photos of humans and producing realistic nude images. Although the software was later purged from the Internet, image translation algorithms such as CycleGAN, pix2pix, or pix2pixHD can easily be applied by anyone to recreate a new version of DeepNude. This work addresses the issue by introducing a novel aspect of image translating algorithms, namely the possibility of adversarially attacking these algorithms. We modify the input images by the adversarial loss, and thereby the edited images would not be counterfeited easily by these algorithms. The proposed technique can provide a guideline to future research on defending personal images from malicious use of image translation algorithms.

1. Introduction

While deep learning has led to many inspiring breakthroughs in recent years, this new technology can be easily misguided, as well as misused. On the one hand, classification models are easily fooled by adversarial examples that are only slightly perturbed versions of the regular data [1, 2], leading to vulnerabilities in deep learning-based applications [3, 4]. On the other, the resolution and quality of images produced by generative models have seen rapid improvement recently. This gives rise to immoral deep learning software [5], *i.e.*, deepfake, which has already set multiple precedents of fake news [6, 7] and fake pornographic images [8, 9], threatening privacy and security. One of the most notorious deepfake applications, DeepNude [10], is based on image-to-image translation technique. The function of DeepNude is simple: input an image and generate the naked version of the image with a single click. consequence is catastrophic: anyone could now find themselves a victim of revenge porn. Although it was pulled offline shortly after the attention [11], the source codes had been released, and thus the same algorithm can be reproduced easily to this date.

Facing the threat of deepfake algorithms, many, including Facebook AI [12], have placed efforts into finding forensics detection methods to detect deepfake contents. However, these detection methods focused on face-swapping techniques [13, 14], and thus are not suitable for DeepNude, which affects different areas of an image (and not the *face*). Furthermore, even if future detection methods catch the footprints of DeepNude, it still causes harm to the individuals in the falsely generated images. This situation necessitates the demand for a more direct intervention to protect personal images from being easily manipulated by deep generative algorithms. As deepfake models harm our confidence in presenting our images online, and classification models err upon adversarial images, we began to wonder: can we obstruct the misuses of deep generative models by misguiding them through adversarial perturbations? Following this idea, we tackle the problem with a new approach, utilizing adversarial attacks to create imperceptible perturbations that would cause deep generative algorithms to fail in generating the fake image in the first place.

Research on adversarial attacks was rarely applied on generative functions [15], and to our best effort, our work is the first to attack image translation GANs at inference time. Naively, seeing that attacks on classification models often utilizes the original model loss as the adversarial loss, one might jump to the conclusion that adversarial attacks on GANs should take the corresponding Discriminator into account. However, as we shall see in Section 5.1, this approach is futile. In addition, we also find image translation GANs robust against inputs added with random noise. Thus, achieving a successful adversarial attack on GANs is a challenging problem.

Our goal for attacking GANs is clear: to cause an image translation GAN model to fail in converting an image to the model's designed outcome. With extensive experiments, we condense the term *fail* to two concrete and plausible definition: to output a similar or unmodified version of the input image, or to output a broken and disfigured image. In the first case, we introduce *Nullifying Attack*, which minimizes the distance between the adversarial output and the original input, thus causing the model to output a similar image of the original image. For the second case, we present *Distorting Attack*, which maximizes the distance between the adversarial output and the original output, causing the model to generate an image distorted away from the original photo-realistic image, resulting in a blurred and distorted output, unrecognizable as a portrait picture and can be easily identified as fake.

Furthermore, we also propose two novel metrics, *i.e.*, the *similarity score* to evaluate *Nullifying Attack*, and the *distortion score* to evaluate *Distorting Attack*. The *similarity score* increases when attacking with a lower degree of adversarial perturbation, as well as having the output closer to the original input. The *distortion score* is higher when the attack distorts the output more than it perturbs the input. To our best knowledge, we are the first to evaluate the adversarial attack on GAN numerically.

The contributions of this work include:

- Two types of adversarial attack on image-to-image translation models, namely, the *Nullifying Attack* and the *Distorting Attack*.
- Two novel metrics, namely, the *similarity score* s_{sim} and the *distortion score* s_{dist} created for the evaluation of the two types of attack methods respectively.

[1]

2. Related Work

Previous research on adversarial attacks had mainly focused on classification models [1, 2, 4, 16, 17] and paid less attention to generative models [15, 18]. While VAE appeared as a means of defense against adversarial attacks in the prior work [19], Tabacof *et al.* [15] conjectured that VAE could itself be vulnerable. They validated this point by misguiding the model to reconstruct adversarial images to selected images. Kos *et al.* [18] motivated the attack by depicting the scenario of using VAEs as a compression device. Besides attacking the latent vector and the final output, they also added a classifier to the latent vector to utilize adversarial attacks on classification models.

Another line of studies utilized the generative model to defend [20, 21] or enhance [22, 23] adversarial attacks on classification models in previous literature. There are some efforts to produce out-domain examples for GANs with noise input [24] and to corrupt the training of image-to-image deep generative models [25]. Compared with the above research, we are the first to investigate and succeed in attacking fully trained image-to-image deep generative models at inference time.

3. Methodology

Our goal is to perform successful adversarial attacks on image translation models. In this section, we first briefly introduce our target models. We then introduce our attacking framework, *i.e.*, PGD attack. Finally, we describe the adversarial losses to be implemented in our attack.

3.1. Image-to-Image Translations

GAN [26] is a deep generative algorithm consisting of two deep learning networks, *i.e.*, the Generator G and the Discriminator D, contesting in the minimax game,

$$\begin{aligned}\min_G \max_D V(D,G) = {} & \mathbb{E}_{\boldsymbol{x}}[\log D(\boldsymbol{x})] \\ & + \mathbb{E}_{\boldsymbol{z}}[\log(1 - D(G(\boldsymbol{z})))].\end{aligned} \tag{1}$$

Where given a training set x, the Discriminator learns to differentiate between samples $G(\boldsymbol{z})$ generated from noise $\boldsymbol{z}$ and real samples $\boldsymbol{x}$, while the Generator tries to fabricate samples that are indistinguishable from the real. One of the most well-known applications, image translation, learns a mapping, *i.e.*, $x \to y$ between two image domains x and y.

For paired datasets, pix2pix [27] and pix2pixHD [28] learn the mapping between paired image by *conditional* GAN, where by feeding in both x and y, the Discriminator can ensure a pixel-to-pixel translation. This can be formally written as:

$$\begin{aligned}\min_G \max_D V(D,G) = {} & \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x},\boldsymbol{y})] \\ & + \mathbb{E}_{\boldsymbol{x}}[\log(1 - D(\boldsymbol{x}, G(\boldsymbol{x})))].\end{aligned} \tag{2}$$

It is worth noting that pix2pixHD is an improved version of the pix2pix, utilizing a coarse-to-fine scheme for its Generator by adding downsampling and upsampling layers, and using multiple scaled Discriminators to significantly improve the image quality.

While it is costly to prepare paired datasets in practice, CycleGAN [29] can work on unpaired datasets. It uses two sets of GANs, in which two Generators transform the images from both domains, *i.e.*, $G_x : x \to y$ and $G_y : y \to x$, and two Discriminator D_x and D_y learn to distinguish between x and $G_Y(y)$ as well as between y and $G_x(x)$. Moreover, by utilizing the *cycle consistency loss*

$$\begin{aligned}L_{cyc}(G_x, G_y) = {} & \mathbb{E}_x[||G_y(G_x(x)) - x||_1] \\ & + \mathbb{E}_y[||G_x(G_y(y)) - y||_1]\end{aligned} \tag{3}$$

CycleGAN can ensure transitivity, that is, image transferred by both Generators consecutively would be similar to the original image, and thereby it does not require the two domains x and y to be paired.

3.2. Projected Gradient Descent Attack

Szegedy *et al.* [1] first brought to attention that deep learning models can be misled with imperceptible perturbations, now known as “adversarial attacks.” The current

[1] source code provided in: https://github.com/jimmy-academia/Adversarial-Attack-CycleGAN-and-pix2pix

state-of-the-art attacking scheme is Projected Gradient Descent Attack (PGD) [17], which can be written as:

$$\begin{aligned} \boldsymbol{x}_0^* &= \boldsymbol{x} + noise, \\ \boldsymbol{x}_{t+1}^* &= clip(\boldsymbol{x}_t^* + \alpha \cdot sign(\Delta_x L_{adv}(\boldsymbol{x}_t^*))) \end{aligned} \tag{4}$$

where $\boldsymbol{x}$ is the original example, $\boldsymbol{x}_i^*$ is the adversarial example at the i_{th} iteration, $\Delta_{\boldsymbol{x}} L_{adv}(\boldsymbol{x}_t^*)$ is the gradient of the adversarial loss function L_{adv} w.r.t $\boldsymbol{x}$. α is the adjusting rate, $clip()$ denotes clipping $\boldsymbol{x}_{t+1}^*$ within the norm bound $(\boldsymbol{x}+\epsilon, \boldsymbol{x}-\epsilon)$ and the valid space $(0,1)$, and $noise$ is random noise within ϵ bound.

The adversarial loss function L_{adv} for classification models is often constructed with the model's original classification output [1, 2], which represents the models' confidence of classifying the input image to each label. The adversarial attack process optimizes the adversarial input $\boldsymbol{x}^*$ to *increase* the adversarial loss. Thus, we can cause the model to decrease its confidence in the original (correct) answer by pairing the output to the correct label, multiplied by -1, or increase its confidence in some incorrect answer by pairing the output to the incorrect label.

While Madry *et al.* [17] identified that PGD is the strongest attack utilizing only gradients of adversarial loss, we incorporate PGD as our attacking framework. Our procedures are the same as Equation 4 with L_{adv} replaced with different adversarial loss alternatives.

3.3. Adversarial Losses

As attacks on classification models utilize the model loss, we take the corresponding Discriminator into account, creating

$$L_D(\boldsymbol{x}_t^*) = -1 \cdot D(G(\boldsymbol{x}_t^*)) \tag{5}$$

where D, G is the corresponding Discriminator and Generator function in the target model. We expand on the idea of using discriminative models as an adversarial loss function. Since a trained Generator transfers images in the direction $x \rightarrow y$, the gradient of a Discriminator loss would possibly be best if it points in the opposite direction $y \rightarrow x$. To this end, we train another Discriminator D' with the objective to minimize $D'(x) - D'(y)$, such that D' exhibits $D'(x) < D'(y)$, creating the adversarial loss

$$L_{D'}(\boldsymbol{x}_t^*) = D'(G(\boldsymbol{x}_t^*)) \tag{6}$$

As we shall see in Section 5.1, both attempts fail to provide satisfying results. However, we find that we are able to influence the result by directly applying distance functions to the Generator outcome. In particular, with certain distance function $\mathcal{L}$, we define the adversarial loss function for *Nullifying Attack* as,

$$L_{Null}(\boldsymbol{x}_t^*) = -1 \cdot \mathcal{L}(G(\boldsymbol{x}_t^*) - \boldsymbol{x}) \tag{7}$$

and the loss function for *Distorting Attack* as,

$$L_{Dist}(\boldsymbol{x}_t^*) = \mathcal{L}(G(\boldsymbol{x}_t^*) - G(\boldsymbol{x})) \tag{8}$$

By applying distance functions, we can guide the output towards a certain desired direction. In the case of *Nullifying Attack*, the objective is to cause the image translation model to output the original input. Thus Equation 7 is set so that the distance between the adversarial output and the original input would be *minimized*. *Distorting Attack*, on the other hand, has the objective to push the adversarial output away from the original output. Therefore, Equation 8 is set so the distance between the two would be *maximized*.

4. Implementation

Following the original works [27, 28, 29], we use 9-blocks ResNet in the Generators for CycleGAN, Unet for pix2pix, and the combination Unet and further upsampling and downsampling layers for pix2pixHD. 70 × 70 PatchGAN architecture is used in all the Discriminators. We train by stochastic gradient descent (SGD) with Adam [30] with batch size 1 and the learning rate set to 0.0002 for the first 100 epochs then linearly decayed to 0 over the next 100 epochs. For a consistent result, we evaluate the proposed method on all three model types trained with the CelebA-HQ dataset [31] and the corresponding mask dataset CelebAMask-HQ [32]. Notice that we load the image at 286 × 286 than randomly cropping to 256 × 256 for CycleGAN and pix2pix, and loading at 572 × 572 than randomly cropping to 512 × 512 for pix2pixHD. For adversarial attack procedures, the default norm bound ϵ, adjust rate α, and attack iteration are 0.2, 0.01, and 100, respectively. We use $\mathcal{L}(x) = x^2$ as the default distance function for Equations 7 and 8. We randomly sample 90% of images for training and 10% of images for testing, and the average results from 50 runs are reported. [2] [3]

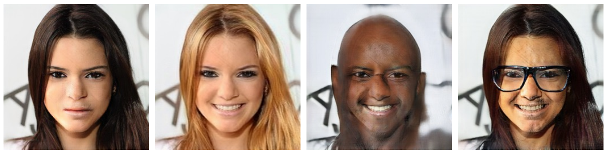
5. Experiments

In this section, we first present the quantitative analysis of different attacking schemes. Then, we introduce two

[2] For CycleGAN, we select two groups of images out of the CelebA-HQ dataset using "Smiling," "HairColor," "Bald" and "Eyeglasses" attributes to create four image domain pairs and train model SMILE that translates smiling to frowning images, model BLOND that translates black hair to blond hair, model BALD that transforms a person with hair to a bald figure, and model GLASS that adds eyeglasses to the figures. The attributes are selected to reflect manipulation of expression, replacement of an area, removal of parts and addition of elements to the portrait.

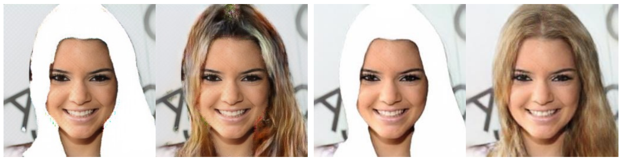
[3] For pix2pix and pix2pixHD, we train model BLOND-PIX and model BLOND-PIXHD having the same functionality as that of model BLOND. Each model consists of a pair of models trained to perform "BlackHair" → "BlackHairMasked" and "BlondHairMasked" → "BlondHair" image translation tasks. The intermediate masked images are created by replacing the hair region with a white mask using the corresponding hair mask images from CelebAMask-HQ.

Figure 1: An image from the CelebaHQ dataset selected as the running example.



(a) SMILE (b) BLOND (c) BALD (d) GLASS

Figure 2: Resulting images from feeding the running example to the CycleGAN models shows the models all work as expected.



(a) BLOND-PIX (b) BLOND-PIXHD

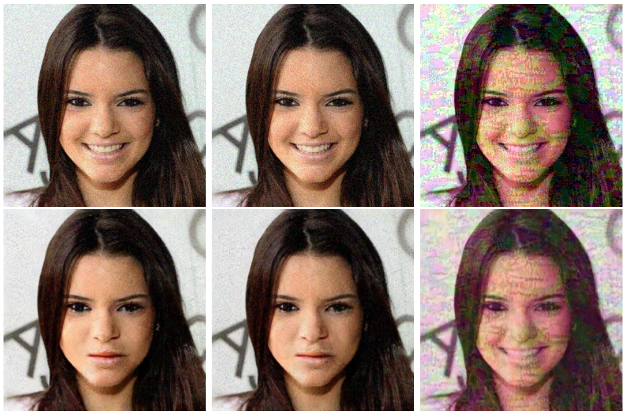
Figure 3: Pix2pix and pix2pixHD results, including masked outputs (left) and final results (right), showing the models all work as expected

novel metrics, the *similarity score* and the *distortion score*, based on the two attacks to give a concrete evaluation. Sensitivity tests are also presented.

5.1. Quantitative Results

Taking Figure 1 as our running example, we present outputs from our CycleGAN models (model SMILE, BLOND, BALD and GLASS) in Figure 2 as well as the intermediate masked image and final output for model BLOND-PIX and BLOND-PIXHD in Figure 3.

In Figure 4, we find that neither adding random noise or using naive adversarial losses constructed with Discriminators properly effect the outcome. On the one hand, using the original Discriminator (Equation 6) in adversarial attack shows poor results because the Generator and the Discriminator evolve simultaneously in Equation 5 and the Discriminator only incrementally changes for the Generator to follow [26]. Once training is complete, the gradient derived from the Discriminator would supposedly only point towards the subtle differences between real samples and generated examples that are realistic. On the other hand, reversely trained Discriminator (Equation 6) only focused on the translated image attribute such that it doesn't consider the quality of the input and output images, and thus the output image retains the smile but is also spotted with an oil-like iridescent color.



(a) random (b) D (c) D'

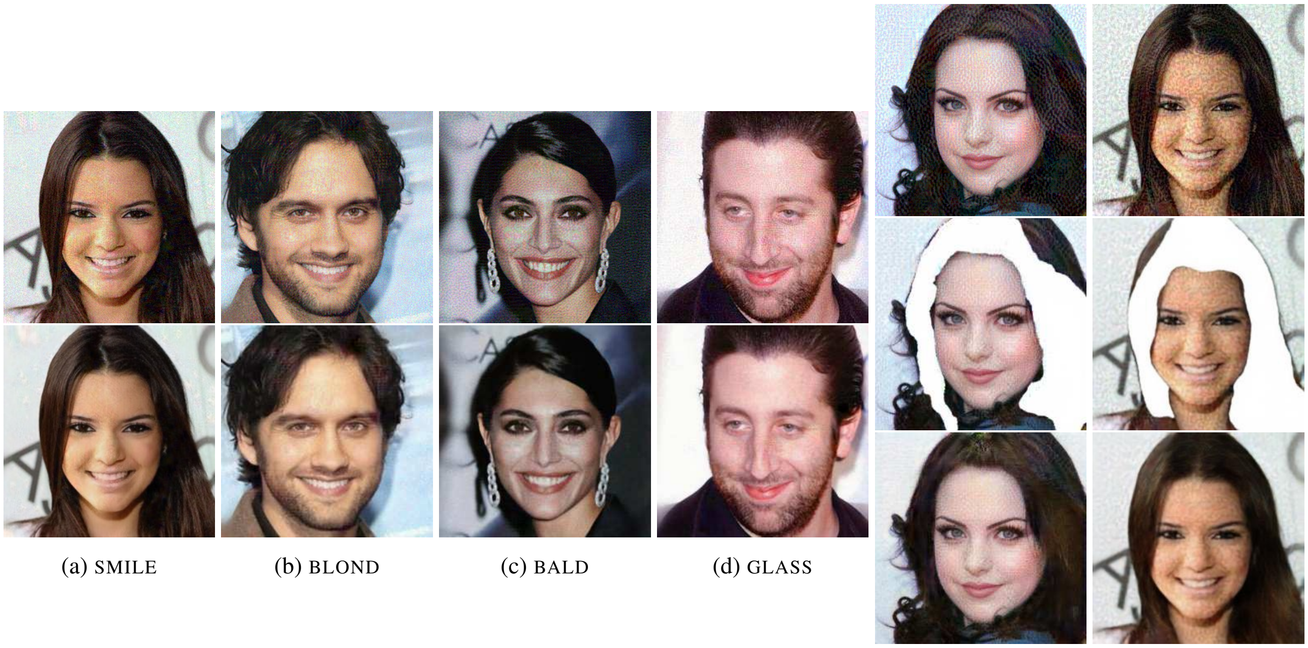
Figure 4: Adversarial inputs and outputs for adding random noise, attacking with L_D and L'_D as adversarial loss on the running example for model SMILE shows ineffective or poor results.

In contrast, *Nullifying Attack* (Equation 7) and *Distorting Attack* (Equation 8) both show great results in all our models, as shown in Figures 5 and 6. *Nullifying Attack* consistently causes the Generator to output an image similar to the original input. Moreover, the perturbations in the adversarial input are translated back to a smooth and photorealistic background most of the time. *Distorting Attack* also successfully distorts the outcomes of CycleGAN models dramatically, and causes pix2pix and pix2pixHD to fail in the second (masked image → image) translation.

Depending on different considerations, one might find one of *Nullifying Attack* and *Distorting Attack* better than the other. For example, if the goal is to maintain image integrity such that the correct image may be delivered, one can resort to *Nullifying Attack*. Alternatively, if the goal is to detect the usage of image translation algorithms, *Distorting Attack* could lead to more dramatic visual changes which can be spotted easily.

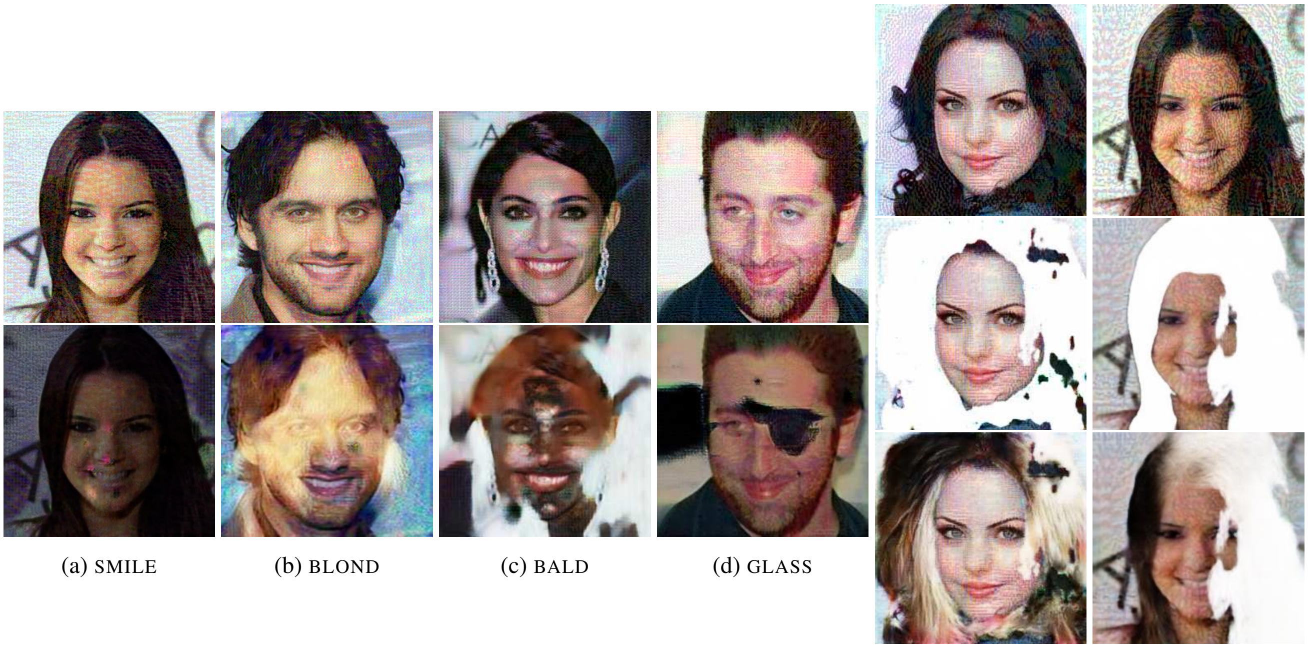
5.2. Similarity and Distortion Scores

In previous research [15], result of adversarially attacking VAEs were evaluated by plotting the distance measures of adversarial perturbation (*i.e.*, the distance between the original input and the perturbed input) as well as the distance between the adversarial output and the target image. Following this approach, we introduce the *similarity score* for evaluating the performance of *Nullifying Attack* and the



(a) SMILE (b) BLOND (c) BALD (d) GLASS (e) BLOND-PIX (f) BLOND-PIXHD

Figure 5: Nullifying Attack results, with adversarial inputs on top, (intermediate result in the middle) and adversarial outputs below. Different images are selected along with the running example to show the generalizability of the proposed method.



(a) SMILE (b) BLOND (c) BALD (d) GLASS (e) BLOND-PIX (f) BLOND-PIXHD

Figure 6: *Distorting Attack* results, with adversarial inputs on top, (intermediate result in the middle) and adversarial outputs below. Different images are selected along with the running example to show the generalizability of the proposed method.

distortion score for *Distorting Attack*. With $\boldsymbol{x}$ and $\boldsymbol{y}$ as the original input and output, $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ as the perturbed input and output, and some distance function $\mathcal{L}$, the *similarity score* can be written as:

	MODEL TYPE					
loss	SMI.	BLO.	BALD	GLA.	PIX.	HD
D	0	.15	.18	0	.09	.16
D'	0	.08	.11	0	0	.1
Null.	**.02**	**.36**	**.41**	**.08**	**.27**	**.34**
Dist.	0	.06	.06	0	0	.02

Table 1: The s_{sim} values for different adversarial loss and model type. Top score for each model is in bold font, indicating *Nullifying Attack* as best method in this scenario. SMI., BLO., BALD, GLA., PIX., HD are shorthands for model SMILE, BLOND, BALD, GLASSES, BLOND-PIX, BLOND-PIXHD.

$$s_{sim} = max(0, \frac{(\log \mathcal{L}(\boldsymbol{y}-\boldsymbol{x}))^2}{\log \mathcal{L}(\boldsymbol{y}^*-\boldsymbol{x}) \cdot \log \mathcal{L}(\boldsymbol{x}^*-\boldsymbol{x})} - 1) \quad (9)$$

and the *distortion score* is:

$$s_{dist} = max(0, \frac{\log \mathcal{L}(\boldsymbol{y}^*-\boldsymbol{y})}{\log \mathcal{L}(\boldsymbol{x}^*-\boldsymbol{x})} - 1) \quad (10)$$

The scores s_{sim} and s_{dist} are formulated using the Target Distance (*i.e.* the distance between the adversarial output and the original input or output, following Equations 7 and 8, and the Adversarial Distortion (*i.e.* the distance between adversarial perturbed image and the original image) to highlight the objective of nullifying the image translation effects or distorting the outcomes respectively, while also taking account the objective of limiting the degree of perturbation. It follows naturally whether to place each distance in the numerator or denominator, such that the resulting ratio would have larger values for better results. For the *similarity score* s_{sim}, it remains that we add a constant distance $\mathcal{L}(\boldsymbol{y}-\boldsymbol{x})$ (the original manipulation of the model) squared to the numerator so as to arrive at a dimensionless quantity.

Since humans perceive change logarithmically [33], we add log scales to the distances. Finally, we set up the rest of Equations 9 and 10 so that attack that fails to keep it closer to the original input than the original output would find $s_{sim} = 0$, whereas attacks that fail to distort the output more than the perturbation made on the input would have $s_{dist} = 0$. Taking $\mathcal{L}(x) = x^2$ as our distance function again, we find clear cut evidence that *Nullifying Attack* and *Distorting Attack* are best methods of choice for each objective, as each attack results in the highest score for every model in Tables 1 and 2 respectively.

	MODEL TYPE					
loss	SMI.	BLO.	BALD	GLA.	PIX.	HD
D	0	.03	.03	0	0	.09
D'	0	.04	.07	.01	.05	.04
Null.	0	.13	.14	.02	.09	.12
Dist.	**.16**	**.16**	**.20**	**.14**	**.17**	**.15**

Table 2: The s_{dist} values for different attack methods and models. Top value for each model is in bold font, indicating *Distorting Attack* as best method in this scenario. Shorthand notations follows Table 1

5.3. Sensitivity Tests for Error Bound ϵ

Tabacof *et al.* [15] reported that for attacks on VAE, there is a quasi-linear trade-off between the adversarial perturbation at the input and the intended adversarial results. However, this is not the case for image translation GANs, as we find that adjusting the norm bound ϵ can lead to abrupt changes. In Figures 7 and 8, we plot the Target Distance against the Adversarial Distortion for 100 equally spaced values of ϵ in $[0, 0.5]$ for *Nullifying Attack* and *Distorting Attack* on the CycleGAN models as a motivating example.

Nullifying Attack show different behaviour for different trained models. We suspect that this is because the attack process pulls the output towards the original image. For some models (*e.g.* model SMILE), the original image translation manipulation is small, so a small adversarial perturbation is enough to reach the original image, and further adversarial overflows to larger distortion. Although there is a larger distortion in the adversarial output with larget ϵ value, visually accessing the output image finds that image translation effect is still nullified and the quality of image acceptable. We display in Figure 9 the output image of several ϵ values for model SMILE, including $\epsilon = 0.495$ which corresponds to the maximum value in Figure 7a.

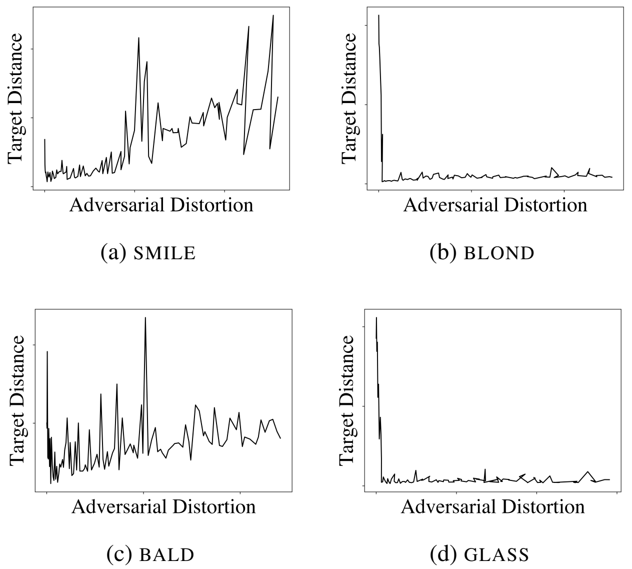
Distorting Attack, on the other hand, shows a more stable trend which saturates towards large adversarial distortions. This is because the attack process pushes the output away from a starting point (the original output) and can continue indefinitely. The saturation trend may arise from inherent robustness of GANs.

6. Case Study

In this section, we first examine results from using different options of distance functions $\mathcal{L}$. Then, we evaluate whether *Nullifying Attack* results can withstand being manipulated again by the same translation model. Finally, we validate the effectiveness of proposed methods for attacking multiple models simultaneously with an ensemble attack.

6.1. Comparison of Distance Functions

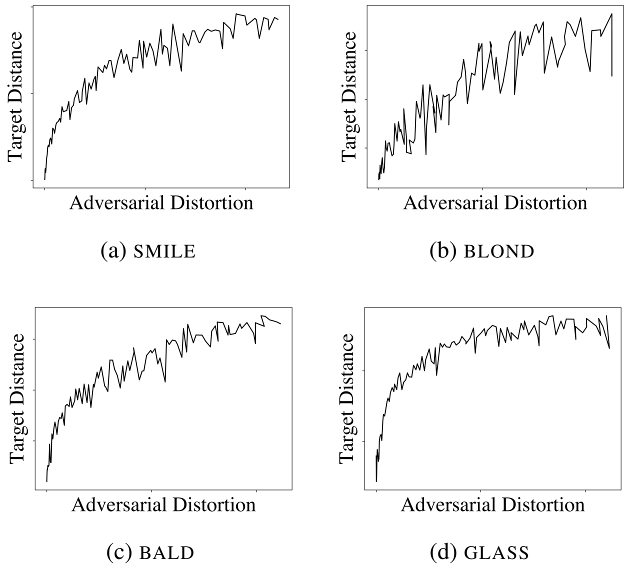
We conduct extensive experiments on different distance functions. Out of ℓ_1, ℓ_2, ℓ_3, ℓ_∞, as well as x^2, $|x^3|$, x^4, $|x^5|$, we find $\mathcal{L}(x) = x^2$ to work the best. We report that Cy-



(a) SMILE (b) BLOND

(c) BALD (d) GLASS

Figure 7: Plotting the Target Distance against Adversarial Distortion for the four CycleGAN models shows that *Nullifying Attack* is highly non-linear and the behaviour varies greatly between different models.

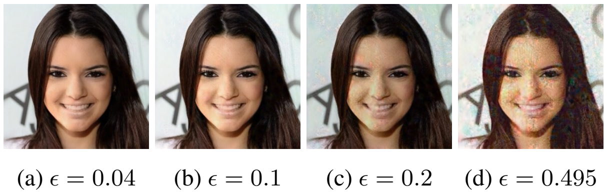


(a) SMILE (b) BLOND

(c) BALD (d) GLASS

Figure 8: Plotting the Target Distance against Adversarial Distortion for the four CycleGAN models for *Distorting Attack* shows a saturating effect.

cleGAN models are easier to attack than pix2pix and only using ℓ_1 distance fails for model BLOND (Figure 10). In the case of pix2pix models, $\ell_2, \ell_3, \ell_\infty$ norms are too weak to effect the outcome (Figure 11), while the effect of perturbation are too strong for $n > 2$ in x^n (Figure 12). This result supports our using x^2 as the default distance function.



(a) $\epsilon = 0.04$ (b) $\epsilon = 0.1$ (c) $\epsilon = 0.2$ (d) $\epsilon = 0.495$

Figure 9: Example *Nullifying Attack* results on model SMILE for various ϵ values.

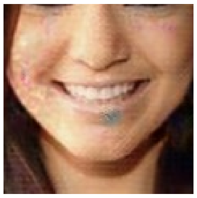
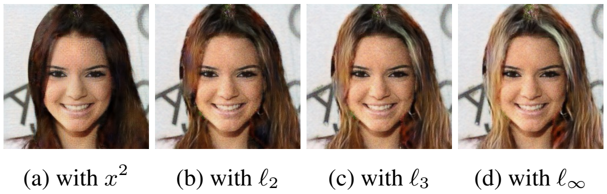
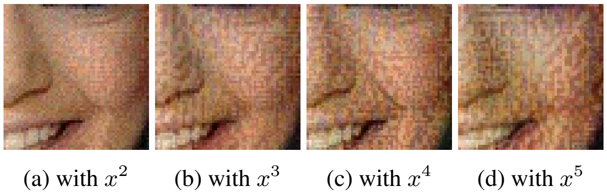


Figure 10: *Nullifying Attack* result with ℓ_1 on model BLOND shows a green spot on the lower lip.



(a) with x^2 (b) with ℓ_2 (c) with ℓ_3 (d) with ℓ_∞

Figure 11: *Nullifying Attack* results with different distance functions on model BLOND-PIX. Compared with x^2, using ℓ_2, ℓ_3 and ℓ_∞ fails to prevent the hair color from changing



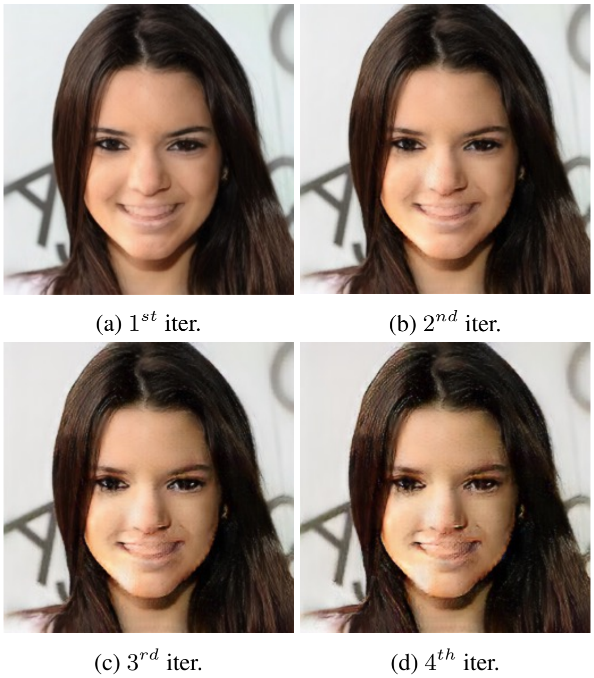
(a) with x^2 (b) with x^3 (c) with x^4 (d) with x^5

Figure 12: Enlarged view of left cheek area for *Nullifying Attack* inputs with different distance functions on model BLOND-PIX. Compared with x^2, using x^3, x^4 and x^5 perturbs the image significantly more.

6.2. Repeated Inference for Nullifying Attack Results

As *Nullifying Attack* results in an image similar to the original input, we are curious to see whether the image translation model could manipulate *Nullifying Attack* results.[4] Figure 13 shows an example of passing the *Nullifying Attack* result through model SMILE four times consecutively. We find that the image does not convert to a frowning

[4] *Distorting Attack* disfigures the output, so feeding the output back to the image translation model would not amount to much.



(a) 1^{st} iter.

(b) 2^{nd} iter.

(c) 3^{rd} iter.

(d) 4^{th} iter.

Figure 13: Sequence of outputs resulting from repeatedly feeding the outputs back through model SMILE starting with the *Nullifying Attack* result on model SMILE. The image resists being converted to a frowning image.

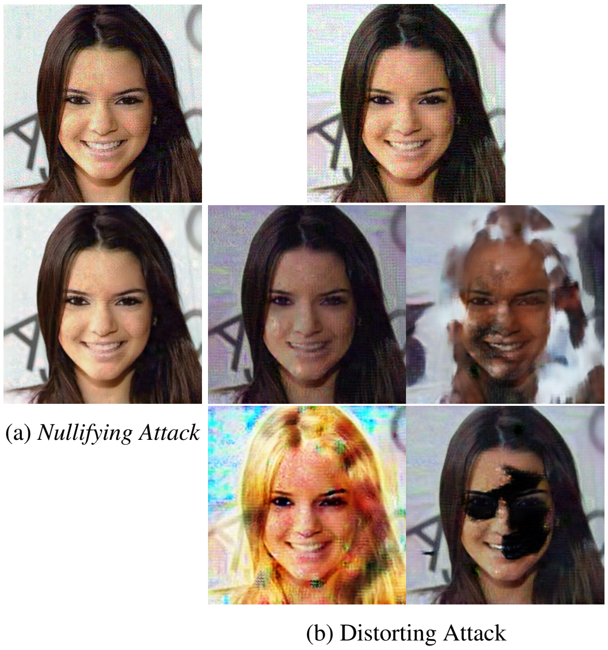
image, indicating that the result of *Nullifying Attack* maintains the original attributes even after multiple tries. We notice that insignificant imperfections in one image translation process accumulate and cause the image to deteriorate in image quality after several iterations.

6.3. Ensemble Attack

To deal with multiple possible deepfake algorithms, we attempt to construct an ensemble attack with loss function written as:

$$L_{ensemble}(\boldsymbol{x}_t^*) = \sum_{m \in \{models\}} L_m(\boldsymbol{x}_t^*) \qquad (11)$$

where L_m are the loss functions, with G in each loss function replaced to G_m. Simply put, the same perturbation steps for each model are now mixed together evenly to create a common adversarial example. We investigate the effectiveness of ensemble attack for model SMILE, BLOND, BALD, GLASS. In Figure 14, *Nullifying Attack* achieves consistent result under the ensemble scheme. However, for *Distorting Attack*, the results are not as distorted as those in Figure 6. We believe this indicates that image translation GANs inherently have similar latent structure, such that the perturbation effect can be more coherent when the target



(a) *Nullifying Attack*

(b) Distorting Attack

Figure 14: Ensemble attack results. The adversarial input (1 on top) and result (1 or 4 at the bottom) for *Nullifying Attack* and Distorting Attack. The four image results for *Nullifying Attack* are all similar to each other, so we only place one.

is the same (*i.e.* the original image for the *Nullifying Attack*) but displays cancellation effect for *Distorting Attack* because the distortion directions are different.

7. Conclusions

The emergence of deepfake applications is a serious ethical issue for research in deep generative algorithms. Past efforts focused on the detection of deepfake generated content but had not thought of the prospect of a more direct means of intervention. In this work, we introduce a novel idea of adversarially attacking image translation models, opening up the doorway to disrupting current or future image translation-based deepfake algorithms directly. We demonstrate that with appropriate adversarial loss functions, one could cause image translation models to be *nonfunctional* as well as *dysfunctional*. We propose the *similarity score* and *distortion score* for evaluating the two types of adversarial attacks, confirming our observations in a more concrete sense. Although conducting various experiments, we believe much work is still needed before we can attain a reliable way to protect our images from malicious use of deep generative models. Future works may include investigation on stronger attack methods that are not necessarily norm bounded, (*e.g.*, utilize deep generative algorithms [22, 23] or be localized in a patch [34]), on the defensive end for image translation models, and on black-box attack methods.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks. iclr, abs/1312.6199, 2014," 2014. 1, 2, 3

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples. corr (2015)," 2015. 1, 2, 3

[3] B. Biggio, P. Russu, L. Didaci, F. Roli *et al.*, "Adversarial biometric recognition: A review on biometric system security from the adversarial machine-learning perspective," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 31–41, 2015. 1

[4] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018. 1, 2

[5] M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar *et al.*, "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation," *arXiv preprint arXiv:1802.07228*, 2018. 1

[6] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6. 1

[7] M.-H. Maras and A. Alexandrou, "Determining authenticity of video evidence in the age of artificial intelligence and in the wake of deepfake videos," *The International Journal of Evidence & Proof*, vol. 23, no. 3, pp. 255–262, 2019. 1

[8] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, "Deep learning for deepfakes creation and detection," *arXiv preprint arXiv:1909.11573*, 2019. 1

[9] D. Lee, "Deepfakes porn has serious consequences," Feb 2018. [Online]. Available: https://www.bbc.com/news/technology-42912529 (Accessed 2019-12-09). 1

[10] github/lwlodo, "Official deepnude algorithm source code," Jul 2019. [Online]. Available: https://github.com/lwlodo/deep_nude/tree/a4a2e3fb83026c932cf96cbecb281032ce1be97b (Accessed 2019-12-11). 1

[11] T. Telford, "'the world is not yet ready for deepnude': Creator kills app that uses ai to fake naked images of women," Jun 2019. [Online]. Available: https://www.washingtonpost.com/business/2019/06/28/the-world-is-not-yet-ready-deepnude-creator-kills-app-that-uses-ai-fake-naked-images-women/ (Accessed 2019-12-09). 1

[12] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (dfdc) preview dataset," *arXiv preprint arXiv:1910.08854*, 2019. 1

[13] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," *arXiv preprint arXiv:1901.08971*, 2019. 1

[14] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," *arXiv preprint arXiv:1811.00656*, vol. 2, 2018. 1

[15] P. Tabacof, J. Tavares, and E. Valle, "Adversarial images for variational autoencoders," *arXiv preprint arXiv:1612.00155*, 2016. 1, 2, 4, 6

[16] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016. 2

[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017. 2, 3

[18] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models. in 2018 ieee security and privacy workshops (spw)," 2018. 2

[19] M. Willetts, A. Camuto, S. Roberts, and C. Holmes, "Disentangling improves vaes' robustness to adversarial attacks," *arXiv preprint arXiv:1906.00230*, 2019. 2

[20] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-gan: Protecting classifiers against adversarial attacks using generative models," *arXiv preprint arXiv:1805.06605*, 2018. 2

[21] H. Lee, S. Han, and J. Lee, "Generative adversarial trainer: Defense to adversarial perturbations with gan," *arXiv preprint arXiv:1705.03387*, 2017. 2

[22] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," *arXiv preprint arXiv:1710.11342*, 2017. 2, 8

[23] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *arXiv preprint arXiv:1801.02610*, 2018. 2, 8

[24] D. Pasquini, M. Mingione, and M. Bernaschi, "Adversarial out-domain examples for generative models," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2019, pp. 272–280. 2

[25] S. Ding, Y. Tian, F. Xu, Q. Li, and S. Zhong, "Poisoning attack on deep generative models in autonomous driving." 2

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680. 2, 4

[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134. 2, 3

[28] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8798–8807. 2, 3

[29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232. 2, 3

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 3

[31] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017. 3

[32] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," *arXiv preprint arXiv:1907.11922*, 2019. 3

[33] L. R. Varshney and J. Z. Sun, "Why do we perceive logarithmically?" *Significance*, vol. 10, no. 1, pp. 28–31, 2013. 6

[34] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017. 8