TYPE I ATTACK FOR GENERATIVE MODELS

Chengjin Sun⋆ *Sizhe Chen*⋆ *Jia Cai*† *Xiaolin Huang*⋆

⋆Department of Automation, Shanghai Jiao Tong University, Shanghai, P.R. China.
†HuaWei Technologies Co.,Ltd, Hangzhou, P.R. China.

ABSTRACT

Generative models are popular tools with a wide range of applications. Nevertheless, it is as vulnerable to adversarial samples as classifiers. The existing attack methods mainly focus on generating adversarial examples by adding imperceptible perturbations to input, which leads to wrong result. However, we focus on another aspect of attack, i.e., cheating models by significant changes. The former induces Type II error and the latter causes Type I error. In this paper, we propose Type I attack to generative models such as VAE and GAN. One example given in VAE is that we can change an original image significantly to a meaningless one but their reconstruction results are similar. To implement the Type I attack, we destroy the original one by increasing the distance in input space while keeping the output similar because different inputs may correspond to similar features for the property of deep neural network. Experimental results show that our attack method is effective to generate Type I adversarial examples for generative models on large-scale image datasets.

Index Terms— type I attack, adversarial examples, generative models

1. INTRODUCTION

Generative models are considered to be one of the greatest inventions in the field of AI. Two most representative types are: the generative adversarial networks (GAN) [1] and the variational autoencoder (VAE) [2]. They have many applications, such as auto-programming [3], compressing information [4], interactive image editing [5, 6], sketch2image [7, 8], and other image-to-image translation tasks [9, 10].

It is now well-known that DNNs are vulnerable to adversarial attacks. In [11], it has been found that imperceptive perturbations on the input of autoencoder cause the reconstruction result to change significantly. In statistical, this attack corresponds to Type II attack on classifiers, i.e., manipulating the input by adding imperceptible perturbations [12, 13, 14, 15] or changing the semantic attributes of images [16, 17, 18], which has attracted many attention of researchers and becomes a big concern. Very recently, we find that it is also possible to implement Type I adversarial attack [19]. Type I attacks on generative models, i.e., changing the input significantly but leading to similar output, is potentially as dangerous as attacks on classifiers and also meaningful to investigate. For instance, Type I attack could do harm to the information transition because autoencoders are widely used for compressing information. A malicious noise image which is far from the clean one may lead to a reconstruction output which is similar to the origin. Type I adversarial examples can also point out the weakness of the generative models and are valuable for enhancing the robustness of the network.

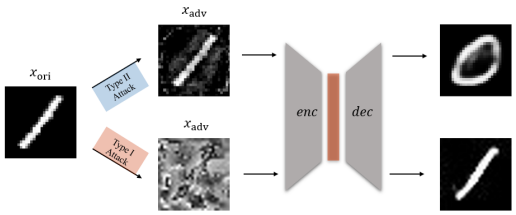

Fig. 1. Type I and Type II adversarial attacks on VAE.

In this paper, we design Type I attack on generative models. The difference between Type I and Type II attacks could be understood by the following example. As illustrated in Fig 1, the top is the Type II attack where we slightly disturb "1" such that the adversarial digit is still "1" but its reconstruction result is another digit "0". Type I attack is shown at the bottom. Although the noisy and meaningless image is fed to the VAE, the generative model outputs a clean example "1".

Mathematically, the Type I attack, i.e., the input is changed significantly but the generator still gets a similar output, is defined as the following,

$$\begin{aligned} \text{From} \quad & x \quad \text{Generate} \quad x' = \mathcal{A}(x) \\ & s.t. \quad G(x') = G(x) \\ & \qquad \|x - x'\| > \epsilon, \end{aligned} \tag{1}$$

where x is the input and x' represents the adversarial example. G denotes the generative models and ϵ is the threshold.

The underlying reasons for Type I attack and Type II attack are different on features as explained in [19]. The exis-

This work was supported by National Key Research Development Project (No.2018AAA0100702) and National Natural Science Foundation of China (No. 61977046).

tence of Type II adversarial examples is due to the unnecessary feature considered by the usual generative model but not used by an ideal one. So the variant in the unnecessary feature ignored by the oracle makes the output of the usual generative model change greatly. Conversely, the missing feature is taken into account in the ideal one but is omitted by the usual one. So attacking missing feature causes Type I attack. The essential difference between both attacks makes the defense designed for Type II do not help the improvement of Type I robustness. Thus, Type I attack should be simultaneously considered together with Type II to strengthen the robustness of generative models.

In this paper, we perform Type I attack on two most representative generative models: VAE and StyleGAN. For VAE, the adversarial image is generated by increasing its distance to the clean one and keeping their outputs similar. Another way to attack is by updating the latent variable which recovers the adversarial image through the decoder. Benefiting from the restriction in the latent space and the gradient related to increasing the distance between the adversarial example and the original, the attack is achieved. The datasets used to perform Type I attack on VAE are MNIST, SVHN, and CelebA. For StyleGAN, the attack is implemented by updating the intermediate latent space which is directly related to the generated images' styles.

The rest of this paper is organized as follows. In Section II, we introduce the techniques of Type I adversarial attack. Section III evaluates the proposed attack on VAE and StyleGAN. In Section IV, a conclusion is given to end this paper.

2. TYPE I ATTACK ON GENERATIVE MODELS

In this paper, we focus on the most representative generative models: VAE and StyleGAN. VAEs are neural networks consisting of an encoder $e(\cdot)$ and a decoder $d(\cdot)$. The encoder outputs the parameters of the latent distribution from the input, and then the decoder samples the latent distribution and reconstructs something similar to the input. StyleGAN [20] is a representative GAN, which has a clear hierarchy of features, can generate ultra-high-resolution samples. StyleGAN is composed of two sub-networks: a non-linear mapping network $f : \mathcal{Z} \to \mathcal{W}$ which maps the latent code z to an intermediate latent code $w = f(z)$, and a synthesis network $Gs(\cdot)$ which starts from a constant, and receives styles from w after affine transformations to control adaptive instance normalization every time before upsampling image. Therefore, the generator of StyleGAN can be represented as $G = Gs \circ f(z)$.

For Type I attack on VAE, it is required to generate an adversarial image which is totally different from the origin. Here, we propose the attack on the image space to generate random noise and keep their reconstruction outputs similar. We also could attack the latent space. That means the gradients do not merely propagate to the image space, but further to the latent space. When attacking the image space, we push the adversarial image away from the original one while minimizing the distance between their reconstruction outputs. Mathematically, the above idea can be described as the following function,

$$L_x = \|d(e(x)) - x_{\text{ori}}\| - \lambda * \|x - x_{\text{ori}}\|, \tag{2}$$

where x_{ori} denotes the original image and x is the input variable to optimize. The first part of the loss function makes sure the similarity of the outputs while the second part destroys the input to a meaningless one. Hyper-parameter λ aims to balance the two parts. Notice that the norm here could be replaced by many distances. In this paper, we use l_1-norm distance when attacking SVHN and CelebA and l_2-norm distance for MNIST.

Also, we can find the adversarial example by searching in the controllable latent space and then decoding it so that the adversarial examples are expected to follow a known distribution. In this way, we can capture more features instead of generating random sharp noise. In the process of attacking the latent space, we enlarge the distance within a specific threshold in the latent space and keep the similarity of the reconstruction outputs simultaneously, as follows,

$$\begin{aligned} L_z = \|d(e(d(z))) - d(e(x_{\text{ori}}))\| - \\ \lambda * ReLU(\varepsilon - \|z - e(x_{\text{ori}})\|), \end{aligned} \tag{3}$$

where z is the latent variable which we need to optimize, $d(z)$ is the adversarial image and ε is a threshold which restricts z on the mainfold of the latent space.

For StyleGAN, instead of revising the latent variable of the mapping network, we choose to optimize in the intermediate latent space because it is disentangled and directly controls the feature of the generation output through learned affine transformations. Therefore, we minimize the following objective function to make the disentangled intermediate latent variable change significantly but the output still similar to the origin,

$$\begin{aligned} L_s = \frac{1}{n_1} \sum \|Gs(w) - Gs(w_{\text{ori}})\| + \\ \lambda * ReLU(\varepsilon - \frac{1}{n_2} \sum \|w - w_{\text{ori}}\|), \end{aligned} \tag{4}$$

where $Gs(w)$ and $Gs(w_{\text{ori}})$ are generated images of StyleGAN which correspond to the adversarial intermediate latent vector w and original vector w_{ori} respectively. n_1 and n_2 are the size of the generated image and the feature vector. Here we choose l_1-norm to optimize for the reason that restriction in every pixel contributes to generating clearer images with more details rather than a fuzzy one.

To show whether the style feature vector changes greatly, we use the following criteria to measure the deviation:

$$Dev = \frac{1}{n} \left\| \frac{w - w_{\text{ori}}}{w_{\text{ori}}} \right\|_2 * 100\%, \tag{5}$$

where n is the size of w_{ori}.

In Eq. (2), (3) and (4), λ reflects the balance between the variation in the input and the similarity in the output. In our method, λ is set to a constant when attacking VAE. For StyleGAN, λ varies for different iterations. At the start of the optimization, we set a large λ to enlarge the change of the input and allow the generated image changing to a totally different one with different features. After that, λ decreases to pull the generated image back so that it is still as same as the original one. Specifically, inspired by [19], a self-adaptive weight strategy is designed for λ to maintain such equilibrium:

$$\begin{aligned}\lambda_{k+1} =& \lambda_k + \alpha(\beta \frac{1}{n_1}\sum \|Gs(w) - Gs(w_{\text{ori}})\| \\ & - ReLU(\varepsilon - \frac{1}{n_2}\sum \|w - w_{\text{ori}}\|)) \\ & + min\{ReLU(\varepsilon - \frac{1}{n_2}\sum \|w - w_{\text{ori}}\|) - \widehat{L}_w, 0\},\end{aligned} \tag{6}$$

where β controls the balance between $\frac{1}{n_1}\sum \|Gs(w) - Gs(w_{\text{ori}})\|$ and $ReLU(\varepsilon - \frac{1}{n_2}\sum \|w - w_{\text{ori}}\|)$ in the iteration process. At the begining, it is set as follows,

$$\beta = \frac{ReLU(\varepsilon - \frac{1}{n_2}\sum \|w - w_{\text{ori}}\|)}{\frac{1}{n_1}\sum \|Gs(w) - Gs(w_{\text{ori}})\|}, \tag{7}$$

where $\widehat{L}_w$ is a loss threshold related to the feature space and this loss term concentrates more on the similarity of the generated images. A larger β means larger diversity in features while a smaller β makes the generated image corresponding to the adversarial input still seem like the origin.

The adversarial samples are generated in an iterative process by minimizing loss function L. The update could be generally described as follows.

Attack on the image space of VAE case:

$$x^{k+1} = x^k - \eta \frac{\partial L_x(x)}{\partial x}\Big|_{x=x^k}. \tag{8}$$

Attack on the latent space of VAE case:

$$z^{k+1} = z^k - \eta \frac{\partial L_z(z)}{\partial z}\Big|_{z=z^k}. \tag{9}$$

Attack on the intermediate latent space of StyleGan case:

$$w^{k+1} = w^k - \eta \frac{\partial L_s(w)}{\partial w}\Big|_{w=w^k}. \tag{10}$$

Specifically, we set $x^0 = x_{\text{ori}}$, $z^0 = e(x_{\text{ori}})$ and $w^0 = w_{\text{ori}}$. By iterative update, we gradually change the variable(x, z, w) until the attack is successful, i.e., the distance between the adversarial input and origin of both generative models reaches a specified threshold ζ and the difference of their outputs can not exceed a threshold ξ.

Fig. 2(a) shows the Type I attack process on the StyleGAN. With the increasing of iterations, the deviation in the feature space is increasing. The generated image of the adversarial feature vector first gets pushed away from the original one and becomes a totally different face. Then it is pulled back to be similar to the origin because of the training strategy. Fig. 2(b) illustrates the adversarial feature changes a lot in every dimension.

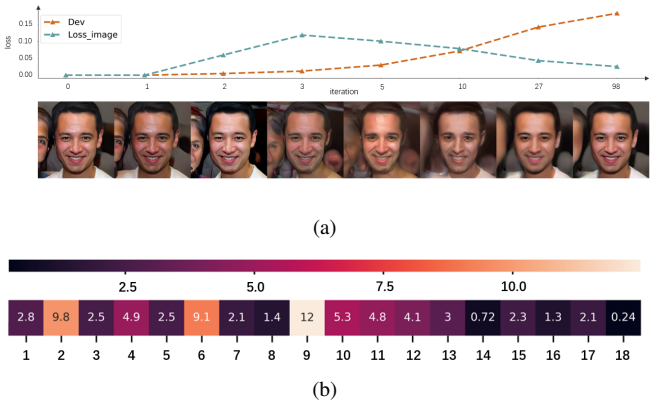

Fig. 2. (a) Type I attack on StyleGAN. The top line shows the deviation of the feature and the loss in the output space. Images of different attack epoch ($0, 1, 2, 3, 5, 10, 27, 98$ from left to right) are displayed at the bottom. (b) Rate of the change in each dimension of the 18-dimensional feature vector.

3. EXPERIMENTS AND RESULTS

In this section, we validate the proposed Type I attack on VAE and GAN to show how the method can change the input significantly but keep the output of the attacked generator unchanged. The datasets we used are MNIST, SVHN, and CelebA. The structure of VAE is modified on [11, 19]. The GAN is the StyleGAN pretrained in FFHQ. The details about the models, attack implementation and attack results are given in the supplemental materials.

3.1. Type I attack on VAE

First, we use the proposed method to attack VAE on MNIST, SVHN, and CelebA. The reconstruction errors of MNIST, SVHN, and CelebA are 0.099, 0.036, 0.040 measured by the root means squared deviation. The attack target is to disturb the input significantly but the output is similar to the origin.

In Fig. 3, we show some typical adversarial examples of Type I attack for VAE on MNIST and SVHN respectively. More examples are provided in supplemental materials. Each image pair $(x_{\text{ori}}, x_{\text{adv}})$ satisfies: the original input x_{ori} and the adversarial example x_{adv} are totally different but the reconstruction results of them are similar. Results are illustrated in Table 1 quantitatively. The distance we used to measure is the root mean squared deviation in each pixel (which is normalized to $[0, 1]$). We can see that although the adversarial

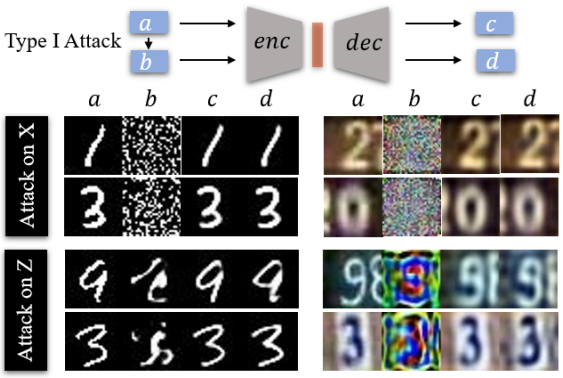

Fig. 3. Type I attack on MNIST and SVHN. The attack process is shown on the top: a. the original image; b. Type I adversarial example; c. reconstruction result of original; d. reconstruction result of adversarial example. The left column indicates Type I attack methods. Attack on MNIST shows on the left and attack on SVHN is shown on the right.

Table 1. The average distance of the adversarial examples and original ones both in the input and reconstruction output.

	Attack on X		Attack on Z	
	Dis_input	Dis_output	Dis_input	Dis_output
MNIST	0.611	0.059	0.213	0.095
SVHN	0.270	0.021	0.247	0.042

images are different from the origin, their outputs are still similar. Utilizing the threshold criterion defined before, all the Type I attack are evaluated to be successful.

For CelebA, some adversarial examples are given in Fig. 4. In each pair, the left is the original image, and its Type I adversarial example shows on the right. Note that all the distances above images are below 0.1. Accordingly, the reconstruction results of each pair can be recognized as the same person. The average attack performance is shown in Table 2. Besides the root mean square error, we use the most popular face recognizers FaceNet [21] and Insightface [22] to see whether the two reconstruction results corresponding to the adversarial image and original are the same person. For FaceNet, two people can be recognized as the same one only if their distance is below 1.2. In Insightface, when the similarity of two images exceeds 0.6, they could be identified as the same person. We can get the conclusion from Table 2 that all the attack is successful.

3.2. Type I attack on StyleGAN

Next, we evaluate the proposed attack method on StyleGAN. In Fig. 5, we display some examples of the Type I attack on StyleGAN. The attack result shows a conflict. When the distance of feature vector w_{adv} and w_{ori} is large, it is expected

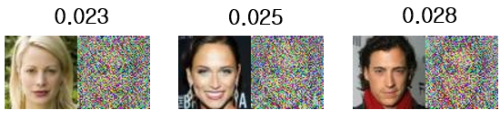

Fig. 4. Adversarial example pairs with their reconstruction distance measured by the root means squared deviation.

Table 2. The average difference of the adversarial examples and original ones measured both in the input and reconstruction output when performing attack on the image space.

	Dis_input	Pixel distance	FaceNet	Insightface
CelebA	0.308	0.029	0.453	0.878

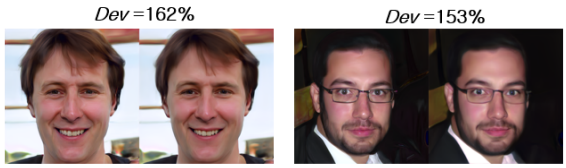

Fig. 5. Type I attack on StyleGAN. The left image is the generated face corresponding to the original feature vector. The right image is the attack result corresponding to the adversarial one. The number on the top is the deviation defined before.

Table 3. The average difference between the adversarial intermediate latent vector and original one and their corresponding generated images.

	Dev	Pixel distance	FaceNet	Insightface
StyleGAN	0.188	0.049	0.332	0.923

the corresponding outputs are totally different persons with different styles. However, under the attack, the deviations are all above 150%, the generated faces are still similar. The attack performance can be found in Table 3, showing that although features vary a lot, the generated image is still be identified as the same one.

4. CONCLUSION

In this paper, we propose Type I attack for the generative models which aims at generating a meaningless adversarial example whose output is similar to the origin. Specifically, we design Type I attack on VAE and StyleGAN and the experiments show that the proposed method successfully generates Type I adversarial examples to cheat generative models. Except for the Type II attack, Type I attack is also important to understand the generative models and worth researching because the underlying mechanisms of them are different. Type I attack for generative models can also be used to evaluate model performance and promote progress in defense methods.

5. REFERENCES

[1] 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*, 2014, pp. 2672–2680.

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

[3] Lili Mou, Rui Men, Ge Li, Lu Zhang, and Zhi Jin, "On end-to-end program generation from user intention by deep neural networks," *arXiv preprint arXiv:1510.07211*, 2015.

[4] Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra, "Towards conceptual compression," in *Advances In Neural Information Processing Systems*, 2016, pp. 3549–3557.

[5] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros, "Generative visual manipulation on the natural image manifold," in *European Conference on Computer Vision*. Springer, 2016, pp. 597–613.

[6] Tali Dekel, Chuang Gan, Dilip Krishnan, Ce Liu, and William T Freeman, "Sparse, smart contours to represent and edit images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3511–3520.

[7] Wengling Chen and James Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9416–9425.

[8] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5400–5409.

[9] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei 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.

[10] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[11] George Gondim-Ribeiro, Pedro Tabacof, and Eduardo Valle, "Adversarial attacks on variational autoencoders," *arXiv preprint arXiv:1806.04646*, 2018.

[12] 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.

[13] 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.

[14] 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*, 2016, pp. 2574–2582.

[15] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*.

[16] Ali Shahin Shamsabadi, Ricardo Sanchez-Matilla, and Andrea Cavallaro, "Colorfool: Semantic adversarial colorization," *arXiv preprint arXiv:1911.10891*, 2019.

[17] Hossein Hosseini and Radha Poovendran, "Semantic adversarial examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1614–1619.

[18] Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde, "Semantic adversarial attacks: Parametric transformations that fool deep classifiers," *arXiv preprint arXiv:1904.08489*, 2019.

[19] Sanli Tang, Xiaolin Huang, Mingjian Chen, Chengjin Sun, and Jie Yang, "Adversarial attack type i: Cheat classifiers by significant changes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[20] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[21] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[22] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.