Once a MAN: Towards Multi-Target Attack via Learning Multi-Target Adversarial Network Once

Jiangfan Han[1]*, Xiaoyi Dong[2]*, Ruimao Zhang[1], Dongdong Chen[2], Weiming Zhang[2],
Nenghai Yu[2]†, Ping Luo[3], Xiaogang Wang[1]†
[1]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong,
[2]University of Science and Technology of China, Key Laboratory of Electromagnetic
Space Information, The Chinese Academy of Sciences, [3]The University of Hong Kong
{jiangfanhan@link., ruimao.zhang@, xgwang@ee.}cuhk.edu.hk,
{dlight@mail., cd722522@mail., zhangwm@, ynh@}ustc.edu.cn, pluo@cs.hku.hk

Abstract

Modern deep neural networks are often vulnerable to adversarial samples. Based on the first optimization-based attacking method, many following methods are proposed to improve the attacking performance and speed. Recently, generation-based methods have received much attention since they directly use feed-forward networks to generate the adversarial samples, which avoid the time-consuming iterative attacking procedure in optimization-based and gradient-based methods. However, current generation-based methods are only able to attack one specific target (category) within one model, thus making them not applicable to real classification systems that often have hundreds/thousands of categories. In this paper, we propose the first Multi-target Adversarial Network (MAN), which can generate multi-target adversarial samples with a single model. By incorporating the specified category information into the intermediate features, it can attack any category of the target classification model during runtime. Experiments show that the proposed MAN can produce stronger attack results and also have better transferability than previous state-of-the-art methods in both multi-target attack task and single-target attack task. We further use the adversarial samples generated by our MAN to improve the robustness of the classification model. It can also achieve better classification accuracy than other methods when attacked by various methods.

1. Introduction

Deep neural networks (DNNs) have been significantly successful in many artificial intelligence tasks such as image classification [16, 33, 12], object detection [8, 29, 20], and natural language processing [35, 7]. However, recent studies [32, 23, 21] showed that DNNs are vulnerable and easy to be attacked by adversarial samples. By adding small perturbation onto an image, it is difficult to differentiate adversarial sample from the original image visually but can mislead modern classification model into definitely incorrect categories easily, which may produce severe security threat for real application systems such as auto-driving and face verification. Adversarial samples also possess the characteristic of transferability, which means adversarial sample generated for attacking one model could also mislead another model. Thus much attention has been concentrated on this phenomenon to better understand the weaknesses of DNNs, as well as to develop and strengthen the robustness of deep networks.

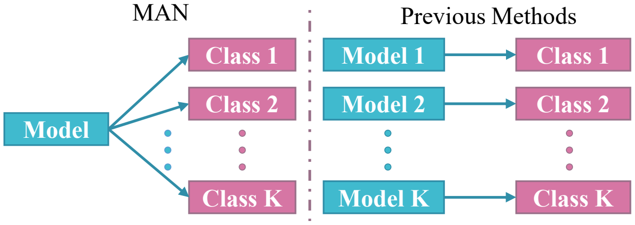


Figure 1. The comparison between our MAN and previous target attack methods. Our network can generate adversarial sample to all classes by training a single model once while others need individual models for each class.

The pioneering work from [32] used an optimization-based method to generate adversarial samples. Then many different types of methods [23, 21, 26, 31] were proposed to improve the attacking performance and speed. For example, Goodfellow *et al.* [9] proposed the first gradient-based method, which can attack DNNs effectively by us-

*Equal contribution.
†Nenghai Yu and Xiaogang Wang are the corresponding authors.

ing the back-propagated gradient to update the input image with small perturbations iteratively. Moosavi-Dezfooli *et al*. [21] designed an iterative linearization of the classifier to generate minimal perturbations that are sufficient to change classification label. Although these optimization-based and gradient-based methods can produce very good attacking results, they often rely on a very time-consuming iterative procedure, which impose a great computation burden for real attacking systems.

Recently, generation-based methods [1, 28, 36] received much attention in the literature. They directly trained a generation model to learn how to transform input images to adversarial samples. This kind of methods can be viewed as the acceleration versions of above optimization-based and gradient-based methods. By training on a large scale of images with the pre-defined attacking object, these models do not need to access the target model again and can generate adversarial samples just with a fast forward pass during the runtime. However, one of the critical problems of current generation-based methods is that they are only able to realize single-target attack, meaning that adversarial samples generated by one model can only mislead the prediction of the attacked model to one specific predefined category during training. If we want to mislead the attacked model to another category, a new model needs to be trained, which is both time and storage consuming. Therefore, they are not feasible to real classification systems which contain hundreds/thousands of categories.

Motivated by these shortcomings, we propose a novel framework called *Multi-target Adversarial Network (MAN)*, which aims to generate multi-target attack samples by training the adversarial model once. As illustrated in Figure 1, MAN does not require specific attack target in training and can produce target adversarial samples for all categories in a dataset. Compared with existing single-target attack methods, MAN just trains once for all categories rather than training different models for different categories.

As shown in Figure 2(a), MAN adopts the encoder-decoder network to embed the target information from incorrect categories and appearance information from input images to generate adversarial samples. It accomplishes the target-independent adversarial training by a simple framework with intuitive loss functions. Moreover, through extensive experiments, we find that the adversarial sample generated by such a model has strong transferability. Figure 3 illustrates the adversarial samples generated by MAN. When we adopt different target information, MAN enables the attack samples to effectively inherit the appearance information from the original image, while cheating the pretrained model with high fooling rates.

Our **contributions** are summarized as follows.

- To the best of our knowledge, this is the first work to present the task of multi-target attack that generates target adversarial samples for all the categories in a dataset using one single model. We accomplish this purpose by presenting a novel Multi-target Adversarial Network (MAN), which can produce multi-target adversarial samples by training a single model, and can significantly reduce the training cost and model storage.
- In the single-target scenario, the proposed method achieves competitive and better attack performance and generalization ability, compared to state-of-the-art methods [1, 28] in various popular deep architectures [30, 12]. For example, MAN achieves 98.55% attack accuracy on the CIFAR10 dataset against VGG16 and 88.95% when transferring to ResNet32, outperforming [28] by 4.39% and 12.49% respectively. For the multi-target attack scenario, our model can maintain these advantages by training the multi-target adversarial network only once.
- The generated attack samples by MAN effectively improve the robustness and adversarial defense ability of attacked models. Take the attacked model pretrained on the CIFAR10 as an example, when using the strongest attack samples generated by MAN to finetune ResNet32, the classification accuracy is improved from 10.58% to 81.14%, outperforming other counterparts by 17.74%.

2. Related Work

2.1. Adversarial Attack Methods

Current methods for generating adversarial samples can be divided into three categories: optimization-based [32, 2], gradient-based [9, 17, 5], and generation-based [1, 28]. Optimization-based methods model the generation of adversarial samples as an optimization problem and use optimizers like box-constrained L-BGFS [6] or Adam [14] to solve it, which are powerful but quite slow. Goodfellow *et al*. [9] proposed the first gradients-based method Fast Gradient Sign Method (FGSM) and Alexey *et al*. [17] performed small step size iteratively to get better attacking performance (I-FGSM). The speed of gradient-based methods is linearly related to the number of parameters of target models and iteration numbers, and is often faster than optimization-based methods. However, due to their inherent model-specific mechanism, previous gradient-based methods have relatively weak transferability. Dong *et al*. [5] further introduced momentum into the iterative steps (MI-FGSM) to get better transferability. Different from the iterative procedure used in the above two types of methods, generation-based methods directly use generation models to transform input images into adversarial samples. For example, an auto-encoder like network was adopted in [1],

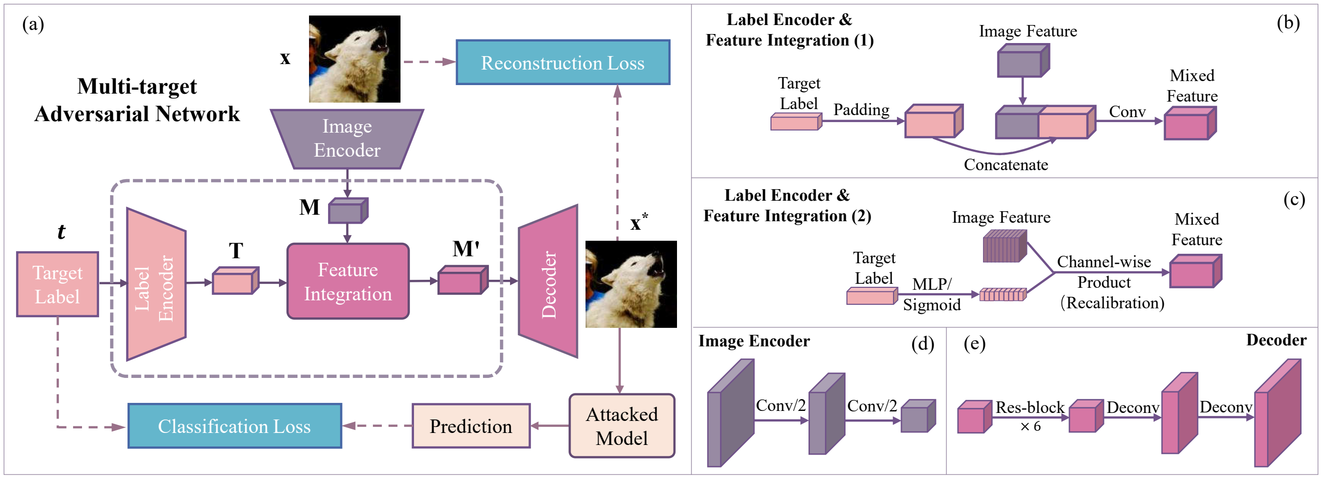


Figure 2. The structure of Multi-Target Adversarial Network (MAN), (a) shows the overall architecture, (b) and (c) show two different label encoding and feature approaches, (d) and (e) are the architecture for image encoder and decoder respectively.

and U-Net and ResNet were leveraged in [28]. Although these methods need extra time to train, they can generate adversarial samples at a constant speed and do not need to access to the target model again in the inference stage. The biggest limitation of all current generation-based methods is that they are only able to attack one specific category with one model, which is very unfriendly for real attacking systems. By contrast, the multi-target adversarial network proposed by us enables attacking any category of a classification model with similar attack success rate, and can also bring better transferability for the adversarial attack task as well as model robustness for the adversarial training task.

2.2. Adversarial Defence Methods

There were also many methods proposed to defend against adversarial attack. Adversarial training used in [10, 18, 34] trained a robust network against adversarial attack by adding the adversarial sample into the training set and jointly trained with original samples. In [11, 24, 3, 19], they used preprocessing procedures to remove the adversarial noise of the input images before feeding into the target model. Other methods [34, 25, 27] used some regularizes or smooth labels to make the target model more robust to the perturbation on input images. In the methods mentioned above, adversarial training is the most effective method. But it requires enough adversarial samples for training, so the speed of generating adversarial samples determines the effectiveness and performance of adversarial training. In this paper, we focus on how to generate adversarial sample to attack all categories with high speed, and this will be helpful for training more robust models.

3. Multi-Target Attack

Non-target *vs*. Single-target attack. Let $\mathcal{H}$ be a deep neural network to be attacked, which is trained on a dataset with K classes. For an image $\mathbf{x}$ with the category label $y \in \{1, 2, ..., K\}$, the predicted probability of the network $\mathcal{H}$ for the i-th category is represented by $\mathcal{H}_i(\mathbf{x})$, where $i \in \{1, 2, ..., K\}$. The non-target adversarial attack tries to generate a new image $\mathbf{x}^*$, such that $\operatorname{argmax}_i \mathcal{H}_i(\mathbf{x}^*) \neq y$, It implies that predicted label of $\mathbf{x}^*$ should not be the ground-truth label of $\mathbf{x}$. By contrast, single-target attack aims to generate the adversarial sample $\mathbf{x}^*$, such that $\operatorname{argmax}_i \mathcal{H}_i(\mathbf{x}^*) = t$, where $t \neq y$ is a predefined class label that is specified in the testing stage. According to the above definition, the non-target attack is much easier to realize than single-target attack, since it only requires the attacked model to make different predictions between the adversarial image and the corresponding original image. Beyond the single-target attack, this work focuses on a more challenging task, multi-target attack, whose goal is to generate multi-target adversarial samples by a single network.

White-Box attack *vs*. Black-Box attack. If we have the knowledge about the attacked model $\mathcal{H}$, including model architecture and parameters, we call such attack as a white-box attack. In this case, the attacker can generate adversarial samples directly using attack methods by accessing the attacked model. On the contrary, for black-box scenario, the attacker knows nothing about the target model. Therefore the designed attacking model need stronger transferability, which means that adversarial samples generated for attacking one model could also mislead another model.

Problem Definition. In order to generate a target adversarial sample $\mathbf{x}^*$, a transformation function $\mathcal{F}_\theta$ with parameters θ is defined to map the input image $\mathbf{x}$ to a target domain, $\mathbf{x}^* = \mathcal{F}_\theta(\mathbf{x}, t)$. In practice, we approximate this function by a deep neural network, where θ represents the network parameters. The purpose of multi-target attack is to train a network $\mathcal{F}_\theta$ such that $\operatorname{argmax}_i \mathcal{H}_i(\mathcal{F}_\theta(\mathbf{x}, t)) = t$, where $t \neq y$. Unlike traditional single-target attack that trains specific network $\mathcal{F}^{t_0}$ for a predefined label t_0, multi-target attack only trains a single network $\mathcal{F}$ for any target label $t \in \{1, 2, ..., K\}$.

3.1. Multi-Target Adversarial Network (MAN)

Architecture Overview. The overall architecture of MAN is shown in Figure 2 (a). It takes image $\mathbf{x}$ and target label t as input and generates adversarial sample $\mathbf{x}^*$ corresponding to the desired target. Compared with traditional single-target adversarial networks [28], the target label t is regarded as a discrete variable rather than a constant in MAN. The range of t is from 1 to K. Accordingly, the encoding network of MAN includes two branches, one for extracting the appearance features from the input image, as shown in Figure 2 (d), and the other for encoding the target label information. The feature integration module is also introduced to integrate the feature representation from these two modalities. As shown in Figure 2 (e), the decoder network with six residual blocks and two deconvolutional layers are adopted to generate the final adversarial samples.

Two intuitive losses are employed to optimize the adversarial network. One is the reconstruction loss to preserve the appearance similarity between the original image and the adversarial samples and the other is the classification loss which is adopted to embed the target label information into the attack samples. To calculate the classification loss, the generated attack sample is fed into the attacked model, *i.e.* the pretrained classification network, to predict the label. Although the parameters in the attacked model are fixed during the MAN training, the gradients can still be passed through this model to guide the generative subnetwork.

Variants of MAN. Next, we describe two versions of feature integration schemes of MAN. The first one is illustrated in Figure 2 (b), which is a quite straightforward idea to integrate the image feature and label feature by concatenating along the channels. Given an input image $\mathbf{x}$, the image encoder first calculates the image feature map $\mathbf{M} \in \mathbb{R}^{C\times H\times W}$, where C, H and W indicate the number of channels, height and width of feature maps respectively. The target label t is presented as the one-hot vector $\mathbf{t} \in \mathbb{R}^{K}$, where $\mathbf{t}$ is expanded along height and width directions to get the label feature map $\mathbf{T} \in \mathbb{R}^{K\times H\times W}$. Then the above two sets of feature maps are concatenated along the channels to get a mixed one $\tilde{\mathbf{M}}' \in \mathbb{R}^{(K+C)\times H\times W}$. An additional convolutional layer is adopted to reshape the mixed feature $\tilde{\mathbf{M}}'$ to $\mathbf{M}' \in \mathbb{R}^{C\times H\times W}$. At last, $\mathbf{M}'$ is fed into the decoder to generate the adversarial sample $\mathbf{x}^*$. In this approach, the mixed feature is generated by concatenating the feature maps of image and target label. We denote this model as the multi-target adversarial concatenate network (MANc).

The second architecture of label encoder and feature integration is illustrated in Figure 2 (c). Inspired by the Squeeze and Excitation Network proposed by Jie *et al.* [13], we claim that different channels of feature representation can capture diverse features for different classes. Thus the channel-wise product, *i.e.* recalibration operation, is used to integrate the label feature and image feature. In such a case, the label encoder contains a two-layer multi-layer perceptron (MLP), and the output is activated by sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$. We denote $\mathbf{t}'$ as the output of MLP, which is given by:

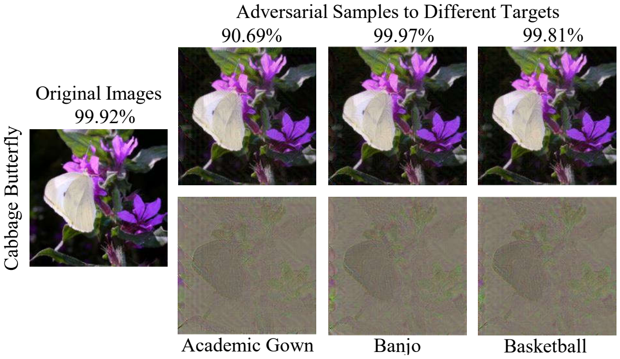


Figure 3. Adversarial samples to different target with $\epsilon = 10\sqrt{N}$. Numbers are the predicted probability for the corresponding label given by a pretrained VGG16 model.

$$\mathbf{t}' = \mathcal{F}_{en}(\mathbf{W}, \mathbf{t}) = \sigma(\mathbf{W}_2\delta(\mathbf{W}_1\mathbf{t})), \tag{1}$$

where δ is the ReLU [22] activation function, $\mathbf{W}_1 \in \mathbb{R}^{U\times K}$ and $\mathbf{W}_2 \in \mathbb{R}^{C\times U}$ are fully connected layers, U is the number of intermediate units, $\mathbf{t}' \in \mathbb{R}^{C}$, and σ limits every element $t'_i \in (0, 1)$. The final integrated feature representation is presented as $\mathbf{M}' = \mathbf{t}' \odot \mathbf{M}$, where $\odot$ indicates the channel-wise product on the tensor: $\mathbf{M}'_i = t_i \cdot \mathbf{M}_i$, in which $\mathbf{M}_i \in \mathbb{R}^{H\times W}$ is channel-wise feature map of image feature $\mathbf{M} = [\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_C]$. We refer this structure as multi-target adversarial recalibrating network (MANr).

3.2. Optimization

To optimize the network parameters, we define the objective function of MAN for target label t as follows,

$$\mathcal{L}(\mathbf{x}) = \mathcal{L}_{cls}(\ \mathcal{H}(\mathcal{F}_\theta(\mathbf{x}, t)), t\ ) + \alpha\mathcal{L}_{re}(\ \mathbf{x}, \mathcal{F}_\theta(\mathbf{x}, t)), \tag{2}$$

where $\mathcal{H}(\mathcal{F}_\theta(\mathbf{x}, t))$ is the predicted probability of the target model on adversarial sample, t is the discrete label variable, weight factor α decides the importance of two loss terms, and $\mathcal{L}_{cls}$ is the classification loss which encourages the attacked model make prediction to the target label. The concrete format of loss function has massive impact on the final result. Previous studies [1, 28, 2] tried different classification losses in their works; we found that the standard cross-entropy loss works well in all of our experiments.

The reconstruction loss $\mathcal{L}_{re}$ measures the appearance difference between adversarial sample and original image.

Attacked model		VGG16	VGG19	Res152	Res101
VGG16	ATN [1]	88.18*	55.68	0.15	0.13
	GAP [28]	99.89*	70.83	0.53	0.30
	MANc	99.85*	**85.79**	**0.90**	**0.50**
	MANr	**99.93***	71.83	0.50	**0.50**
Res152	ATN [1]	0.07	0.08	81.95*	2.31
	GAP [28]	19.17	16.51	99.65*	73.53
	MANc	3.58	5.48	**99.67***	52.75
	MANr	**20.47**	**23.74**	99.72*	**77.61**

Table 1. The success rate (%) of single-target adversarial attack on the ImageNet dataset with different target model. * indicates the white-box attacks.

Attacked model		VGG16	VGG19	Res32	Res14
VGG16	ATN [1]	87.97*	65.71	69.85	71.81
	GAP [28]	94.16*	77.60	76.46	78.90
	MANc	97.86*	87.67	88.26	88.63
	MANr	**98.55***	**91.07**	**88.95**	**90.33**
Res32	ATN [1]	37.41	50.74	93.17*	69.60
	GAP [28]	43.71	57.19	97.02*	74.30
	MANc	51.27	68.11	98.86*	**81.01**
	MANr	**58.56**	**72.26**	**99.26***	79.33

Table 2. The success rate (%) of single-target adversarial attack on the CIFAR10 dataset with different target model. $*$ indicates the white-box attacks.

It is usually measured by vector norm $||\mathbf{x} - \mathcal{F}_\theta(\mathbf{x}, t)||_p$ where $p \in \{0, 1, 2, \infty\}$ [2]. In this paper, we adopt Euclidean distance, *i.e.* L_2 norm, as our measurement. To ensure the difference between adversarial sample and the original image is small, *i.e.* the perturbation is restricted to a range. A fixed distance upper bound ϵ is given and we force the constraint $||\mathbf{x} - \mathcal{F}_\theta(\mathbf{x}, t)||_2 < \epsilon$. In practice, we use α to leverage the power of the two loss terms. Larger α encourages better reconstruction quality while smaller ones get higher success attack rate.

4. Experiments

4.1. Adversarial Attack

In this section, we evaluate the attack effectiveness of MAN on the ImageNet [4] and the CIFAR10 [15].

General setup. On the ImageNet [4] dataset, we conduct the experiments by using popular network architectures VGG16 [30], and ResNet152 (Res152) [12] as attacked models . Meanwhile, we use VGG19 and Res101 as "black-box" model. For fair comparison, in the testing stage, we scale the norm of all perturbations to a certain threshold ϵ. *i.e.* $\hat{\mathbf{x}}^* = \mathbf{x} - \frac{\epsilon}{||\mathbf{x} - \mathcal{F}_\theta(\mathbf{x}, t)||_2}(\mathbf{x} - \mathcal{F}_\theta(\mathbf{x}, t))$. Here $\hat{\mathbf{x}}^*$ is the scaled adversarial sample for testing. The threshold of the L_2 norm ϵ in our experiments is calculated by $\delta\sqrt{N}$, where $N = C \times H \times W$ is the dimension of input images $\mathbf{x}$. We set $\delta = 10$ in this part and further exploration on δ is listed in the ablation study. We use Adam [14] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to train all of the adversarial networks. The batch size is 32. For the CIFAR10 [15], the attacked networks are VGG16 [30] and ResNet32 (Res32) [12], and "black-box" models are VGG19 and Res14. We set batchsize to 128, and other settings are the same as the Imagenet.

Single-target attack. In this part, we evaluate the attack performance on the single-target attack task. We fix the input label t during the single-target training process to get a degraded version of our model. We compare the proposed MANc and MANr with two other state-of-the-art methods ATN [1] and GAP [28]. For all of the methods, we train adversarial models for 120K iterations, with an initial learning rate of 0.002 and decreased by 10 after 80K iterations on the ImageNet. On the CIFAR10 training set, we train 20K iterations with an initial learning rate of 0.001 and decreased by 10 after 16K iterations. The hyperparameter α is set to 100 (1 for ATN) on the ImageNet and 800 (80 for ATN) on the CIFAR10. During testing, adversarial samples are generated by 50K images in the ImageNet validation set and 5k images in the CIFAR10 validation set. We use the success rate as the evaluation metric, which is defined as the ratio of the attacked model classifying the adversarial samples to the target class.

Table 1 and Table 2 report the attack success rates on two datasets. We first evaluate the white-box attack performance. From the results, we can find that ATN performs quite weak for all cases. Comparing with GAP, both MANc and MANr achieve better results in most cases, *e.g.* MANr achieves 98.55% attack accuracy against VGG16 on the CIFAR10, outperforming GAP by 4.39%.

When it comes to the black-box attack. ATN shows very weak transferability. When transferring to a model with similar architecture, *e.g.* from VGG16 to VGG19, our MANc get 85% success rate on the ImageNet and nearly 90% on the CIFAR10, outperform GAP by over 10%.

When transferring to a model with totally different architecture *e.g.* from VGG to ResNet, we find results on the CIFAR10 decrease a bit but still higher than 88% in most cases. But on the ImageNet, the success rate of all methods are lower than 25%. We think this is because there are too many categories in the ImageNet dataset, making it hard to transfer to another model with a specific attack target.

According to the above results, our method performs much better in most cases. We think this is because our method splits the task into two branches. One branch guides the generation model to generate adversarial samples similar to input as the previous methods did. The other branch encodes the target label and provides extra guidance helping generate more powerful adversarial samples.

Multi-target attack. In the training phase for multi-target attack, a random target label is assigned to each train-

Attacked model		VGG16	VGG19	Res152	Res101
VGG16	MANc	**99.22***	**66.81**	**2.48**	**1.42**
	MANr	99.13*	54.16	1.73	1.33
Res152	MANc	**7.61**	**7.71**	98.14*	**55.39**
	MANr	5.76	6.06	**98.23***	49.69

Table 3. The success rate (%) of multi-target adversarial attack on the ImageNet dataset with different model. $*$ indicates the white-box attacks.

Attacked model		VGG16	VGG19	Res32	Res14
VGG16	MANc	99.14*	92.97	90.08	88.56
	MANr	**99.50***	**95.47**	**92.68**	**89.74**
Res32	MANc	76.86	86.87	**99.30***	**90.27**
	MANr	**78.36**	**88.15**	98.94*	84.98

Table 4. The success rate (%) of multi-target adversarial attack on the CIFAR10 dataset with different model. $*$ indicates the white-box attacks.

ing image. We train models for 280K iterations, with an initial learning rate of 0.002 and decreased by 10 after 240K iterations on the ImageNet training set. On the CIFAR10 training set, we train 80K iterations, with an initial learning rate of 0.001 and decreased by 10 after 64K iterations. At the evaluation phase, we randomly pick 5000 samples from the ImageNet validation set and randomly assign ten labels for each sample. For the CIFAR10, the validation set contains 5000 samples and we assign all of the ten labels to each sample. In total, there are 50K adversarial samples generated to attack the pretrained models on each dataset.

Table 3 and Table 4 report the attack success rates of variants of MAN on the ImageNet and on the CIFAR10 respectively. For white-box case, we find our models keep a great performance with attack accuracy over 98%. When evaluating the black-box performance on both datasets, we find our models also keep a high transferability. Since the evaluation metric in Table 1, 2 and Table 3, 4 are different, the result can not be compared directly. We make further comparison under the same evaluation metric in the ablation study.

Results show the feasibility and effectiveness of our methods to realize the multi-target attack task. Even on a large dataset with numerous categories, our method can still attack arbitrary category with a single model.

There is also an interesting phenomenon that MANr performs better on CIFAR10 while MANc has better results on ImageNet. MANr re-calibrates each channel of the feature map. It controls better when size is small. When size is large, it is not easy to re-calibrate the feature map. MANc concatenates feature maps to get a mixed one, which is not influenced by the size of feature map and thus performs evenly on both datasets. Figure 3 shows the original samples and adversarial samples generated by different target label.

Method	No. of Params	Total Models	Total Training Iters
ATN [1]	7.84M	1000	120M
GAP [28]	7.84M	1000	120M
MANc	10.74M	1	280K
MANr	8.55M	1	280K

Table 5. Comparison of our method with single-target methods on Number of parameters for each model, the total number of models used and total training iteration needed to realize attack all categories in the ImageNet.

Parameter Comparison. Table 5 lists the number of parameters for each model, the number of models needed and the total training iterations needed to attack all categories in the ImageNet. Previous models have about 8.3% fewer parameters comparing with MANr, but 1K models with in total 7840M parameters and 120M iterations are required to attack all categories in the ImageNet. By contrast, our method just adopts a single model with much fewer parameters and 0.3M iterations to achieve the same goal. Thus our method is significantly more time and storage efficient.

4.2. Adversarial Training

Adversarial training is one of the most effective method to improve the robustness of models against adversarial attack. It finetunes the network by adversarial samples with the groundtruth labels of the images used to generate the adversarial samples. In this part, we evaluate the improvement of classification model against adversarial attack when the adversarial samples are used for finetuning.

Setups. All of the following experiments are conducted on the CIFAR10 dataset. To improve the robustness of the classification model, we use adversarial samples and their ground truth labels to finetune the pretrained networks. In the rest part of the paper, we use "adv-model" to indicate the model finetuned by the adversarial samples. The attacked model architecture is ResNet32, and the adversarial samples are generated by MANc, MANr and GAP [28] respectively with $\epsilon = 10\sqrt{N}$.

For MANc and MANr, we generate adversarial samples with random target labels and finetune the attacked model with these samples. We also train ten GAP models with respect to ten attack target, each of which is a single-target attack model, to produce adversarial samples. We finetune the attacked model with adversarial samples generated by one, five and ten GAP models to explore the influence of the number of classes used during adversarial training. The corresponding settings are denoted as GAP1, GAP5, and GAP10 respectively. All the adv-models are finetuned with 40K iterations with batch size 128 (64×2, each image is coupled with its adversarial sample). The learning rate is set to 0.001. To further evaluate the robustness of these adv-models, we use MI-FGSM [5] and ATN [1] as attack methods. In practice, we use MI-FGSM to generate adversarial samples with random target labels and follow the op-

Attack Strength	Attack Method	GAP 1	GAP 5	GAP 10	MANc	MANr	Raw Res32
$8\sqrt{N}$	ATN [1]	13.38	11.60	11.37	**11.17**	11.43	86.41
	MI-FGSM [5]	18.18	13.96	13.76	**12.34**	13.24	99.40
$12\sqrt{N}$	ATN [1]	19.40	13.50	13.08	**12.57**	13.28	94.56
	MI-FGSM [5]	23.56	18.86	15.28	14.86	**13.84**	99.68
$16\sqrt{N}$	ATN [1]	24.93	16.09	15.27	**14.69**	15.88	97.97
	MI-FGSM [5]	29.22	22.22	17.44	17.18	**16.88**	99.76

Table 6. Attack success rate (%) against classification models. Lower success rate indicate the model is more robust. ResNet32 is the attacked model to generate adversarial samples by MI-FGSM and ATN with L_2 threshold $8\sqrt{N}$,$12\sqrt{N}$,$16\sqrt{N}$. We also add Raw Res32 (ResNet32 without adversarial training) to make sure the effectiveness of attack methods.

Attack Strength	Attack Method	GAP 1	GAP 5	GAP 10	MANc	MANr	Raw Res32
$8\sqrt{N}$	ATN [1]	86.17	89.48	89.85	**89.91**	89.36	20.77
	MI-FGSM [5]	81.90	86.60	88.74	89.24	**89.34**	9.54
$12\sqrt{N}$	ATN [1]	75.48	85.38	85.70	**86.06**	84.87	13.04
	MI-FGSM[5]	73.70	81.40	84.78	**86.14**	86.06	10.12
$16\sqrt{N}$	ATN [1]	63.41	78.87	78.64	**79.15**	77.17	10.78
	MI-FGSM [5]	63.40	73.86	79.08	**81.14**	81.00	10.58

Table 7. Classification accuracy (%) for classification models. Higher accuracy indicate the model is more robust.

timal settings used in [5]. For ATN method, we follow the generation procedure stated in Sec. 4.1. Note that all the attack samples for testing are generated by using pretrained ResNet32 architecture on the CIFAR10 test set.

We consider the robustness of a classification model in two aspects. On the one hand, we define the attack success robustness by the attack success rate of adversarial samples to the adv-models. In such a case, a lower success rate indicates the model is more robust. On the other hand, the classification success robustness is also defined to evaluate whether the adv-models could classify the adversarial samples to their ground truth classes. In this subsection, we set the attack strength (threshold) $\epsilon = 8\sqrt{N}, 12\sqrt{N}, 16\sqrt{N}$. Two of these thresholds are larger than that in the last part because the adv-models has already finetuned by adversarial samples. A larger threshold can also reflect the robustness of adv-models against stronger attack.

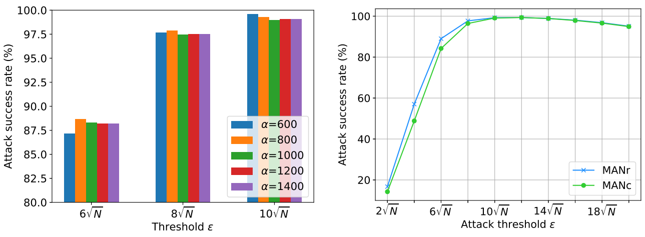
Attack success robustness. We show the result of attack success rates in Table 6. According to the results, the attack success rate grows as the attack strength grows. When the attack strength is small, *i.e.* $8\sqrt{N}$, it is quite hard to use adversarial samples to fool any of the adv-models, while GAP1 is slightly worse than others. When the strength becomes larger, *i.e.* $12\sqrt{N}$, GAP1 and GAP5 models tend to be vulnerable to adversarial samples, which almost increase the attack success rate by around 5%, whereas GAP10 and our models are still robust, and the success attack rate is slightly increased by around 2%. When the attack strength is set to $16\sqrt{N}$, GAP1 and GAP5 let over 20% adversarial samples attack successfully. Other models also tend to be vulnerable in such case, but our MANs still slightly outperform the GAP10. *e.g.* MANr can defense 0.56% more adversarial samples against MI-FGSM attack.

Classification success robustness. We report the results of classification success robustness in Table 7. By using small attack strength, *i.e.* $8\sqrt{N}$, all of the adv-models can classify most adversarial samples correctly and achieve high classification accuracy (larger than 80%). When the attack strength becomes larger, *i.e.* $12\sqrt{N}$, the accuracy of the GAP1 model decrease obviously, while other adv-models can still maintain a promising accuracy over 80%. When setting the attack strength to $16\sqrt{N}$, only GAP10, MANc and MANr can keep the accuracy around 80%. Moreover, our methods get better results. *e.g.* For MI-FGSM attack method, MANc can get 81.14% accuracy, outperforming GAP10 by 2.06%

In summary, when we compare the adversarial training results among GAP1, GAP5 and GAP10 models, we find that when the number of target models increases, the adv-models can obtain more ability to resist the adversarial attack, which demonstrates the necessity to retrain the attacked model using multiple single-target attack models. At the same time, compared with single-target methods, our models show better performance for almost all cases. It demonstrates that our methods can generate diverse adversarial samples to promote the retraining efficiency of adv-models. Our methods can use fewer training time and storage resources to produce richer adversarial samples and make the attacked models more robust.

4.3. Ablation Study

Weight factor α. In this part, we explore the influence of weight factor α for attack result. As we stated above, a larger α encourages better reconstruction quality while smaller ones lead to higher attack rate. When evaluating the results, we compare the attack rate under a certain re-



(a) (b)

Figure 4. (a) The relation between weight factor α and attack success rate with different ϵ on the CIFAR10 to a pretrained VGG16 model. (b) The attack accuracy under different threshold ϵ.

		VGG16	VGG19	Res32	Res14
Res32	Single-target	58.71	75.69	**99.70***	90.51
	Multi-target	**67.69**	**82.19**	99.16*	**93.37**

Table 8. The success rate (%) of MANr based Single-target and Multi-target attack model. Multi-target model attack the same target as single ones.

construction threshold ϵ, and the influence of it becomes complicated.

Figure 4 (a) shows the attack success rate on the CIFAR10 dataset with different weight factor α to a VGG16 model. We can find that smaller α performs well when the permitted threshold ϵ is large. Larger α often performs better when the ϵ at a small value, but sometimes it may depress the performance among all ϵ cases.

Threshold ϵ. In this part, we explore the influence of different threshold ϵ. Larger ϵ means more severe change to the original image. We test the attack accuracy under different ϵ from $2\sqrt{N}$ to $20\sqrt{N}$ with step size $2\sqrt{N}$.

Figure 4 (b) shows the attack success rate on the CIFAR10 dataset. We find that smaller ϵ (such as $2\sqrt{N}$, $4\sqrt{N}$) limits the success rate. When ϵ increases, success rate grows at the same time. $10\sqrt{N}$ and $12\sqrt{N}$ get the best performance so we use $10\sqrt{N}$ in our main experiments. But when the ϵ is too large, the performance decreases a bit. We think this is because perturbation with too large ϵ impedes the representation of adversarial features.

Transferability. In this part, we make further exploration about the transferability of single-target attack model and multi-target attack model. Different from last section which we take more iterations to train the multi-target attack model, in this part we use same training iterations for all models. For single-target models, we fix the attack target label during training, while multi-target models accept random input labels. In testing phase, we use the same attack target for multi-target models as used in single target ones.

The results are listed in Table 8. According to the results, even we train two models with the same iterations, the multi-target model performs better transferability than the single-target model in all black-box cases, *e.g.* it obtains 8.98% gain when transferring from ResNet32 to VGG16.

		VGG16	VGG19	Res32	Res14
VGG16	MANc	97.86*	87.67	88.26	88.63
	MANr	98.55*	91.07	88.95	90.33
Res32	MANc	51.27	68.11	98.86*	81.01
	MANr	58.56	72.26	99.26*	79.33
VGG16+Res32	MANc	98.90*	96.75	99.39*	96.79
	MANr	**99.60***	**98.07**	**99.69***	**98.06**

Table 9. The success rate (%) of multi-target adversarial attack against different models on the CIFAR10 dataset. The last row indicates combine VGG16 and ResNet32 as white-box model.

We conjecture that the competition between different target label promote the model to learn more generalization feature with more generalization ability and this result proves our conjecture.

Model Ensemble. We also try to combine different attacked models during the training process, the last row of Table 9 shows the success rate jointly trained on VGG16 and ResNet32 models. It is obvious that ensemble different attacked models make the transferability much better than a single architecture. Considering the performance transferring form ResNet32 to VGG19, ensemble method exceeds the single architecture over 10%. This is reasonable since the network attempt to fit various architectures, thus it can be generalized well to other unseen models.

5. Conclusion

In this paper, we propose a novel adversarial model named Multi-target Adversarial Network (MAN) to deal with the multi-target attack problem. It can produce multi-target adversarial samples by training a single model and significantly reduces the training cost and model storage. By embedding the target label information into the generated adversarial samples, a great improvement on attack ability and transferability is shown by these samples. MAN also produces diverse adversarial samples efficiently for adversarial training, which greatly improves the robustness of models. Future work may lie on adding random noise or introducing additional regularization term to generate adversarial samples with better transferability to black-box attack. More applications about the adversarial samples are also considered to be developed, *e.g.* adversarial attack under more realistic constraints.

Acknowledgement

This work is supported in part by SenseTime Group Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants CUHK14202217, CUHK14203118, CUHK14205615, CUHK14207814, CUHK14213616, in part by the Natural Science Foundation of China under Grant U1636201 and 61572452, and by Anhui Initiative in Quantum Information Technologies under Grant AHY150400.

References

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

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

[3] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E. Kounavis, and Duen Horng Chau. Keeping the Bad Guys Out: Protecting and Vaccinating Deep Learning with JPEG Compression. *arXiv e-prints*, page arXiv:1705.02900, May 2017.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Xiaolin Hu, Jianguo Li, and Jun Zhu. Boosting adversarial attacks with momentum. *arXiv preprint arXiv:1710.06081*, 2017.

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

[7] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[9] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv e-prints*, page arXiv:1412.6572, Dec 2014.

[11] Shixiang Gu and Luca Rigazio. Towards Deep Neural Network Architectures Robust to Adversarial Examples. *arXiv e-prints*, page arXiv:1412.5068, Dec 2014.

[12] 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.

[13] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 7, 2017.

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

[15] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[16] 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.

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

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial Machine Learning at Scale. *arXiv e-prints*, page arXiv:1611.01236, Nov 2016.

[19] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser. *arXiv e-prints*, page arXiv:1712.02976, Dec 2017.

[20] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017.

[21] 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.

[22] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[23] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[24] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Prez-Cabo. No bot expects the deepcaptcha! introducing immutable adversarial examples, with applications to captcha generation. *IEEE Transactions on Information Forensics and Security*, 12(11):2640–2653, Nov 2017.

[25] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical Black-Box Attacks against Machine Learning. *arXiv e-prints*, page arXiv:1602.02697, Feb 2016.

[26] 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.

[27] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the Science of Security and Privacy in Machine Learning. *arXiv e-prints*, page arXiv:1611.03814, Nov 2016.

[28] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. *arXiv preprint arXiv:1712.02328*, 2017.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

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

[31] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[32] 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.

[33] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[34] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble Adversarial Training: Attacks and Defenses. *arXiv e-prints*, page arXiv:1705.07204, May 2017.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[36] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.