Enhancing transferability of adversarial examples via rotation-invariant attacks

Yexin Duan\textsuperscript{1,2} | Junhua Zou\textsuperscript{1} | Xingyu Zhou\textsuperscript{3} | Wu Zhang\textsuperscript{1} | Jin Zhang\textsuperscript{2} | Zhisong Pan\textsuperscript{1}

\textsuperscript{1}Command and Control Engineering College, Army Engineering University of PLA, Nanjing, China
\textsuperscript{2}Army Military Transportation University of PLA, Zhenjiang Campus, Zhenjiang, China
\textsuperscript{3}Communication Engineering College, Army Engineering University of PLA, Nanjing, China

Correspondence
Zhisong Pan, Command and Control Engineering College, Army Engineering University of PLA, Nanjing 210000, China.
Email: hotpos@hotmail.com
Jin Zhang, Army Military Transportation University of PLA, Zhenjiang Campus, Zhenjiang 212000, China.
Email: zhang_jin_1115@163.com

Funding information
National Natural Science Foundation of China, Grant/Award Number: 62076251

Abstract
Deep neural networks are vulnerable to adversarial examples. However, existing attacks exhibit relatively low efficacy in generating transferable adversarial examples. Improved transferability to address this issue is proposed via a rotation-invariant attack method that maximizes the loss function w.r.t the random rotated image instead of the original input at each iteration, thus mitigating the high correlation between the adversarial examples and the source models and making the adversarial examples more transferable. Extensive experiments show that the proposed method can significantly improve the transferability of the adversarial examples with almost no extra computational cost and can be integrated into various methods. In addition, when this method is easily applied through a plug-in, the average attack success rate against six robustly trained models increases by 5.4% over the state-of-the-art baseline method, demonstrating its effectiveness and efficiency. The codes used are publicly available at https://github.com/YeXinD/Rotation-Invariant-Attack.

1 | INTRODUCTION

Deep neural networks (DNNs) have led to remarkable achievements in visual recognition tasks [1]. However, DNNs have been found vulnerable to adversarial examples generated by adding imperceptible perturbations to benign images to fool deep models [2]. Besides image classification [3], adversarial examples have also been present in object detection and semantic segmentation [4]. Because of potentially serious consequences, adversarial examples have been a subject of great concern in security-critical areas such as autonomous vehicles and medical diagnosis. Investigating adversarial examples can help us evaluate and facilitate the robustness of various deep models.

Attackers can generate adversarial examples using white-box or black-box approaches. Because it is generally impossible to access the internal structures and parameters of the target model, a black-box method must be used to perform the attack. There are two main types of black-box attacks, query-based and transfer-based. The query-based methods [5, 6] require too many queries to be viable in the real world. Research [2, 7, 8] has shown that adversarial examples have cross-model transferability; that is, an adversarial example generated for one model can fool another model, enabling black-box attacks for real-world adversaries.

According to the number of steps of gradient computation, the attacks can be categorized into single-step [3] and iterative attack [9]. Compared with the single-step attack, the iterative method can generate stronger adversarial examples, but it is easier to overfit the specific model parameters, making the transferability less effective [10]. Many methods have been proposed to improve the transferability of adversarial examples, such as momentum [11], diverse input [12], and translation invariance [13]. However, there are also several defence methods to counter the threat of adversarial attacks, such as ensemble adversarial training [14], image denoising, and...
transformation [15, 16], which can significantly improve the robustness of models and make it more challenging to enhance the transferability of the adversarial examples.

Data augmentation [1, 17] strategy shows the effectiveness in preventing deep models from overfitting by applying a set of image transformations in the training process. In addition, [12] demonstrates that applying random resizing and padding to the input images at each iteration can generate more transferable adversarial examples. Because rotation can change image features more than other transformations, deep models can obtain more diverse input features and generate more robust and transferable adversarial examples.

Motivated by the above discussion, in order to enhance the transferability and alleviate the overfitting phenomenon of iterative methods, we proposed a rotation-invariant attack method that maximizes the loss function w.r.t. the random rotated image instead of the original input at each iteration and thus mitigates the effect of high correlation between the adversarial example and the gradient of the source model, making the adversarial example more robust to small transformations and have better transferability. We conduct comprehensive experiments under both single-model and ensemble-based attack settings. Extensive experiments demonstrate that our method can boost adversarial attacks significantly with almost no extra computational cost. In addition, by using our method as an easily applied plug-in, it obtains an average attack success rate of 5.4% higher than the strong baseline method to evade six state-of-the-art robustly trained models under the black-box setting. We show some adversarial examples generated by the proposed method and other baselines in Figure 1.

In summary, our main contributions are as following:

- We propose a rotation-invariant method, which optimizes the perturbations of the randomly rotated images instead of the original input at each iteration and significantly enhances the transferability of the adversarial examples.
- The proposed method mitigates the effect of high correlation between the adversarial examples and the source model, making the resultant adversarial examples more robust to small transformations.
- The proposed method can be integrated into various gradient-based methods to improve the attacks with almost no extra computational cost.

2 | RELATED WORK

2.1 | Adversarial attacks

Szegedy [2] first generated adversarial examples through a box-constrained L-BFGS method, which had a high success rate of attack, but the computation was too expensive due to the second derivative. Goodfellow [3] proposed a one-step fast gradient sign method (FGSM) to generate adversarial examples that linearized the loss around an input to find the directions that the model predictions were most sensitive to. Kurakin [9] extended FGSM to an iterative version that used a small step to iterate multiple times along the ascending direction of the gradient and demonstrated the possibility of adversarial examples in the physical world. Eykholt [18] generated physical adversarial examples robust to widely varying distances/angles, and Jacobsen [19] showed that deep networks are too invariant to a wide range of task-relevant changes, thus making vast regions in input space vulnerable to adversarial attacks. But both of these workers focussed on the white-box setting, whereas black-box attacks are more realistic in practice. Adversarial examples generated by the iterative method have higher attack success rates than the single-step method under the white-box setting, but they usually show poor transferability. Dong [11] proposed a momentum-based iterative algorithm that stabilized the update directions to generate more transferable adversarial examples and studied several approaches to attack an ensemble of models. The ensemble-based approach [7] attacks multiple models simultaneously and can generate much stronger adversarial examples for non-targeted attacks. However, even with the ensemble strategy, the transferability of the targeted adversarial examples generated by the fast gradient-based method was hardly improved. Xie [12] applied random resizing and padding to the input images to enhance adversarial attacks. Dong [13] proposed a translation-invariant method to generate more transferable adversarial examples.

FIGURE 1  Adversarial examples generated by the proposed rotation-invariant method and other baseline methods on the Ine-v3 model
2.2 | Defences

Various defence methods have been proposed to improve the robustness of the deep models to defend against adversarial examples. Meng [20] used a separate detector network and a reformer network to against attacks. Papernot [21] tried to alleviate adversarial examples by defence distillation. Xie [16] and Guo [22] passed the image to a DNN for classification after random transformation to mitigate adversarial effects. Liao [15] removed the adversarial noise by a guided denoiser as a defensive measure. By feeding the adversarial examples back to training [2] or mixing the adversarial objective function with the classification function as a regularizer [3, 10], the target model would learn a robust decision boundary to achieve security. However, because of the coupling between the generation of adversarial examples and the training parameters of the specific model, they do not guarantee robustness against black-box attacks. Tramèr [14] proposed ensemble adversarial training, which augmented the training data with the adversarial examples generated from other models. The ensemble-based approach decoupled the adversarial examples from the specific model and significantly increased the resistance to adversaries.

3 | METHODOLOGY

Let $x$ denote a benign input that can be correctly classified by a classifier $f(\cdot)$ as label $y$. Adversarial examples can be categorized into two types, non-targeted and targeted. A non-targeted adversarial example $x^*$ is crafted by adding imperceptible perturbations to $x$, which leads to misclassification as $f(x^*) \neq y$. A targeted adversarial example $x^*$ satisfies $f(x^*) = y'$, where $y'$ is the targeted label and $y' \neq y$. We use the $L_\infty$ norm to constrain $x^*$ in the vicinity of $x$; that is, the allowed perturbation should be smaller than a threshold $\epsilon$ as $\|x^* - x\|_\infty \leq \epsilon$.

Some methods [6, 23] concentrate on the quality of adversarial noises generated by different attacks on one image; here, we study the generation of transferable adversarial examples with low computational costs and compare the misclassification rate with a fixed $L_\infty$ norm.

To generate non-target adversarial examples, we should maximize the cross-entropy loss function $f(x^*, y)$ of the classifier. We omit network parameters $\theta$ in the loss function because they are fixed. For networks with a softmax output layer, generating the adversarial example solves this constrained optimization problem:

$$\arg \max_{x^*} f(x^*, y), \text{ s.t. } \|x^* - x\|_\infty \leq \epsilon$$

The loss function applied to integer class labels can be expressed as $f(x^*, y) = -1_y \cdot \log(\text{softmax}(f(x^*)))$, where $1_y$ is the one-hot encoding of $y$.

3.1 | Existing adversarial attack methods

In this section, we briefly introduce the mainstream methods used to generate adversarial examples.

Fast gradient sign method: FGSM [3] proposes a single-step gradient-based method, which computes the gradient only once and obtains the gradient directions of perturbations by using the sign function. It can be expressed as

$$x^* = x + \epsilon \cdot \text{sign}(\nabla_x f(x, y))$$

where $\text{sign}(\cdot)$ is the sign function that makes the generated adversarial examples meet the $L_\infty$ distance metric limit. FGSM is usually not effective for white-box attacks, resulting in low success rates of transferability [10].

Iterative fast gradient sign method (I-FGSM): Kurakin [9] extends FGSM to I-FGSM, which performs attack iteratively with a small step size. The update equation is

$$x_{t+1} = \text{Clip}_x\{x_t + \alpha \cdot \text{sign}(\nabla_x f(x_t, y))\}$$

where $\text{Clip}_x\{x'\}$-function performs per-pixel clipping of the image $x'$, which is $\min\{255, x + \epsilon \cdot \max\{0, x - \epsilon, x'\}\}$, so the result will be constrained within $\epsilon$-ball of the original image $x$. $t$ is the $t$-th iteration step, and $\alpha$ is the step size. I-FGSM outperforms FGSM under the white-box setting, but it usually has lower success rates for the black-box models [10].

Momentum iterative fast gradient sign method (MI): MI [11] combines momentum term with I-FGSM to stabilize the update directions and overcome the shortage of trapping into the local maximum, which can be expressed as

$$\begin{align*}
g_{t+1} &= \mu \cdot g_t + \frac{\nabla_x f(x_t^*, y)}{\|\nabla_x f(x_t^*, y)\|_1}, \\
x_{t+1} &= \text{Clip}_x\{x_t^* + \alpha \cdot \text{sign}(g_{t+1})\}
\end{align*}$$

where $g_t$ accumulates the iterated gradient vector of the loss function with a decay factor $\mu$.

Diverse inputs method (DI): DI [12] alleviates the over-fitting phenomenon by applying random and differentiable transformations (resizing and padding) to the input image with a given probability at each iteration and maximizes the loss function w.r.t the transformed image. The stochastic transformation function is

$$D(x_t^*; p) = \begin{cases} 
D(x_t^*) & \text{with probability } p \\
x_t^* & \text{with probability } 1 - p
\end{cases}$$

Translation-invariant method (TI): TI [13] calculates the gradient for the untranslated input image at each step, then convolutes the gradient with a predefined kernel instead of calculating the gradients for a set of translated images, which greatly improves the computational efficiency of attacks. The gradient has the following update rule:

$$\nabla_x f(x_t^*, y) = W \cdot \nabla_x f(x_t^*, y)$$
The TI was originally used to attack defence models, and thus it is not good at attacking normally trained models.

**DeepFool:** DeepFool [24] optimizes the $L_2$ distance metric; it supposes the neural networks are completely linear and that each class can be separated from another by a hyperplane. It analytically derives the optimal solution of the minimal adversarial perturbation $r$ and repeats the process until a true adversarial example is found:

$$\arg\min_r ||r||_2 \text{ s.t. } f(x + r) \neq f(x) \tag{7}$$

**Universal adversarial perturbations (UAP):** UAP [8] seeks the perturbation vectors $v$ that fool the classifier $\hat{k}$ on almost all data points sampled from the distribution of images and seeks the extra perturbation $\Delta v_i$ with a minimal norm that allows the fooling of data point $x_i$ by solving the following optimization problem:

$$\Delta v_i \leftarrow \arg\min_r ||r||_2 \text{ s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i) \tag{8}$$

**Carlini & Wagner method (C&W):** C&W attack [25] is an optimization-based method that minimizes the distance between the benign input and the adversarial example and maximizes the classification loss. Its update procedure is formalized as follows:

$$\arg\min_{x^*} ||x^* - x||_p - c \cdot J(x^*, y) \tag{9}$$

However, similar to I-FGSM, this optimization-based method is not good at black-box attacks [12].

**Expectation over transformation method (EOT):** EOT [26, 27] optimizes the expectation over the transformation $E_{t \sim T}(f(t(x)))$ for defences that employ randomized transformations to the input. It solves the following optimization problem:

$$\arg\max_{x} \left\{ E_{t \sim T} \left[ \log P(y_i \mid t(x^*)) \right] - \lambda E_{t \sim T} [d(t(x^*) \cdot t(x))] \right\} \tag{10}$$

### 3.2 Rotation-invariant attack method

The transfer-based methods first generate adversarial examples under the white-box setting and then leverage the transferability of the resultant adversarial examples to attack the target model under the black-box setting. However, the adversarial examples obtained by iterative methods are easy to be overfitted to the attacked white-box model, which leads to poor transferability. In this section, we propose a rotation-invariant attack method to enhance the transferability of the adversarial examples.

#### 3.2.1 Rotation-invariant loss

Convolutional neural networks (CNNs) are supposed to have rotation-invariant property [28], that an object in the input can be recognized despite its rotation angle. However, CNNs are not truly rotation-invariant, so we assume that the rotation-invariant property is nearly held for the input within a certain rotation angle range (which is empirically validated in Sec. 4.2), and $J(R(x), y) \approx f(x, y)$ will be satisfied for any clean input $x$ under this setting.

We apply randomized rotation $R(\cdot)$ to the input at each iteration to mitigate the effect of the high correlation between the adversarial examples and the source model and make the resultant adversarial examples less sensitive to the specific parameters of the source model. Therefore, the overfitting phenomenon would be alleviated, and the resultant adversarial examples would be more transferable. The constrained optimization problem can be rewritten as

$$\arg\max_{x^*} J(R(x^*), y), \text{ s.t. } ||x^* - x||_\infty \leq \epsilon \tag{11}$$

#### 3.2.2 Rotation input patterns

**Rotation momentum iterative fast gradient sign method (R-MI):** We combine the rotation strategy with MI and apply random rotation $R(\cdot)$ within a given rotation angle range $\beta$ to the inputs at each iteration, the update procedure is

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x^*} J(R(x^*), y)}{\nabla_{x^*} J(R(x^*), y)} ||_1$$

$$x_{t+1}^* = \text{Clip}_x \left\{ x_t^* + \alpha \cdot \text{sign}(g_{t+1}) \right\} \tag{12}$$

Because of the negligible computation involved in the rotation operation, it requires almost no extra computational complexity compared with MI.

We can combine the rotation and diverse input patterns to form a more powerful attack. Moreover, we integrate TI into it and obtain the rotation and diverse input translation-invariant method (R-DI-TI-MI) method, which can be expressed as

$$g_{t+1} = \mu \cdot g_t + \frac{W \cdot \nabla_{x^*} J(f \cdot R(x^*), y) + (1 - \gamma) \cdot D(x^* \cdot p), y)}{W \cdot \nabla_{x^*} J(f \cdot R(x^*), y) + (1 - \gamma) \cdot D(x^* \cdot p), y)} ||_1$$

$$x_{t+1}^* = \text{Clip}_x \left\{ x_t^* + \alpha \cdot \text{sign}(g_{t+1}) \right\} \tag{13}$$

where $W$ is a predefined kernel to blur the gradients, and $\gamma$ is the scaling factor set to balance the effects of the rotation and diverse input. If $\gamma$ is set to 1, the DI term will be invalid; if $\gamma$ is set to 0, the R term will be invalid. If $W$ is set to a $1 \times 1$ scale, it is equivalent to a constant and does not affect the result. The relationships between different attacks are shown in Figure 2.

#### 3.2.3 Ensemble attack algorithm

If adversarial examples can fool multiple models, they may show strong transferability to other models [7], thus attacking an ensemble of models enables more powerful black-box attacks.
Algorithm 1 R-DI-TI-MI for an ensemble of models

Input: A clean image $x$; $K$ classifiers $f_1, f_2, ..., f_K$; ensemble weights $\omega_1, \omega_2, ..., \omega_K$;
Parameter: Perturbation size $\epsilon$; iteration number $N$; momentum decay factor $\mu$; transformation probability $p$; pre-defined kernel $\hat{w}$; rotation angle range $\beta$ and scaling factor $\gamma$.
Output: An adversarial example $x^*$.

1: $\alpha = \epsilon / N$;
2: $g_0 = 0; x_0^* = x$;
3: for $t = 0$ to $N - 1$ do
4: Input $x_t^*$ and output the logits for $k = 1, 2, ..., K$:
   $l_k(x_t^*) = \gamma \cdot l_k(R(x_t^*)) + (1 - \gamma) \cdot l_k(D(x_t^*; p))$;
5: Fuse the logits as $l(x_t^*) = \sum_{k=1}^{K} \omega_k l_k(x_t^*)$;
6: Get the cross-entropy loss $J(x_t^*, y)$ based on $l(x_t^*)$;
7: Obtain the gradient $\nabla_{x_t^*} J(x_t^*, y)$;
8: Update $g_{t+1}$ and $x_{t+1}^*$ by Equation (13);
9: end for
10: return $x^* = x_0^*$.

We follow the ensemble scheme in [11], which suggests ensemble in logits outperforms other ensemble schemes. Attacking $K$ models with logit activations rule can be expressed as

$$l(x) = \sum_{k=1}^{K} w_k l_k(x)$$

where $l_k(x)$ is the logits output of the $k$-th model, $w_k$ is the ensemble weight with $w_k \geq 0$ and $\sum_{k=1}^{K} w_k = 1$.

We summarize the R-DI-TI-MI algorithm for attacking an ensemble of models in Algorithm 1; the other algorithms can be obtained by setting different hyperparameters.

4 | EXPERIMENTS

4.1 | Experimental settings

Data set: It is less meaningful to study the attack success rates if the models cannot correctly classify the original images. Therefore, we randomly choose 5000 images from the ImageNet validation set, and all source models correctly classify these images. All these images are resized to $299 \times 299 \times 3$ beforehand.

Source models: We choose four models: Inception-v3 (Inc-v3) [29], Inception-v4 (Inc-v4), Inception-Resnet-v2 (IncRes-v2) [30], ResNet-v2-152 (Res-152) [31] as the source models, which are unaware of defense strategies.

Target models: We consider 13 target models, seven of which are normally trained models: Inc-v3, Inc-v4, IncRes-v2, Res-152, DenseNet-169 (Dense-169) [32], Xception-71 (Xception-71) [33], and the top-3 models in the NIPS 2017 defense competition: high-level representation guided denoiser (HGD) [15], input transformation through random resizing and padding (R&P) [16], and rank-3 solution$^1$ in the NIPS 2017 defense competition (NIPS-r3).

Baselines: We choose three state-of-the-art gradient-based methods—MI, DI, and TI—and their combinations as the baselines. For the experimental hyperparameter settings, we set the maximum perturbation $\epsilon = 16$ with a pixel value in $[0, 255]$. The total iteration number is set to be $N = 10$. For DI, the transformation probability $p$ is set to 1.0 to prevent result fluctuations due to randomness. The other hyperparameters are set as in their original papers.

Evaluation metrics: The percentage of the adversarial examples generated on one model misclassified by another model is taken as the evaluation metric of the attacks. The percentage is denoted as the black-box attack success rate. A higher black-box attack success rate means better transferability.

4.2 | Rotation-invariant property

In order to verify the rotation-invariant property of CNNs, we use 1000 images from the data set and randomly rotate them in a given rotation angle range. We input the original and the rotated images into the four source models and show the top-1 accuracy with different rotation angle ranges in Figure 3.

As shown in Figure 3, even if the random rotation angle is within the range of $[-20^\circ, 20^\circ]$, the top-1 accuracy does not drop much, the error remains around 0.15.

The losses of the rotated images would be very similar to that of the original images, so we could assume that the rotation-invariant property is almost held within a certain angle range.

4.3 | Effect of the rotation angle range

The rotation angle plays a key role in improving the transferability of the adversarial examples. If the rotation angle is set to 0, the rotation input methods will degrade to their vanilla versions. We conduct an ablation study on 1000 images to find the appropriate angle range.

1. https://github.com/anitha/nips-2017/tree/master/stand
We attack the Inc-v3 model by R-MI, R-DI-MI and R-DI-TI-MI and set absolute values of the angle range from 0° to 20°. To precisely measure the effect of the rotation, the granularity starts small and increases with angle range. We show the attack success rates against three normally trained and three robust models in Figure 4.

We can see that for all black-box attacks, even a small rotation angle range can significantly increase the success rates; for example, in Figure 4(b), when the random rotation angle is within 0.3°, the success rate for Res-152 increases to 63.3%, while the baseline of DI-MI (β = 0) is 57.0%. This phenomenon demonstrates the importance of integrating rotation into the attack methods.

It shows that attack success rates increase at first and then tend to stabilize after the angle range becomes greater than 10°. Therefore, we set the rotation angle range of the input within $\beta \in [-10^\circ, 10^\circ]$ in the following experiments.

### 4.4 Scaling factor $\gamma$

The scaling factor $\gamma$ is set to balance the effects of the rotation and the diverse inputs term. If $\gamma = 0$, R-DI-MI degrades to DI-MI, and R-DI-TI-MI degrades to DI-TI-MI. Therefore, we conduct an ablation experiment to study the appropriate value of $\gamma$; the other hyperparameter settings are described above. We attack Inc-v3 by R-DI and R-DI-TI, and the scaling factor ranges from 0.0 to 1.0 with a granularity of 0.1.

We show the black-box attack success rates against Inc-v4, Res-152, Inc-v3ens4, IncRes-v2ens, and HGD in Figure 5. The success rate curves are almost unimodal, and all black-box target models share the same best selection of $\gamma$, which makes the $\gamma$ selection quite simple. Therefore, we set $\gamma = 0.5$ in the following experiments.

![Figure 3](image3.png)

**Figure 3** The top-1 accuracy of four source models for the input with different random rotation angle range

![Figure 4](image4.png)

**Figure 4** The attack success rates (%) of the adversarial examples generated on Inc-v3 against six models with the absolute rotation angle range varying from 0° to 20°. The starting points (when $\beta = 0$) of all curves are the results of corresponding baseline methods. It can be seen that when $\beta > 0$, the black-box attack success rates increase significantly even within a small rotation angle range

![Figure 5](image5.png)

**Figure 5** The black-box attack success rates (%) of the adversarial examples generated on Inc-v3 against five other models with $\gamma$ ranging from 0.0 to 1.0
4.5 | Single-model attacks

In this section, we attack each of the four normally trained models and test the attack success rates and computational costs of 5000 images on all 13 models and compare the results of the rotation-invariant method with the baseline attacks.

We first average losses over all the adversarial examples generated by attacking Inc-v3 or IncRes-v2 against an ensemble of the three ensemble adversarially trained models at each iteration and compare the average losses of MI, DI-MI, and our R-MI.

We can see from Figure 6 that the losses of our R-MI grow faster than MI and DI-MI. According to Equation 1, it can be inferred that the attack success rates of our method are higher. In other words, the larger the loss, the larger the categorization difference between the adversarial examples and the original inputs, and the higher the attack success rate. We can also infer that to attack a black-box model with a required success rate, R-MI can use a smaller perturbation, which would be more indistinguishable for humans.

We then attack Inc-v3, Inc-v4, IncRec-v2, and Res-152, respectively, using MI, DI-MI, and our R-MI, and test the attack success rates against seven normally trained models. Because TI attack is more effective for attacking the robustly trained models, we combine it with other methods and obtain the TI-MI, DI-TI-MI, and R-TI-MI to attack the six robustly trained models, where DI-TI-MI is current the strongest attack against the robustly trained models [13]. The results of attack success rates against the normally trained and the robustly trained models under the single-model setting are summarized in Table 1 and Table 2, respectively.

Our work achieves significantly higher attack success rates in most settings by leveraging the random rotation input strategy. Because the computational cost involved in the rotation operation is negligible, our method requires almost no extra computational complexity.

Unlike the ensemble adversarially trained models that augment the training data with adversarial examples generated from other models, the NIPS top-3 defence methods apply image transformations for defending against adversarial examples. From Table 2, we can see that for the transformation defence models, the proposed rotation input method also superior to the baseline methods, indicating that the generated adversarial examples are more robust to small transformations.

**FIGURE 6** The average losses of adversarial examples generated on (a) Inc-v3 or (b) IncRes-v2 against an ensemble of the three ensemble adversarially trained models at each iteration using MI, DI-MI, and our R-MI.

**TABLE 1** The attack success rates (%) against seven normally trained models under the single-model setting.

| Model      | Attack   | Inc-v3 | Inc-v4 | IncRes-v2 | Res-152 | Dense-169 | Xcep-71 | PNAS | Time (s) |
|------------|----------|--------|--------|-----------|---------|-----------|---------|------|----------|
| Inc-v3     | MI [11]  | 100.0* | 48.4   | 45.7      | 36.8    | 46.0      | 43.6    | 33.6 | 853.8    |
|            | DI-MI [12]| 99.2*  | 72.9   | 67.3      | 57.5    | 68.6      | 66.0    | 58.1 | 889.6    |
|            | R-MI (ours)| 99.5*  | 78.2   | 72.1      | 62.1    | 72.5      | 69.1    | 64.3 | 896.6    |
| Inc-v4     | MI       | 65.6   | 100.0* | 51.7      | 44.6    | 59.1      | 56.7    | 49.9 | 1311.6   |
|            | DI-MI    | 80.3   | 99.0*  | 71.4      | 62.7    | 77.1      | 75.5    | 70.4 | 1453.9   |
|            | R-MI (ours)| 82.9*  | 99.0*  | 74.4      | 65.4    | 78.6      | 76.3    | 73.5 | 1354.6   |
| IncRes-v2  | MI       | 68.7   | 61.0   | 99.5*     | 50.9    | 58.7      | 52.9    | 48.6 | 1346.0   |
|            | DI-MI    | 81.0   | 79.5   | 96.0*     | 69.0    | 75.2      | 72.6    | 70.6 | 1522.8   |
|            | R-MI (ours)| 82.5   | 80.3   | 96.0*     | 70.7    | 76.3      | 72.8    | 72.5 | 1518.7   |
| Res-152    | MI       | 53.2   | 47.1   | 45.3      | 98.6*   | 58.8      | 48.3    | 42.1 | 1518.4   |
|            | DI-MI    | 78.2   | 75.7   | 73.4      | 98.1*   | 80.6      | 72.2    | 69.7 | 1764.8   |
|            | R-MI (ours)| 76.6   | 73.2   | 71.1      | 97.9*   | 78.0      | 69.4    | 67.8 | 1599.1   |

Note: The adversarial examples are generated on Inc-v3, Inc-v4, IncRes-v2, and Res-152, respectively, using MI, DI, and R-MI.

Abbreviation: MI, Momentum iterative.

*Indicates the white-box attacks. The best results are in bold.
Table 2  The black-box attack success rates (%) against six robustly trained models under the single-model setting

| Model | Attack  | Inc-v3  | Inc-v4  | IncRes-v2  | HGD   | R&P   | NIPS-r3 | Time (s) |
|-------|---------|---------|---------|------------|-------|-------|--------|---------|
| Inc-v3 | TI-MI [13] | 30.1    | 28.5    | 19.7       | 20.1  | 17.0  | 21.0   | 923.6   |
|        | DI-TI-MI [13] | 40.3    | 39.4    | 28.7       | 31.7  | 29.0  | 33.4   | 977.4   |
|        | R-TI-MI (ours) | **46.3** | **45.3** | **32.8**   | **35.4** | **32.1** | **38.8** | **982.2** |
| Inc-v4 | TI-MI   | 32.3    | 31.0    | 23.5       | 24.0  | 21.6  | 24.7   | 1354.8  |
|        | DI-TI-MI | 41.9    | 40.5    | 31.3       | 35.5  | 32.2  | 35.8   | 1541.4  |
|        | R-TI-MI (ours) | **46.5** | **44.3** | **34.6**   | **38.1** | **34.6** | **40.2** | **1459.5** |
| IncRes-v2 | TI-MI   | 44.0    | 41.1    | 39.6       | 37.0  | 35.8  | 38.8   | 1399.3  |
|        | DI-TI-MI | 52.9    | 50.2    | 49.9       | 49.5  | 50.9  | 52.1   | 1513.4  |
|        | R-TI-MI (ours) | **56.4** | **53.7** | **53.8**   | **51.9** | **52.4** | **55.4** | **1484.6** |
| Res-152 | TI-MI   | 34.8    | 33.7    | 27.8       | 27.3  | 25.8  | 29.7   | 1589.1  |
|        | DI-TI-MI | 51.8    | 49.9    | 43.6       | 44.9  | 42.4  | 46.2   | 1781.1  |
|        | R-TI-MI (ours) | 51.5    | 49.6    | 43.0       | 43.8  | 40.8  | 46.8   | 1600.6  |

Note: The adversarial examples are generated on Inc-v3, Inc-v4, IncRes-v2, and Res-152, respectively, using TI-MI, DI-TI-MI, and R-TI-MI. The best results are in bold. Abbreviation: HGD, high-level representation guided denoiser.

Figure 7  Average black-box attack success rates (%) against the normally trained and the robustly trained models. The adversarial examples are generated on the four single-source models using different attack methods under the single-model setting.

We further average the black-box attack success rates of Table 1 and Table 2 and show the results in Figure 7. We can see more intuitively that our method’s average attack success rates are about 3% higher than that of the corresponding DI baseline method for the normally trained and the robustly trained models, which validates the effectiveness of applying random rotation to the input for generating more transferable adversarial examples.

4.6  Ensemble-based attacks

The attack success rates of the generated adversarial examples are significantly improved, but they are still relatively weak, especially when attacking the robustly trained models. Liu [7] has demonstrated that ensemble-based attacks perform better on generating transferable adversarial examples than single-model attacks. Therefore, we attack an ensemble of models to further enhance the transferability of the generated adversarial examples.

We adopt the logits ensemble scheme in [11] and generate the adversarial examples by attacking an ensemble of the four source models (Ens4) with equal weights using different methods. Note that the TI method is originally used to attack robustly trained models, although here we use it to attack both normally and robustly trained models. The ensemble attack success rates against the three normally trained and six robustly trained models are summarized in Table 3 and Table 4, respectively.

It can be seen that our proposed rotation-invariant method consistently outperforms the baseline methods, and it requires almost no extra computational consumption. This observation shows that the proposed method is better at learning transferable adversarial examples.

Moreover, the proposed random rotation strategy can be easily combined with any other method to further improve the

Table 3  The black-box attack success rates (%) of adversarial examples crafted by different methods on an ensemble of four source models against three normally trained models

| Attack  | Dense-169 | Xcep-71 | PNAS | Average | Time (s) |
|---------|-----------|---------|------|---------|---------|
| TI-MI   | 75.4      | 73.0    | 75.1 | 74.5    | 5299.73 |
| DI-TI-MI| 81.9      | 79.4    | 79.1 | 80.1    | 5444.04 |
| R-TI-MI (ours) | **83.5** | **80.9** | **81.6** | **82.0** | 5727.98 |
| MI      | 85.9      | 84.5    | 83.8 | 84.7    | 5211.81 |
| DI-MI   | 93.1      | 92.6    | 91.5 | 92.4    | 5599.82 |
| R-MI (ours) | **94.7** | **93.7** | **92.9** | **93.8** | 6059.02 |
| R-DI-MI | **94.9** | **94.2** | **93.4** | **94.2** | 12,328.86 |

Note: The best results are in bold.
attack success rate. The average attack success rate of DI-MI against the normally trained models increased from 92.4% to 94.2%; the average attack success rate of the state-of-the-art DI-TI-MI method against the six robustly trained models increased from 69.4% to 74.8% by integrating the rotation strategy into it, and the improvement is 5.4%. These results demonstrate the effectiveness and efficiency of the proposed method.

4.7 Further analysis

Perturbation comparison: We measure the perturbation between the adversarial and original images by root mean square deviation, which is computed by
\[
 d(x', x) = \sqrt{\frac{1}{n} \sum_{i}(x'_i - x_i)^2}
\]
between \(x'\) and \(x\), where \(n\) is the dimensionality of the input images, and \(i\) denotes the \(i\)-th dimension of the image. The maximum perturbation is \(\epsilon = 16\) with pixel value in \([0, 255]\).

It can be seen from Table 5 that although our rotation method significantly improves the attack success rates, the magnitude of the perturbations is almost the same as that of the baseline methods. We also visualize two randomly selected original images and their adversarial counterparts generated on the Inc-v3 by our method and other baseline methods in Figure 1.

Comparison with optimization-based attacks: To further verify the effectiveness of our method, we have made more comprehensive comparisons with four recent optimization-based methods in adversarial attacks, including DeepFool, UAP, C&W, and EOT.

| TABLE 4 | The black-box attack success rates (%) of adversarial examples crafted by different methods on an ensemble of four source models against six robustly trained models |
|------------|-----------------------------------------------|
| Attack     | Inc-v3\(_{ens}\) | Inc-v3\(_{ens}\) | IncRes-v2\(_{ens}\) | HGD | R&P | NIPS-r3 | Average | Time (s) |
| MI         | 44.5          | 38.8          | 25.0          | 33.9 | 25.3 | 37.7    | 34.2    | 5211.81 |
| DI-MI      | 58.0          | 52.0          | 35.9          | 49.7 | 41.7 | 55.2    | 48.7    | 5599.82 |
| R-MI (ours) | 65.0          | 60.0          | 42.7          | 55.9 | 47.7 | 63.2    | 55.8    | 6059.02 |
| TI-MI      | 68.8          | 66.5          | 61.4          | 64.9 | 59.4 | 63.5    | 64.1    | 5299.73 |
| DI-TI-MI   | 72.7          | 70.5          | 65.8          | 69.6 | 68.1 | 69.6    | 69.4    | 5444.04 |
| R-TI-MI (ours) | 76.9          | 75.7          | 70.0          | 73.1 | 70.6 | 73.9    | 73.4    | 5727.98 |
| R-DI-TI-MI | 77.6          | 76.2          | 72.1          | 75.0 | 72.9 | 75.1    | 74.8    | 11,575.38 |

Note: The best results are in bold.

| TABLE 5 | The average root mean square deviation of all adversarial images generated on different models, it can be seen that the rotation strategy does not increase the perturbation size |
|------------|-----------------------------------------------|
| Attack     | Inc-v3 | Inc-v4 | IncRes-v2 | Res-152 | Dense-169 | Xcep-71 | PNAS | Time (s) |
| MI         | 11.97  | 12.05  | 12.07     | 11.93   | 11.90     |        |      |        |
| DI-MI      | 12.01  | 12.06  | 12.08     | 12.04   | 12.08     |        |      |        |
| R-MI (ours) | 12.01  | 12.04  | 12.07     | 12.04   | 12.01     |        |      |        |
| TI-MI      | 11.70  | 11.78  | 11.92     | 11.79   | 11.69     |        |      |        |
| DI-TI-MI   | 12.02  | 12.04  | 12.14     | 12.13   | 12.14     |        |      |        |
| R-TI-MI (ours) | 12.04  | 12.03  | 12.13     | 12.10   | 12.15     |        |      |        |

Note: The best results are in bold.

| TABLE 6 | The attack success rates (%) against the normally trained models under the single-model setting |
|------------|-----------------------------------------------|
| Attack     | Inc-v3 | Inc-v4 | IncRes-v2 | Res-152 | Dense-169 | Xcep-71 | PNAS | Time (s) |
| DeepFool   | 93.5\(^a\) | 4.8    | 3.3     | 3.4    | 4.9      | 6.5     | 3.7   | 7492.9 |
| UAP        | 84.7\(^a\) | 4.9    | 4.0    | 5.3    | 9.0      | 8.9     | 3.2   | 8108.6 |
| C&W        | 100.0\(^a\) | 9.3    | 8.5    | 7.3    | 10.6     | 11.9    | 4.3   | 122,030.2 |
| EOT        | 99.1\(^a\) | 7.8    | 6.4    | 5.7    | 7.1      | 7.5     | 4.2   | 55,012.8 |
| R-MI (ours) | 99.5\(^a\) | 78.2   | 72.1   | 62.1   | 72.5     | 69.1    | 64.3  | 896.6  |

The adversarial examples are generated on Inc-v3, and the time required to compute 5000 adversarial examples for each method is given in the time column.

Abbreviations: EOT, Expectation over transformation method; UAP, Universal adversarial perturbations.

\(^a\)Indicates the white-box attacks.
The results of adversarial examples generated on Inc-v3 are reported in Table 6. Under the black-box setting, our method significantly outperforms all four optimization-based methods. It may be because adversarial examples generated by the optimization-based methods tend to overfit the white-box model and are unlikely to transfer across models.

Computational cost comparison: All the experimental results are computed on a GTX 1080Ti GPU. It can be seen from Tables 1–4 that for generating 5000 adversarial examples, the proposed rotation method requires almost no extra computational cost. Moreover, we show the running time for generating 5000 adversarial examples on the Inc-v3 model in the time column of Table 6. We can observe that the optimization-based methods are time-consuming, taking orders of magnitude more time than our fast gradient-based method.

5 CONCLUSION

We proposed enhancing adversarial attacks by a rotation-invariant method that maximizes the loss function w.r.t the random rotated image instead of the original input at each iteration, thus mitigating the effect of high correlation between the adversarial examples and the source model and making the adversarial examples more transferable. Experiment results demonstrate that the generated adversarial examples exhibit much better transferability against both the normally and the robustly trained models with almost no extra computational cost. In addition, simply integrating our proposed rotation method into any baseline method can achieve a much higher attack success rate. Our method can be used to better evaluate the robustness of various deep learning models and the efficacy of potential defences.

ACKNOWLEDGMENT

National Natural Science Foundation of China 62,076,251.

ORCID

Yexin Duan https://orcid.org/0000-0003-0769-4773

REFERENCES

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105. Lake Tahoe, Nevada (2012)
2. Szegedy, C., et al.: Intriguing properties of neural networks. 2nd International Conference on Learning Representations. ICLR, Banff, Canada (2014)
3. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. 3rd International Conference on Learning Representations. ICLR, San Diego, CA (2015)
4. Xie, C., et al.: Adversarial examples for semantic segmentation and object detection. Proceedings of the IEEE International Conference on Computer Vision. pp. 1369–1378. Venice, Italy (2017)
5. Chen, P.Y., et al.: Zoo: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. pp.15–26. Dallas, TX (2017)
6. Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: reliable attacks against black-box machine learning models. 6th International Conference on Learning Representations. ICLR, Vancouver, Canada (2018)
7. Liu, Y., et al.: Delving into transferable adversarial examples and black-box attacks. 5th International Conference on Learning Representations. ICLR, Toulon, France (2017)
8. Seyed-Mohsen, M.D., et al.: Universal adversarial perturbations. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1765–1773. Honolulu, HI (2017)
9. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. 5th International Conference on Learning Representations. ICLR, Toulon, France (2017a)
10. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. 5th International Conference on Learning Representations. ICLR, Toulon, France (2017b)
11. Dong, Y., et al.: Boosting adversarial attacks with momentum. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9185–9193. CVPR, Salt Lake City, UT (2018)
12. Xie, C., et al.: Improving transferability of adversarial examples with input diversity. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2730–2739. CVPR, Long Beach, CA (2019)
13. Dong, Y., et al.: Evading defenses to transferable adversarial examples by translation-invariant attacks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4312–4321. CVPR, Long Beach, CA (2019)
14. Tramèr, F., et al.: Ensemble adversarial training: attacks and defenses. 6th International Conference on Learning Representations. ICLR, Vancouver, Canada (2018)
15. Liu, Y., et al.: Delving into transferable adversarial examples and black-box attacks. 5th International Conference on Learning Representations. ICLR, San Diego, CA (2015)
16. Eykholt, K., et al.: Robust physical-world attacks on deep learning visual classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1625–1634. Salt Lake City, UT (2018)
17. Jacobson, J.H., et al.: Excessive Invariance Causes Adversarial Vulnerability. arXiv preprint arXiv:1811.00401 (2018)
18. Xie, C., et al.: Mitigating adversarial effects through randomization. 6th International Conference on Learning Representations. ICLR, Vancouver, Canada (2018)
19. Simonov, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations. ICLR, San Diego, CA (2015)
20. Guo, C., et al.: Countering adversarial images using input transformations. 6th International Conference on Learning Representations ICLR, Vancouver, Canada (2018)
21. Papernot, N., et al.: Distillation as a defense to adversarial perturbations against deep neural networks. 2016 IEEE Symposium on Security and Privacy (SP), pp. 582–597. IEEE, San Jose, CA (2016)
22. Dong, Y., et al.: Boosting adversarial attacks with momentum. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9185–9193. CVPR, Salt Lake City, UT (2018)
23. Karras, T., et al.: A style-based generator architecture for generative adversarial networks. Proceedings of the 2019 Conference on Neural Information Processing Systems. pp. 12908–12919. NeurIPS, Vancouver, Canada (2019)
24. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. 5th International Conference on Learning Representations. ICLR, Toulon, France (2017)
25. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. IEEE Symposium on Security and Privacy (SP) pp. 39–57. IEEE, San Jose, CA (2017)
26. Athalye, A., et al.: Synthesizing robust adversarial examples. International Conference on Machine Learning (ICML). pp. 284–293. PMLR, Stockholm, Sweden (2018)
27. Athalye, A., Carlini, N., Wagner, D.: Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples (2018). arXiv preprint arXiv:1802.00420

28. Cohen, T., Welling, M.: Group equivariant convolutional networks. International conference on machine learning. pp. 2990–2999. New York, NY (2016)

29. Szegedy, C., et al.: Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826. Las Vegas, NV (2016)

30. Szegedy, C., et al.: Inception-v4, inception-resnet and the impact of residual connections on learning. Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA (2017)

31. He, K., et al.: Identity mappings in deep residual networks. European conference on computer vision. pp. 630–645. Springer, Amsterdam, The Netherlands (2016)

32. Huang, G., et al.: Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708. Honolulu, HI (2017)

33. Chollet, F.: Xception: deep learning with depthwise separable convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258. Honolulu, HI (2017)

34. Liu, C., et al.: Progressive neural architecture search. Proceedings of the European Conference on Computer Vision, pp. 19–35. ECCV, Munich, Germany (2018)

How to cite this article: Duan Y, Zou J, Zhou X, Zhang W, Zhang J, Pan Z. Enhancing transferability of adversarial examples via rotation-invariant attacks. IET Comput. Vis. 2021;1–11. https://doi.org/10.1049/cvi2.12054