Boosting Adversarial Transferability via Gradient Relevance Attack

Hegui Zhu[1], Yuchen Ren[1]*, Xiaoyan Sui[1], Lianping Yang[1], Wuming Jiang[2]
[1]College of Sciences, Northeastern University, Shenyang, China
[2]Beijing EyeCool Technology, Beijing, China
{zhuhegui,yanglp}@mail.neu.edu.cn,{ryc_981015,suixy0926}@163.com,{jiangwuming}@eyecool.cn

Abstract

Plentiful adversarial attack researches have revealed the fragility of deep neural networks (DNNs), where the imperceptible perturbations can cause drastic changes in the output. Among the diverse types of attack methods, gradient-based attacks are powerful and easy to implement, arousing wide concern for the security problem of DNNs. However, under the black-box setting, the existing gradient-based attacks have much trouble in breaking through DNN models with defense technologies, especially those adversarially trained models. To make adversarial examples more transferable, in this paper, we explore the fluctuation phenomenon on the plus-minus sign of the adversarial perturbations' pixels during the generation of adversarial examples, and propose an ingenious Gradient Relevance Attack (GRA). Specifically, two gradient relevance frameworks are presented to better utilize the information in the neighborhood of the input, which can correct the update direction adaptively. Then we adjust the update step at each iteration with a decay indicator to counter the fluctuation. Experiment results on a subset of the ILSVRC 2012 validation set forcefully verify the effectiveness of GRA. Furthermore, the attack success rates of 68.7% and 64.8% on Tencent Cloud and Baidu AI Cloud further indicate that GRA can craft adversarial examples with the ability to transfer across both datasets and model architectures. Code is released at `https://github.com/RYC-98/GRA`.

1. Introduction

Deep neural networks (DNNs) have made numerous achievements [13, 14, 10, 3, 32, 6, 35]. However, especially in the computer vision field, recent researches on the model robustness verify that DNNs are extremely susceptible to human-imperceptible malicious perturbations [1, 11, 31, 4], attracting many researchers to dive into the generation of adversarial examples [7, 5, 22, 20]. Furthermore, crafting adversarial examples with strong transferability can expose security defects and explore the inner mechanism of current DNNs [11, 23, 33], and it has become a vital task in computer vision.

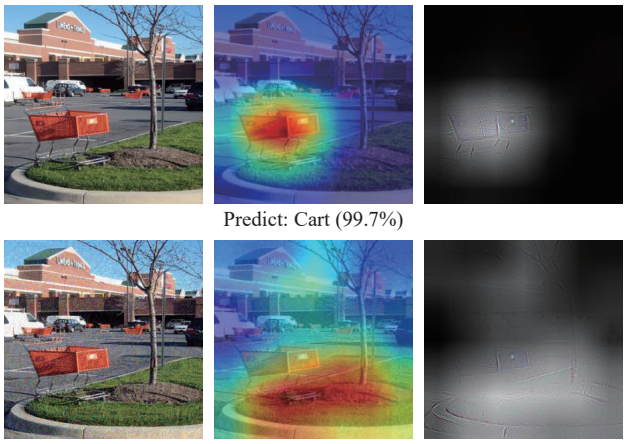

Figure 1. Attention maps [28] of a clean image and its adversarial example crafted by GRA on Inc-v3. The target model is Res-152.

From the perspective of the attackers' knowledge, there are two types of attack settings, *i.e.*, black-box setting and white-box setting. In the white-box setting, all information about the target model can be acquired by the attackers, and many previous attack methods can already hoax the source model with a nearly 100% attack success rate under this setting [16]. Conversely, in the black-box setting, only the model output is available, which will often degrade the attacking performance on target models, especially for models with defense mechanisms [26, 15, 12, 21, 24]. To deal with this issue, diverse gradient-based attack methods [18, 7, 9, 36, 39], input augmentation transformations [8, 38, 37], and ensemble strategy [19] are presented in recent years. Among them, variance tuning (VT) [36] is one of the most promising attack methods, which introduces neighborhood information of the input at the last iteration to stabilize the current update direction. Unfortunately, it ignores the gradient relevance between the input and its neigh-

*Corresponding author: Yuchen Ren

borhood, failing to make full use of the neighborhood information.

In this research, we propose a new gradient-based attack named Gradient Relevance Attack (GRA). An example is provided to show the misdirection capacity of GRA in Figure 1. Concretely, inspired by the framework of dot-product attention [2, 34], we first devise two gradient relevance frameworks to dig out neighborhood information. We view the current gradient as the query vector [34] and the gradients calculated from the neighborhood as the key vectors [34], then establish relevance between them through cosine similarity. With the inner relevance information, the update direction is determined by a group of samples' gradients adaptively.

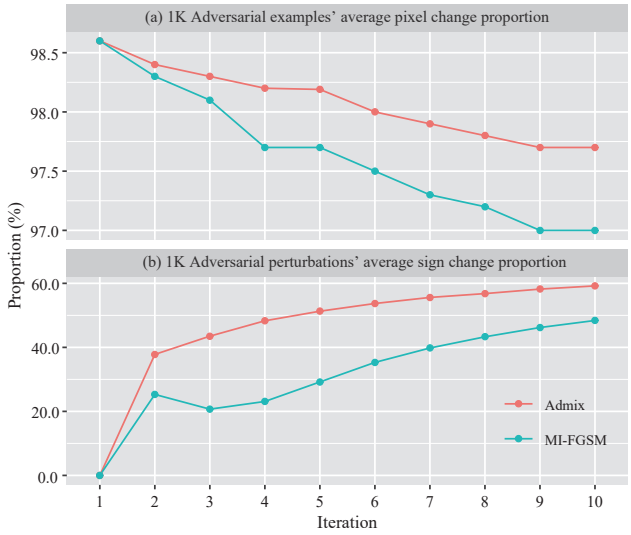

Figure 2. The illustration for two kinds of pixel change tendency: (a) Adversarial examples' pixel value changes; (b) Adversarial perturbations' sign changes. Note that adversarial examples are crafted on Inc-v3.

Besides, we calculate the mean pixel changes of the adversarial examples compared with their clean images on a subset (1k) of ILSVRC 2012 validation set [27]. Five popular gradient-based attacks (MI-FGSM [7], NI-FGSM [18], VTMI-FGSM, VTNI-FGSM [36] and Admix [37]) are taken into consideration. The result indicates the mean pixel changes are all between 10 and 11 under the maximum constraint $\varepsilon = 16$ and the maximum iteration number $T = 10$. Current methods typically add adversarial perturbations with the magnitude of ε/T on the input at each iteration to craft adversarial examples. We conclude two reasons may result in this fact. One is that many pixels' values remain unchanged after certain iterations, and we name it the early stop. While the other is caused by the frequent plus-minus sign changes on the pixels of adversarial perturbations (we simply call it the adversarial perturbations' sign changes in the following context without ambiguity).

To figure out the real reason, we study the adversarial examples' pixel changes and adversarial perturbations' sign changes between two adjacent iterations. Admix and MI-FGSM are selected as examples, their results are displayed in Figure 2. Figure 2 (a) shows that more than 95% pixels keep changing from the beginning to the end, therefore, the first early stop is impossible. Figure 2 (b) certifies the frequent fluctuation of the sign (see Figure 3), because more than half of the adversarial perturbations' signs are changing even at the end of the generation. In fact, the fluctuation phenomenon in adversarial perturbations' sign is not always bad, because it can help us find the optimum. Whereas the step size is fixed during the generation of adversarial examples, and it keeps us from getting closer to the optimum when facing frequent fluctuation. Consequently, we further integrate a decay indicator to adjust the step size and counter the fluctuation. Combining MI-FGSM with the gradient relevance framework and decay indicator, we propose the gradient relevance attack (GRA).

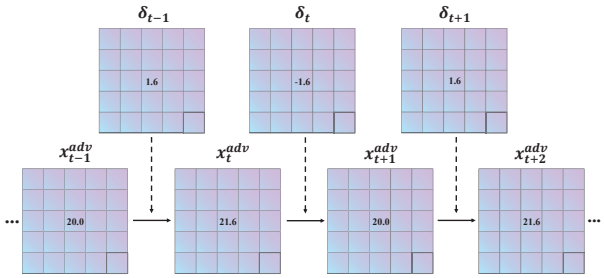

Figure 3. The illustration for adversarial perturbations' sign changes. The x_t^{adv} and δ_t are input and adversarial perturbations at the t-th iteration severally.

Experiment results persuasively verify that GRA has better performance than other advanced attacks, and becomes the state-of-the-art gradient-based attack method. For example, the average attack success rate of our method can reach 83.0% on models with defense technologies [33, 26, 15, 12, 21, 24], and it achieves at least 12.3.% improvement over other advanced attack methods.

Our main contributions are summarized as follows:

- We explore the fluctuation on adversarial perturbations' plus-minus sign during the generation of adversarial examples, and devise a decay indicator of the step size to counter the fluctuation.
- We present two kinds of gradient relevance frameworks, which can make full use of the neighbor information by establishing the gradient relevance between the input and its neighborhood at each iteration.
- We propose an ingenious Gradient Relevance Attack (GRA) combing with current input augmentation transformations, which can boost the transferability of adversarial examples largely.

- Comprehensive experiments on normal classification classifiers, defended classifiers and practical online classifiers verify that GRA is superior to the latest state-of-the-art gradient-based attacks.

2. Related Work

2.1. Gradient-based Attack Method

The gradient-based attack is a mighty kind of attack method, which adds perturbation to the clean image along the gradient's sign to confuse the classifier. Goodfellow *et al*. [11] present the fast gradient sign method (FGSM) to generate the adversarial example and find the linear property of networks. However, FGSM obtains the adversarial examples with only one iteration, which is underfitting. To solve this issue, Kurakin *et al*. [16] construct the iterative version of FGSM called I-FGSM by increasing the number of iterations. Momentum iterative fast gradient sign method (MI-FGSM) [7] follows the idea of gradient descent with momentum [25] to reduce the volatility in the update direction. Nesterov Iterative Fast Gradient Sign Method (NI-FGSM) [18] improves MI-FGSM by taking an extra step at each iteration. Variance Tuning (VT) [36] utilizes the gradient information obtained at the last iteration to correct the current gradient and it can be integrated into MI-FGSM and NI-FGSM.

2.2. Input Augmentation Transformation

Diverse input (DI) [38] enhances input images through a combination of two transformations, *i.e.*, random padding and resizing with a constant probability, and then it sends the processed images to craft adversarial examples. Translation-Invariance (TI) [8] calculates the average gradient on the translated input images at each iteration, and its author proves that the process above can be approximately calculated with a special kernel matrix directly convolving the gradients of the input. Scale-Invariant (SI) [18] puts forward the scale-invariant property of the deep neural network, and then gets the average gradient over scaled images to introduce extra foreign gradient information when producing adversarial examples. Admix [37] mixes the input image with other images randomly selected in the same batch to augment the input, and then updates with gradients calculated on the mixed image.

2.3. Adversarial Defense

Similar to the effect of vaccines, adversarial training [11, 23, 33] notably improves the robustness of models by extending the training dataset with crafted adversarial examples. Whereas adversarial training is hard to extend to the complex models [17]. Except for adversarial training, there are other defense methods that are simple to implement. Guo *et al*. [12] apply diverse non-differentiable transformations (*e.g.*, JPEG compression) to the input images and increase the prediction accuracy when faced with adversarial examples. Naseer *et al*. [24] eliminate the malicious perturbations with a prearranged neural representation purifier (NRP) which is an automatically derived supervision. ComDefend [15] defends the adversarial examples by feeding them into an end-to-end image compression model which can partly alleviate the malicious perturbations on the image. Feature distillation (FD) [21] purifies the adversarial input perturbations by redesigning the image compression framework, which is a novel low-cost strategy. Pixel deflection (PD) [26] can allay the malicious perturbations effectively with pixel corruption and redistribution.

3. Approach

3.1. Preliminary

Our task is to find an adversarial example $x^{adv} = x^{clean} + \delta$ that can hoax the target classifier F_θ with parameter θ and satisfy the given constraint:

$$F_\theta(x^{clean}) \neq F_\theta(x^{adv}), s.t. \|\delta\|_\infty < \varepsilon, \tag{1}$$

where x^{clean} is the clean image, δ is the malicious perturbation, and ε is the maximum magnitude of malicious perturbation under the infinite norm [7, 8, 18, 38, 36, 37]. When all information about F_θ is transparent, the process above can be considered as an optimization problem to search an adversarial example x^{adv} as:

$$\arg\max_{x^{adv}} L(x^{adv}, y^{true}), \tag{2}$$

where L is the loss function and y^{true} is the true label of x^{clean}. Whereas, in most cases, we can only acquire the outputs of F_θ in the black-box setting. Therefore, it is typical to attack the target F_θ with adversarial examples generated on another source model F_ψ with parameter ψ, and the ability to successfully deceive another model is called adversarial transferability.

3.2. Gradient Relevance Framework

Variance tuning [36] modifies the current gradient with gradient variance computed in the neighborhood at the last iteration, which can promote the adversarial transferability to a new level. But we argue that the gradient variance at the last iteration can't reflect the variation trend of the loss function exactly at the current iteration. We hence correct the current gradient with neighbor information at the current iteration in a new way.

Let x_t^{adv} represent the input at the t-th iteration, $x_t^i = x_t^{adv} + \gamma_t^i$ denote the sampled image nearby x_t^{adv}, where $i = 1, 2, ..., m$, and m is the sample quantity. Here γ_t^i is the i-th random noise satisfying $\gamma_t^i \sim U\left[-(\beta \cdot \varepsilon)^d, (\beta \cdot \varepsilon)^d\right]$,

where $\beta \cdot \varepsilon$ is the upper bound of the random noise's magnitude, β is the upper bound factor, $U[,]$ is the uniform distribution and d is the dimension.

Our purpose is to seek the inner relevance between the current gradient $G_t(x)$ calculated on x_t^{adv}:

$$G_t(x) = \nabla_{x_t^{adv}} L(x_t^{adv}, y^{true}), \tag{3}$$

and the gradient $G_t^i(x)$ calculated on x_t^i:

$$G_t^i(x) = \nabla_{x_t^i} L(x_t^i, y^{true}). \tag{4}$$

Inspired by the framework of dot-product attention [34], we treat the current gradient $G_t(x)$ calculated on x_t^{adv} as a query vector and the gradients $G_t^i(x)$ calculated nearby x_t^{adv} as the key vector, then establish a relevance among them with cosine similarity and output the individually weighted gradient WG_t^i by:

$$\begin{cases} s_t^i = \frac{G_t(x) \cdot G_t^i(x)}{\|G_t(x)\|_2 \cdot \|G_t^i(x)\|_2}, \\ WG_t^i = s_t^i \cdot G_t + (1 - s_t^i) \cdot G_t^i. \end{cases} \tag{5}$$

Finally, we obtain the global weighted gradient WG_t by

$$WG_t = \frac{1}{m} \sum_{i=1}^{m} WG_t^i. \tag{6}$$

The individual gradient relevance framework is shown in Figure 4 (a). It can be found that this framework is derived from the dot-product attention [34], and we need to calculate the similarity between G_t and each G_t^i m times at each iteration. To be more efficient, we put forward another average gradient relevance framework in Figure 4 (b). Instead of calculating similarity with all the nearby gradients, we directly establish the relevance with their average gradient, which only needs to calculate the similarity once at each iteration. The average gradient $\overline{G_t}(x)$ is defined as:

$$\overline{G_t}(x) = \frac{1}{m} \sum_{i=1}^{m} G_t^i(x) = \frac{1}{m} \sum_{i=1}^{m} \nabla_{x_t^i} L(x_t^i, y^{true}), \tag{7}$$

and the average gradient relevance framework can be written as:

$$\begin{cases} s_t = \frac{G_t(x) \cdot \overline{G_t}(x)}{\|G_t(x)\|_2 \cdot \|\overline{G_t}(x)\|_2}, \\ WG_t = s_t \cdot G_t + (1 - s_t) \cdot \overline{G_t}. \end{cases} \tag{8}$$

It's worth noting that both the two relevance frameworks above can be integrated with MI-FGSM [7]. Taking the average gradient relevance framework as an example, after obtaining the global weighted gradient WG_t by Eq.(8), the momentum accumulation g_{t+1} in MI-FGSM can be expressed as:

$$g_{t+1} = \mu \cdot g_t + \frac{WG_t}{\|WG_t\|_1}, \tag{9}$$

where μ is the decay factor of the momentum accumulation.

The idea of the average gradient framework is simple and the average gradient $\overline{G_t}(x)$ contains the update information near the input which can be regarded as an auxiliary correction term. When the current gradient G_t is similar to $\overline{G_t}(x)$, we allocate a large weight to G_t and a small weight to $\overline{G_t}(x)$, because G_t doesn't need much correction in this case. If they differ widely, we prefer to give $\overline{G_t}(x)$ a large weight, namely trusting it more, because it is calculated on m samples near the input rather than a single input. Besides, it needs to point out that both two frameworks are on the basis of neighbor information at the current iteration instead of the last iteration. This is also a major difference between our methods and previous variance tuning.

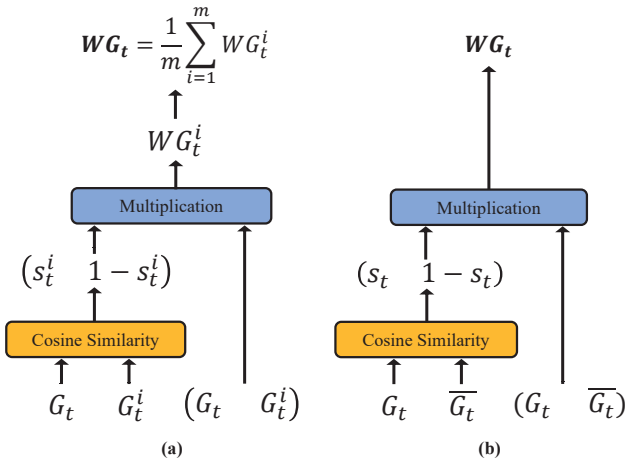

Figure 4. Illustration for two gradient relevance frameworks: (a) Individual gradient relevance framework; (b) Average gradient relevance framework.

3.3. Decay Indicator

The fluctuation phenomenon of adversarial perturbation's sign and the fixed step size will make the adversarial example oscillate around the optimum. We hence design a decay indicator M_{t+1} to decrease the step size when encountering frequent fluctuation on the adversarial perturbation's sign. On account of the fact that the adversarial perturbation's sign is dependent on the momentum accumulation's sign [7, 36, 37], we define the decay indicator M_{t+1} by

$$M_{t+1} = M_t \odot (M_{t+1}^e + \eta \cdot M_{t+1}^d), \tag{10}$$

where $\eta \in (0, 1)$ is the attenuation factor and the elements of M_0 are all set to $1/\eta$. M_{t+1}^e and M_{t+1}^d denote the unchanged and changed position of the adversarial perturbation severally in two adjacent iterations, their elements are defined as:

$$M_{t+1,j}^e = \begin{cases} 1, & if \ \ sign(g_t^j) = sign(g_{t+1}^j), \\ 0, & otherwise, \end{cases} \tag{11}$$

where $M^e_{t+1,j}$ is the j-th element on the M^e_{t+1} and g^j_{t+1} is the j-th element on the g_{t+1}.

$$M^d_{t+1,k} = \begin{cases} 1, & if \ \ sign(g^k_t) \neq sign(g^k_{t+1}), \\ 0, & otherwise, \end{cases} \tag{12}$$

where $M^d_{t+1,k}$ is the k-th element on the M^d_{t+1} and g^k_{t+1} is the k-th element on the g_{t+1}. Eq.(10) means if there is no

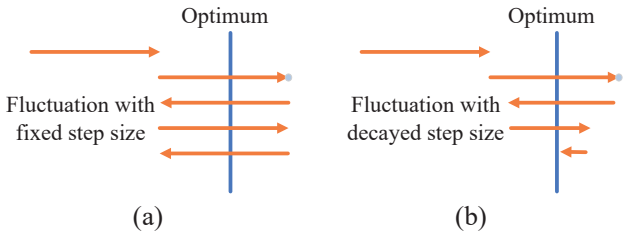

Figure 5. Two situations when encountering the fluctuation on adversarial perturbation's sign: (a) Without decay indicator; (b) With decay indicator

fluctuation, we just keep the fixed step size. However, if a pixel's sign fluctuates repeatedly, it indicates the value of this pixel on the adversarial example is probably near the optimum. We thereby decrease the step size with η at each oscillation to let it be closer to the optimum (see Figure 5). Then the update process of each iteration can be expressed as:

$$x^{adv}_{t+1} = Clip\{x^{adv}_t + \alpha \cdot M_{t+1} \odot sign(g_{t+1})\}, \tag{13}$$

where $Clip$ means limiting each pixel on the image within a given constraint and α is the fixed step size.

Combining MI-FGSM with the average gradient relevance framework and decay indicator, we have the final form of the Gradient Relevance Attack (GRA) in Algorithm 1. Note that when combining MI-FGSM and decay indicator with individual gradient relevance framework, we use I-GRA especially to distinguish two kinds of frameworks. Comparisons of GRA and I-GRA are reported in Table 1.

Algorithm 1 Gradient Relevance Attack

Input: A source model F_ψ with loss function L, a clean image x^{clean} and its corresponding true label y^{true}. The maximum magnitude of adversarial perturbation ε, the iteration number T and the decay factor of momentum accumulation μ, attenuation factor η, the upper bound factor β and the sample quantity m.

Output: The adversarial image x^{adv}_T.

1: **Initialize** $\alpha = \varepsilon/T, g_0 = 0, v_0 = 0, x^{adv}_0 = x$ and set all elements of M_0 to $1/\eta$

2: **for** $t = 0 \rightarrow T-1$ **do**

3: Calculate the current gradient $G_t(x)$ by Eq.(3) and the average gradient $\overline{G_t}(x)$ by Eq.(7)

4: Calculate the cosine similarity s_t and the global weighted gradient WG_t by Eq.(8)

5: Update momentum accumulation g_{t+1} with WG_t

$$g_{t+1} = \mu \cdot g_t + \frac{WG_t}{\|WG_t\|_1}$$

6: Update decay indicator M_{t+1} by Eq.(10)

7: Update x^{adv}_{t+1} with M_{t+1} and the sign of g_{t+1}

$$x^{adv}_{t+1} = Clip\{x^{adv}_t + \alpha \cdot M_{t+1} \odot sign(g_{t+1})\}$$

8: **end for**

4. Experiments

4.1. Experiment Setup

Dataset. We follow the tradition of using 1,000 clean images from ILSVRC 2012 validation set [27] to verify the availability of GRA, the same as previous works [18, 36, 37]. These clean images can be classified with almost 100% accuracy by the models involved in this paper.

Models. Four classical source models are selected to craft adversarial examples, containing Inception-v3 (Inc-v3) [30], Inception-v4 (Inc-v4), Inception-Resnet-v2 (IncRes-v2) [29] and Resnet-v2-101 (Res-101) [13]. Target models consist of the source models above and adversarially trained models [33] such as adv-Inception-v3 (Inc-v3$_{adv}$), ens3-adv-Inception-v3 (Inc-v3$_{ens3}$), ens4-Inception-v3 (Inc-v3$_{ens4}$) and ens-adv-Inception-ResNet-v2 (IncRes-v2$_{ens}$). Additionally, five defense methods including PD [26], NRP [24], JPEG [12], ComDefend [15] and FD [21] are also taken into consideration. Note that, only PD is combined with Resnet-v2-50 [13], and the rest four defense methods are combined with Inc-v3$_{ens3}$. Finally, we also conduct the proposed attack on two practical online models to show the threat in the real world.

Baseline Methods. Three of the latest gradient-based attacks VTMI-FGSM, VTNI-FGSM [36] and Admix [37] are taken into consideration in our experiments, which have exhibited higher attack success rates compared with previous attacks such as MI-FGSM [7] and NI-FGSM [18]. Additionally, we also involve the combined transformation (CT) [36] to verify the compatibility, where CT is the combination of DI [38], TI [8] and SI [18]. In the following context, we simply write VTMI-FGSM and VTMI-FGSM as VTMI and VTNI without ambiguity.

Parameter Setting. The attack setting is identical with previous works [7, 36, 37] where the iteration number T is set to 10, the upper bound on the perturbation magnitude ε is set to 16 and step size α is set to 1.6. We set the decay factor μ to 1.0 for MI, the transformation probability p to 0.5 for DI, the kernel size to 7×7 for TI, and the number of

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Average
Inc-v3	VTMI [36]	**100.0***	71.7	68.1	60.2	32.8	31.2	17.5	54.5
	VTNI [36]	**100.0***	76.5	74.9	66.0	35.0	32.8	18.8	57.7
	Admix [37]	**100.0***	82.6	80.9	75.2	39.0	39.2	19.2	62.3
	I-GRA (ours)	99.6*	86.1	84.6	78.6	57.9	56.8	38.4	71.7
	GRA (ours)	99.9*	**87.1**	**85.5**	**79.5**	**58.8**	**57.4**	**39.8**	**72.6**
Inc-v4	VTMI [36]	77.9	99.8*	71.2	62.2	38.2	38.7	23.2	58.7
	VTNI [36]	83.4	**99.9***	76.1	66.9	40.0	37.7	24.5	61.2
	Admix [37]	87.8	99.4*	83.2	78.0	55.9	50.4	33.7	69.8
	I-GRA (ours)	87.9	98.6*	85.6	79.2	65.2	62.1	50.9	75.6
	GRA (ours)	**89.8**	98.6*	**86.5**	**79.6**	**66.1**	**64.8**	**51.3**	**76.7**
IncRes-v2	VTMI [36]	77.9	72.1	97.9*	67.7	46.4	40.8	34.4	62.5
	VTNI [36]	80.8	76.9	98.5*	69.8	47.9	40.3	34.2	64.1
	Admix [37]	**89.9**	**87.5**	**99.1***	**81.9**	64.2	56.7	50.0	75.6
	I-GRA (ours)	86.0	84.2	95.0*	81.1	68.4	**64.0**	61.9	77.2
	GRA (ours)	86.0	83.1	96.3*	81.7	**69.0**	63.2	**62.6**	**77.4**
Res-101	VTMI [36]	75.1	68.9	70.5	99.2*	45.2	41.4	30.1	61.5
	VTNI [36]	79.8	74.6	73.2	**99.7***	46.1	42.5	32.1	64.0
	Admix [37]	85.4	80.8	79.6	**99.7***	51.0	45.3	30.9	67.5
	I-GRA (ours)	85.8	82.6	82.1	99.2*	71.5	67.7	**58.7**	78.2
	GRA (ours)	**87.1**	**83.0**	**83.8**	99.3*	**72.3**	**68.4**	57.8	**78.8**

Table 1. The attack success rates (%) on seven models by a single attack. The adversarial examples are generated on Inc-v3, Inc-v4, IncRes-v2, and Res-101 separately. * denotes the success rate of the white-box attack and the result in bold is the best.

scale copies c to 5 for SI. We let sample quantity m be 20 and the sample range factor β be 1.5 for VTMI and VTNI. For Admix, the number of copies m_1 is set to 5, the number of mixed images m_2 is set to 3, and the mixed ratio is set to 0.2. In our method, the sample quantity m is 20, the upper bound factor of sample range β is 3.5 and the attenuation factor η is 0.94. Furthermore, attacks combined with an ensemble attack strategy are conducted on a single NVIDIA Tesla V100 GPU, while other attacks are conducted on a single NVIDIA RTX 2080Ti GPU.

4.2. Attack with a Single Method

We craft adversarial examples by VTMI, VTNI, Admix, I-GRA, and GRA on four source models, and then attack seven target models. The attack results are illustrated in Table 1, where the attack success rate represents the misclassification rate of the target model. It is apparent that GRA outperforms other attacks on all the normally trained models except for a small gap on the IncRes-v2 and under the white-box setting. However, GRA exhibits overwhelming superiority on adversarially trained models and the average attack success rates are the highest among the four attack methods. For example, when the adversarial examples are generated on Res-101 and the target model is Inc-v3$_{ens4}$, GRA can yield an attack success rate of 68.4%, while the Admix can only achieve a 45.3% attack success rate.

For the comparison between I-GRA and GRA, both of them surpass the involved attack methods in most cases, while GRA is slightly better than I-GRA. We deem the reason is that I-GRA introduces much local relevance information, *i.e.*, m times cosine similarity computations each iteration, which degrades the generalization ability of the adversarial examples. Therefore, in the following experiments (Sec.4.3 and Sec.4.4), we only contain the GRA for simplicity because it is more threatening than I-GRA.

4.3. Attack with the Combined Transformation

To achieve a higher attack success rate, we need to examine the compatibility of the proposed GRA with the combined transformation (CT) [36] which is the combination of DI [38], TI [8] and SI [18]. Note that Admix has already included the SI, hence the Admix-CT only contains two extra augmentation transformations, *i.e.*, DI, and TI. As can be seen from Table 2, GRA-CT also has the highest attack success rates. Take the adversarial examples crafted on Inc-v3 for example, GRA-CT yields an average success rate of 90.4%, while the VTNI-CT merely obtains an average success rate of 83.7%, which persuasively certifies the good compatibility of GRA with other transformations.

4.4. Attack with Ensemble Strategy

Ensemble strategy can effectively boost the transferability of adversarial examples [7, 19]. We apply the ensemble strategy proposed in [7] to strengthen our GRA and attack nine defended models in Table 3. Note that we only include Admix and VTNI as our rivals, because the above results have shown that they are more challenging than VTMI. From Table 3, it is obvious that even with advanced

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Average
Inc-v3	VTMI-CT [36]	99.3*	88.6	86.7	82.9	78.6	76.2	64.7	82.4
	VTNI-CT [36]	99.5*	91.2	89.0	85.3	78.6	76.7	65.3	83.7
	Admix-CT [37]	**99.9***	89.0	87.0	83.1	72.2	71.1	52.4	79.2
	GRA-CT (ours)	99.7*	**92.7**	**92.3**	**91.2**	**89.0**	**88.1**	**79.8**	**90.4**
Inc-v4	VTMI-CT [36]	90.0	98.8*	86.6	81.9	78.3	76.6	68.3	82.9
	VTNI-CT [36]	92.1	99.2*	89.2	85.1	80.1	78.3	70.4	84.9
	Admix-CT [37]	90.4	99.0*	87.3	82.0	75.3	71.9	61.6	81.1
	GRA-CT (ours)	**92.5**	**99.3***	**90.1**	**87.9**	**86.1**	**86.0**	**79.9**	**88.8**
IncRes-v2	VTMI-CT [36]	88.9	87.0	97.0*	85.0	83.4	80.5	79.4	85.9
	VTNI-CT [36]	**92.9**	90.6	**99.0***	88.2	85.2	82.5	81.8	88.6
	Admix-CT [37]	90.1	87.6	97.7*	85.9	82.0	78.0	76.3	85.4
	GRA-CT (ours)	92.1	**91.4**	97.9*	**88.6**	**87.9**	**87.1**	**87.1**	**90.3**
Res-101	VTMI-CT [36]	86.9	84.2	86.4	98.6*	81.0	78.6	71.6	83.9
	VTNI-CT [36]	90.7	85.5	87.2	99.1*	82.6	79.7	73.3	85.4
	Admix-CT [37]	91.0	**87.7**	89.2	**99.9***	81.1	77.4	70.1	85.2
	GRA-CT (ours)	**91.4**	85.1	**89.4**	99.5*	**89.4**	**88.9**	**84.3**	**89.7**

Table 2. The attack success rates (%) on seven models by four gradient-based iterative attacks augmented with CT. The adversarial examples are generated on Inc-v3, Inc-v4, IncRes-v2, and Res-101 separately. * denotes the success rate of the white-box attack and the result in bold is the best.

Model	Attack	Inc-v3$_{adv}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	JPEG	ComDefend	NRP	FD	PD
Ens	VTNI [36]	70.7	70.3	66.8	52.2	78.3	80.8	16.0	74.6	79.2
	Admix [37]	72.5	77.8	73.2	59.1	84.1	83.1	24.1	77.5	84.7
	GRA (ours)	**88.4**	**88.4**	**87.2**	**82.1**	**92.3**	**90.9**	**29.2**	**88.1**	**100.0**

Table 3. The attack success rates (%) on nine defended models attacked by adversarial examples crafted on Inc-v3, Inc-v4, IncRes-v2, and Res-101 synchronously. The result in bold is the best.

defense methods, GRA can still achieve an average attack success rate of 83.0% and exceeds other attacks by more than 12.3%, which demonstrates the effectiveness of our attack.

Moreover, we select 150 adversarial examples crafted with ensemble strategy, and all of their corresponding clean images can be classified as the same category by two online models, *i.e.*, Tencent Cloud[1] and Baidu AI Cloud[2]. Then we evaluate their robustness and report the evaluation results in Table 4. Obviously, under the ensemble setting, GRA can exhibit a strong threat to the target models with an average success rate of 66.8%, which demonstrates an 11.1% and 11.4% improvement over VTNI and Admix, respectively. All these results reveal the vulnerability of current models in the real world.

Model	Attack	Tencent Cloud	Baidu AI Cloud
Ens	VTNI [36]	57.3	54.0
	Admix [37]	56.0	54.7
	GRA (ours)	**68.7**	**64.8**

Table 4. The attack success rates (%) on two online models attacked by adversarial examples crafted with ensemble strategy. The result in bold is the best.

4.5. Ablation Study

To verify the effectiveness of the components in GRA, we analyze three crucial hyper-parameters including the sample quantity m, the upper bound factor of sample range β, and the attenuation factor η.

Sample quantity m decides the amount of information extracted from the neighborhood of x_t^{adv}. As illustrated in Figure 6, the attack success rates increase rapidly on normally trained models with the increase of m, then tend to be stable after $m = 20$. Nevertheless, the attack success rates on adversarially trained models tend to increase even after $m = 50$. For a fair comparison, we set $m = 20$, the same as previous work [36].

Furthermore, taking into consideration both Table 1 and Figure 6, we can find that GRA is more effective than VTMI and VTNI. For example, when the sample quantity $m = 5$, GRA can fool Inc-v4 and IncRes-v2$_{ens}$ with attack success rates of 78.1% and 24.8% respectively. However, VTNI can only achieve 76.5% and 18.8% with $m = 20$, which forcefully demonstrates the superiority of our GRA.

Upper bound factor of the sample range β is a significant hyper-parameter, which determines the receptive scope of GRA. As displayed in Table 1 and Figure 7, when

[1]https://cloud.tencent.com/

[2]https://cloud.baidu.com/

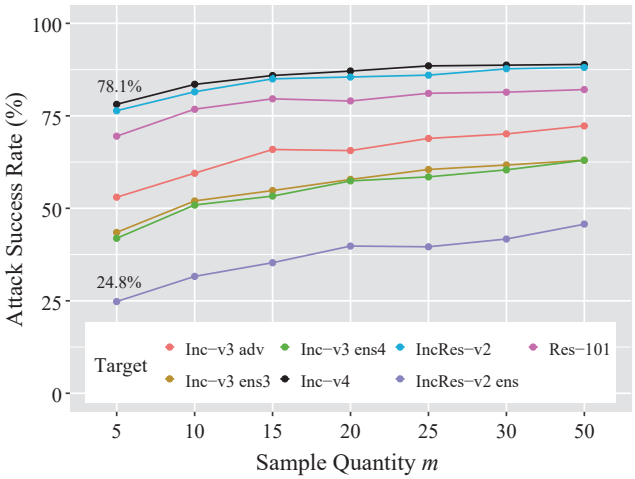

Figure 6. The attack success rates (%) of GRA with varying sample quantity and the adversarial examples are crafted on Inc-v3. Note that $\beta = 3.5$ and $\eta = 0.94$.

$\beta = 3.5$, GRA achieves the highest success rates on seven target models, which obtains a larger sample range than VT ($\beta = 1.5$) [36]. We argue that the gradient relevance framework gives the attack a broader receptive scope, which enables the crafted adversarial examples to absorb more unfamiliar neighbor information without reducing the attacking performance. And it is exactly the capability of receiving more unfamiliar neighbor information that makes the adversarial examples crafted by GRA more transferable.

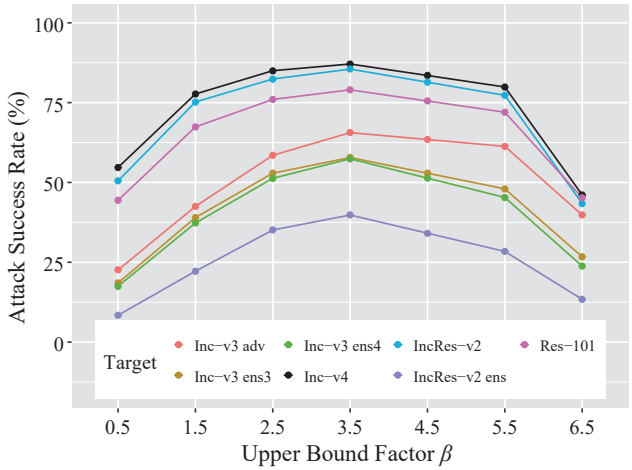

Figure 7. The attack success rates (%) of GRA with different upper bounds of the sample range factor β, where the adversarial examples are crafted on Inc-v3. Note that $m = 20$ and $\eta = 0.94$.

Attenuation factor η influences the decay speed of the step size when facing the fluctuation of the adversarial perturbation's sign. To search for a proper decay speed, we visualize the trend of the attack success rates with different attenuation factors η in Figure 8. To exhibit a clearer varying trend, we divide seven target models into two groups, *i.e.*, normally trained group and adversarially trained group, then compute their average attack success rates. It can be observed from Figure 8 that the average attack success rates are relatively high in both groups when attenuation factor η is near 0.94, so we let η be 0.94 in the proposed GRA.

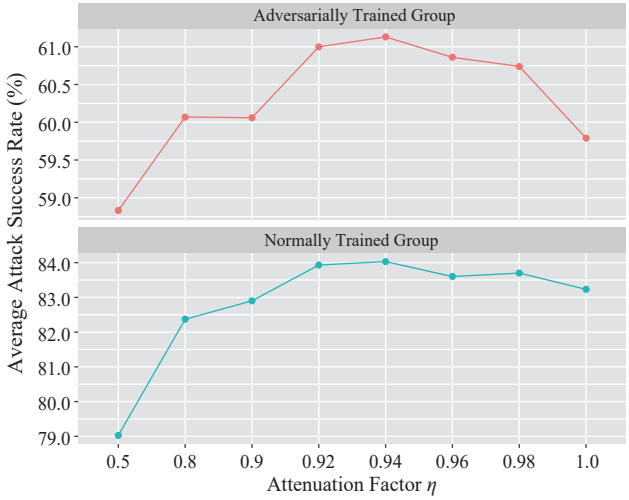

Figure 8. Under GRA, two groups' average attack success rates (%) with varying attenuation factor η and the adversarial examples are crafted on Inc-v3. Note that $m = 20$ and $\beta = 3.5$.

5. Conclusion and Discussion

In this paper, we proposed a novel Gradient Relevance Attack (GRA) and explored the fluctuation phenomenon on the plus-minus sign of the adversarial perturbations during the generation of adversarial examples. Concretely, we devised two gradient relevance frameworks to mine the potential neighbor information around the input. Considering both the efficiency of calculating cosine similarity and the attacking performance, we adopted the average gradient relevance framework in GRA. Moreover, we also devised the decay indicator to decrease the step size when encountering frequent fluctuation. Abundant experiment results on the subset of ILSVRC 2012 validation set convincingly verified the superiority of GRA. Finally, we also demonstrated the poor robustness of two practical online models with adversarial examples crafted by our method, which revealed a worrying fact that the models deployed in the real world were probably unreliable.

In the future, we think further improvements in gradient-based attacks may concentrate on two aspects. One is developing effective methods to utilize the extra information. There are many methods to enhance the input data, but few methods are put forward to make full use of them. The other is to devise a more proper way to fine-tune the step size, which can make the update direction more reasonable.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2014.

[3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016.

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

[5] Weilun Chen, Zhaoxiang Zhang, Xiaolin Hu, and Baoyuan Wu. Boosting decision-based black-box adversarial attacks with random sign flip. In *Proceedings of the European Conference on Computer Vision*, pages 276–293. Springer International Publishing, 2020.

[6] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3D object proposals using stereo imagery for accurate object class detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1259–1272, 2018.

[7] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018.

[8] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[9] Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. Patch-wise attack for fooling deep neural network. In *Proceedings of the European Conference on Computer Vision*, pages 307–322. Springer International Publishing, 2020.

[10] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1440–1448, 2015.

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2014.

[12] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2017.

[13] 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.

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2261–2269, 2017.

[15] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. ComDefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6085, 2019.

[16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations*, 2016.

[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2016.

[18] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks, 2019.

[19] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2016.

[20] Ye Liu, Yaya Cheng, Lianli Gao, Xianglong Liu, Qilong Zhang, and Jingkuan Song. Practical evaluation of adversarial robustness via adaptive auto attack. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15084–15093, 2022.

[21] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: DNN-oriented JPEG compression against adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 860–868, 2019.

[22] Cheng Luo, Qinliang Lin, Weicheng Xie, Bizhu Wu, Jinheng Xie, and Linlin Shen. Frequency-driven imperceptible adversarial attack on semantic similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15294–15303, 2022.

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2017.

[24] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 259–268, 2020.

[25] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[26] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8571–8580, 2018.

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg,

and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[28] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 618–626, 2017.

[29] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence*, 2016.

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[31] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2013.

[32] 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.

[33] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2017.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

[35] Jianzhou Wang, Kang Wang, Zhiwu Li, Haiyan Lu, He Jiang, and Qianyi Xing. A multitask integrated deep-learning probabilistic prediction for load forecasting. *IEEE Transactions on Power Systems*, pages 1–11, 2023.

[36] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1924–1933, 2021.

[37] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16138–16147, 2021.

[38] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2725–2734, 2019.

[39] Junhua Zou, Yexin Duan, Boyu Li, Wu Zhang, Yu Pan, and Zhisong Pan. Making adversarial examples more transferable and indistinguishable. In *AAAI Conference on Artificial Intelligence*, 2022.