Article

Probability Distribution-Guided Adversarial Sample Attacks

Hongying Li [†] , Miaomiao Yu [†] , Xiaofei Li [†] , Jun Zhang [*] , Shuohao Li , Jun Lei , Hairong Huang

Posted Date: 12 May 2023

doi: 10.20944/preprints202305.0872.v1

Keywords: Probability Distribution; Adversarial Sample; Transferability; Interpretability

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2023 doi:10.20944/preprints202305.0872.v1

Article

Probability Distribution-Guided Adversarial Sample Attacks

Hongying Li [1,†], Miaomiao Yu [1,†], Xiaofei Li [1,†], Jun Zhang [1,*], Shuohao Li [1], Jun Lei [1], and Hairong Huang [2]

[1] Laboratory for Big Data and Decision, National University of Defense Technology, Changsha, Hunan, 410000, China
[2] Teacher Training College, Zhongxian, Chongqing, 404300, China
* Correspondence: zhangjun1975@nudt.edu.cn
† These authors contributed equally to this work.

Abstract: In recent years, with the rapid development of technology, artificial intelligence(AI) security issues represented by adversarial sample attack have aroused widespread concern in society. Adversarial samples are often generated by surrogate models and then transfer to attack the target model, and most AI models in real-world scenarios belong to black boxes, thus transferability becomes a key factor to measure the quality of adversarial samples. The traditional method relies on the decision boundary of the classifier and takes the boundary crossing as the only judgment metric without considering the probability distribution of the sample itself, which results in an irregular way of adding perturbations to the adversarial sample, an unclear path of generation, and a lack of transferability and interpretability. In the probabilistic generative model, after learning the probability distribution of the samples, a random term can be added to the sampling to gradually transform the noise into a new independent and identically distributed sample. Inspired by this idea, we believe that by removing the random term, the adversarial sample generation process can be regarded as static sampling of the probabilistic generative model, which guides the adversarial samples out of the original probability distribution and into the target probability distribution, and helps to improve transferability and interpretability. Therefore, we propose a Score Matching-Based Attack(SMBA) method to perform the adversarial sample attacks by manipulating the probability distribution of the samples, which can show good transferability in the face of different datasets and models, and give reasonable explanations from the perspective of mathematical theory and feature space. In conclusion, our research establishes a bridge between probabilistic generative models and adversarial samples, provides a new entry angle for the study of adversarial samples, and brings new thinking to AI security.

Keywords: Probability Distribution; Adversarial Sample; Transferability; Interpretability

1. Introduction

With the advent of the era of big data and the development of deep learning theory, AI technology is like a sword of Damocles, which brings social progress but also brings serious security risks [1]. For example, the adversarial sample attacks [2], which adds subtle perturbations to the source sample to generate new sample objects. Detection, classification, and other AI algorithms are very sensitive to these subtle perturbations and thus yield false results. In driverless scenarios in the civilian field, attackers can transform road signs into corresponding adversarial samples, causing the driverless system to misjudge the road signs and thus causing traffic accidents [3]. In the UAV strike scenario in the military field, adversarial sample generation can be used to enforce camouflage stealth on the target for protection, which blinds the UAV from completing the attacks [4]. Thus, it can be seen that research on adversarial samples is an important way to improve the AI security.

Adversarial sample attack can be divided into targeted and non-targeted attacks [1]. Taking the targeted attacks as an example, the traditional adversarial sample generation process is $x' =$

$x - \nabla_x L\left(y^p, y^t\right) = x + \nabla_x \log p_\theta\left(y^t \mid x\right)$, where x is sample, x' is adversarial sample, $L\left(y^p, y^t\right)$ is the loss function of predicted labels y^p and target labels y^t, and $p_\theta\left(y^t \mid x\right)$ is the probability that the classifier predicts the sample x as target label y^t. The above method is guided by the decision boundary of the classifier, so that the adversarial sample moves in the direction of the gradient that reduces the classification loss or increases the probability of the target class label, which takes the boundary crossing as the only judgment metric, has no consideration for the probability distribution of the sample itself, and results in an irregular way of adding perturbations to the adversarial sample, an unclear path of generation, and a lack of interpretability. For security reasons, most deep learning models in realistic scenarios are black boxes, and the different structures of classifiers lead to more or less differences in decision boundaries, which also seriously affect transferability of the adversarial samples and will fail when attacking realistic black box models. As shown in the blue box in the left part of Figure 1, The adversarial sample generated by the classifier decision boundary-guided approach cannot break the decision boundary of classifier B even if it breaks the decision boundary of classifier A, thus the transferable attack fails.

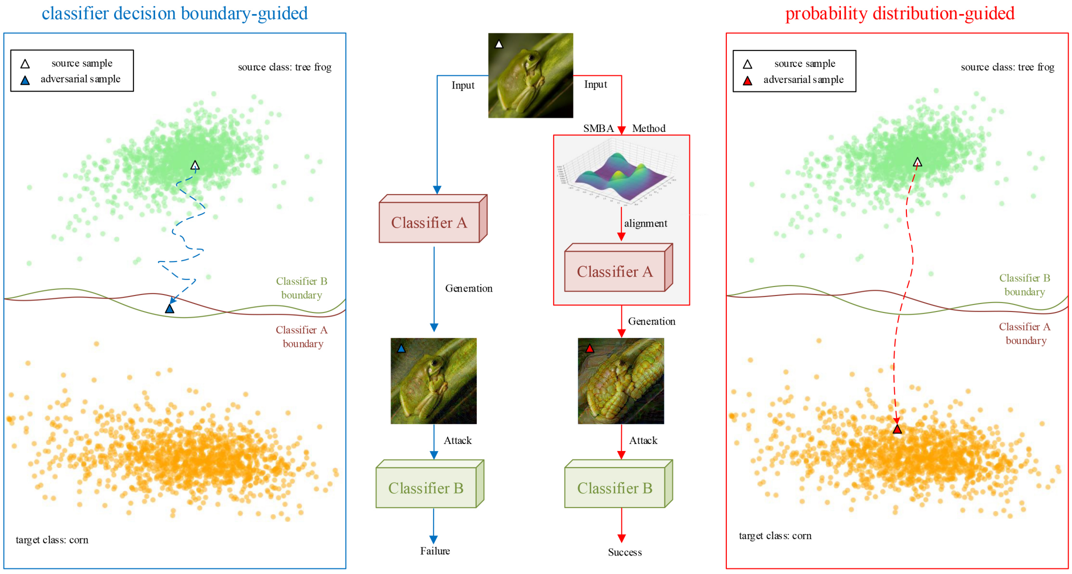


Figure 1. Comparison of the traditional method(blue mark) with our proposed method(red mark). The blue and red boxes indicate the effect of the adversarial samples generated by the classifier decision boundary-guided approach and the probability distribution-guided approach to attack different structural classifiers, respectively. The classifier decision boundary-guided method has poor transferability and fails to attack classifiers with different structures, while the probability distribution-guided method can move the adversarial sample from the source class to the probability distribution space of the target class, breaking the structural limitation of the classifier and achieving high transferability.

From the perspective of probability, any type of sample has its own probability distribution, which reflects the unique semantic characteristics. Classifiers with different structures will generally give similar classification results for a batch of independently and identically distributed samples, in other words, the probability distribution of samples plays a more critical role in the classification process than the structure of the classifier. If we manipulate the probability distribution of samples to guide the generation and attack of adversarial samples, we can get rid of the limitation of classifier structure to generate adversarial samples with high aggressiveness and transferability, and explain the process of generation from the perspective of mathematical theory. As shown in the red box in the right part of Figure 1, the adversarial sample generated by the probability distribution-based approach not only breaks through the decision boundary of classifiers A and B, but also reaches the probability distribution space of the target class, and the generation path is clear, thus the transferable attack is successful.

How to obtain the probability distribution of samples? Let's solve this problem from the perspective of a probabilistic generative model. The generation of adversarial samples is essentially a special probabilistic generative model, except that the data generation process is less random and more directional. For the probabilistic generative model, if $p_\theta(x)$ learned by the neural network can estimate the true probability density $p_{\text{data}}(x)$ of the sample, according to the Stochastic Gradient Langevin Dynamics(SGLD) sampling method [5], we iteratively move the initial random noise in the direction of the logarithmic gradient of the sample probability density, then a new independent and identically distributed sample x_k can be sampled according to Eq.(1):

$$x_k = x_{k-1} + \frac{\alpha}{2} \cdot \nabla_{x_{k-1}} \log p_{\text{data}}\ (x_{k-1}) + \sqrt{\alpha} \cdot \varepsilon = x_{k-1} + \frac{\alpha}{2} \cdot \nabla_{x_{k-1}} \log p_\theta\ (x_{k-1}) + \sqrt{\alpha} \cdot \varepsilon \tag{1}$$

where ε is the random noise used to promote diversity in the generation process, k is the number of iterations, and α is the sampling coefficient. Inspired by the above idea, if the randomness due to noise is reduced by removing the tail term $\sqrt{\alpha} \cdot \varepsilon$, the adversarial sample generation can be regarded as static SGLD sampling according to Eq.(2):

$$x_k = x_{k-1} + \alpha \cdot \nabla_{x_{k-1}} \log p_\theta\left(y^t \mid x_{k-1}\right) = x_{k-1} + \alpha \cdot \nabla_{k-1} \log p_{\text{data}}\ \left(x_{k-1} \mid y^t\right) \tag{2}$$

At this point, it is only necessary to use the classifier to approximate the logarithmic gradient $\nabla \log p_{\text{data}}\ \left(x \mid y^t\right)$ of the sample true conditional probability density, then the adversarial sample can be moved out of the original probability distribution space and approached toward the probability distribution space of the target class, which naturally can obtain a higher transferability and a more reasonable explanation.

Therefore, in order to solve the problems of insufficient transferability and poor interpretability of traditional adversarial sample attack methods, we propose a Score Matching-Based Attack(SMBA) method to guide the adversarial sample generation and attack by manipulating the probability distribution of samples. The main contributions of this paper are shown as follows:

- The limitations of traditional adversarial sample generation methods based on the decision boundary guidance of classifiers are broken through, and the generation mechanism of adversarial samples can be interpreted from the perspective of sample probability distribution.
- The transformation of the classification model into the energy-based model is achieved, making it possible to estimate the probability density of samples using the classification model.
- The classifier learns the probability distribution of the samples by aligning the gradient of the classifier with the logarithmic gradient of the probability density of the samples, which can guides the adversarial sample generation directionally and improves the transferability and interpretability in the face of different structural models.

2. Related Works

In this section, the literature related to adversarial sample attack is reviewed first, then the research related to probability density estimation in probabilistic generative models is introduced, and finally the advantages and application scenarios of the Score Matching(SM) method are presented, explaining the reasons why it can be used to adversarial sample attacks.

2.1. Adversarial Sample Attack Methods

Adversarial sample attack methods can be divided into targeted and non-targeted attacks according to the presence or absence of a specific target to be attacked, white-box and black-box attacks according to whether the attacker knows the target model, and gradient-based attacks, optimization-based attacks, transfer-based attacks, decision-based attacks, and other attacks according to the attack methods [1].

Gradient-based attacks: Goodfellow et al. [6] are the first to propose the gradient-based attack method FGSM, which adds perturbations along the reverse direction of the gradient to make the loss function change and eventually lead to model misclassification. However, the attack success rate is low due to the single-step attack with large computational perturbation error. To address this problem, Kurakin et al. [7] propose the iteration-based method I-FGSM, which subdivides the single-step perturbation into multiple steps and restricts the image pixels to the effective region by clipping, thus improving the attack success rate. I-FGSM tends to overfit to the local extremes, which affects the transferability of the adversarial samples, thus Dong et al. [8] propose MI-FGSM, which introduces the momentum idea to stabilize the gradient update direction while crossing the local extrema. PGD [9] is also an improvement based on the above method, which greatly improves the attack effect by adding a layer of random initialization processing and increasing the number of iterations. The alternative method such as DI^2-FGSM [10] is designed on the preprocessing of the image, which enhances the transferability and stability of the method. Besides FGSM and its variants, Papernot et al. [11], inspired by the saliency map concept [12], propose the JSMA method, which uses gradient information to calculate the pixel positions that have the greatest impact on the classification results and adds perturbations to them.

Optimization-based attacks: The essence of the adversarial sample attack algorithm is to find relatively small perturbations to generate an effective adversarial sample to implement the attack, so the process of adversarial sample generation can be defined as an optimization problem to be solved. The Box-constrained L-BFGS method propose by Szegedy et al. [2] is a prototype of an optimization-based attack method, which uses quasi Newton method in numerical optimization to solve. C& W [13] is the most classical optimization method, which defines different objective functions and can increase the space size of the optimal solution by changing the variables in the objective function, thus significantly improving the success rate of the attack. Unlike the above attack methods, Su et al. [14] propose the One-pixel method that requires only one pixel point to be modified for a successful attack, which uses differential evolutionary optimization algorithm to generate the perturbation by determining the location of the single pixel point to be modified.

Transfer-based attacks: An attacker uses a white-box attack approach to perform an attack on the surrogate model of the target model to generate a transferable adversarial sample and successfully attack the target model. Querying the target model to obtain a similar training dataset to generate a surrogate model is the main idea of obtaining a surrogate model [15]. Li et al. [16] select the most informative sample for querying through an active learning strategy, which further reduces the query cost while improving the model training quality. Inspired by data enhancement strategy, Xie et al. [10] quickly and effectively expand the dataset by transformations such as cropping and rotating the training dataset, thus the overfitting phenomenon of surrogate models has been solved. Wu et al. [17] introduce the concept of model attention and use the attention-weighted combination of feature mappings as a regularization term to further address the overfitting phenomenon of surrogate models. Li et al. [18] find that multiple integrated surrogate models do not need to have large variability, thus they use existing surrogate models to generate several different virtual models for integration, which significantly enhances the transferability of adversarial samples and reduces the training cost of alternative models.

Decision-based attacks: In transfer-based attack methods, querying the target model is an essential step, and the attack will fail when the query to the target model is restricted. In contrast, the decision-based black-box attack methods successfully get rid of the reliance on querying the target model by random wandering, which are more in line with the actual attack scenario. In simple terms, the attacker first obtains the initial adversarial sample with a large perturbation value and uses it as a basis to search for smaller perturbation values near the model decision boundary to obtain the final adversarial sample. Hence, how to determine the search direction for smaller perturbation values and how to improve its search efficiency are the two aspects that need to be focused on. Dong et al. [19] use CMA-ES [20] to model the search direction on the decision boundary with local geometry, thus reducing the search dimension and improving the search efficiency. Brunner et al. [21] propose a biased decision boundary search framework to find better search directions by restricting the decision boundary of search to perturbations with higher attack success rate. Shi et al. [22] propose CAB approach by exploring the relationship between initial perturbations and search-improved perturbations, which is able to obtain smaller values of adversarial perturbations. Rahmati et al. [23] observe that the decision boundaries of deep neural networks usually have a small mean curvature near the data samples, and accordingly propose GeoDA with high query efficiency.

Other attacks: Attack methods based on generative adversarial network(GAN) [24] can generate adversarial samples without knowing information about the target model. GAN uses a generator to generating adversarial samples, and then feeds the adversarial samples to a discriminator to ensure that the differences between the adversarial samples and the source images are small enough. An external target model determines the difference between the predicted label and the true label of the adversarial sample. The attack is successful if the final adversarial sample obtained is true and natural and misclassifies the target model. The above method is named as AdvGAN [25]. Subsequently, Jandial et al. [26] change the input of the generator from the source image to its potential feature vector, which reduces the GAN training time and significantly improves the attack success rate. Based on the idea of hyperplane classification, Moosavi et al. [27] proposes the DeepFool method by calculating the shortest distance between the source sample and the classification boundary of the target model. Later, Moosavi et al. [28] go on to propose the UAP method, which generates adversarial perturbations with strong generalization capability by calculating the shortest distance between the source sample and multiple classification decision boundaries of the target model. Similar to the idea of JSMA method to find salient graphs, Zhou et al. [29] use CAM to filter out important features of images to generate adversarial samples by content perception means to achieve low-cost and high transferability of adversarial attack.

2.2. Probability Density Estimation Methods

Probability density distribution estimation is originally applied to probabilistic generative models and can be used to model different raw data. We can estimate the true distribution by observing finite samples and sample a new independent and identically distributed sample. Its working principle is mainly based on maximum likelihood estimation. On the one hand, for the estimation of explicit probability distributions with parameters θ, the model gives a likelihood of $L = \prod_{i=1}^{m} p_{\text{model}}\left(x^{(i)}, \theta\right)$ to the m training samples, and the maximum likelihood principle is to choose the parameter $\theta^* = \arg\max_{\theta} \prod_{i=1}^{m} p_{\text{model}}\left(x^{(i)}, \theta\right)$ that maximizes that probability; on the other hand, for the estimation of implicit probability distributions, the maximum likelihood can be approximated as the solution of the parameter $\theta^* = \arg\max_{\theta} D_{KL}\left(p_{\text{data}}(x) \| p_{\text{model}}(x; \theta)\right)$ that minimizes the Kullback-Leibler Divergence[30] between the model distribution and the data distribution. Therefore, likelihood-based generative models can be divided into implicit models and explicit models [31].

Implicit models: Typical representatives are GAN and GSN [32], the core purpose of which is to make the distribution $p_g(x)$ of data generated by the model approximate the true distribution $p_{\text{data}}(x)$ of the original data. GAN does not explicitly model the probability density function $p_g(x)$ and cannot be solved by the maximum likelihood method, however, it can directly use the generator to sample from the noise and output samples, and force the minimum distance between $p_g(x)$ and $p_{\text{data}}(x)$ to be learned with the help of a discriminator. The GSN differs from GAN in that it needs to use Markov chains to sample after reaching a smooth distribution, and the huge computational effort makes it difficult to extend to high-dimensional spatial data.

Explicit models: For models with explicitly defined probability density distribution $p_{\text{model}}(x;\theta)$, the process of maximum likelihood is relatively straightforward by substituting the probability density distribution into the expression of the likelihood and updating the model in the direction of the increasing gradient, but the challenge is to define the model in such a way that it can express the complexity of the data and facilitate the calculation. Tractable explicit models define a probability density distribution that is easy to compute, and the main model is FVBN [33], which uses the chain rule of probability and transforms it into the form of the joint product of conditional probabilities, but the disadvantage is that the generation of element values depends on the previous element values, which is inefficient. MADE [34] and PixelRNN [35] all belong to this class of models. Approximate explicit models avoid the above limitation of needing to set a probability density function that is easy to solve, and use some approximate methods to solve the maximum likelihood instead. VAE [36] transforms solving the maximum likelihood into solving the extreme value problem with ELBO(Evidence Lower Bound) by variational approximate inference. MCMC [37] is a method that uses Markov chains to simplify the computational process of Monte Carlo random sampling to obtain approximate results. EBM [38] represents the maximum likelihood by constructing an energy function to estimate the minimum energy of the samples.

2.3. Score Matching Methods

Considering a dataset of Set $= \{x_1, x_2, \cdots, x_n\}$ from the true distribution $p_{\text{data}}(x)$ of source data, probabilistic generative model based on the maximum likelihood estimation must find $p_\theta(x)$ to approximate $p_{\text{data}}(x)$. take EBM as an example, the probability density function is modeled as $p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)}$. where $E_\theta(x)$ denotes the energy of the sample x, the lower the energy the higher the probability, which is a non-normalized probability and can be trained by a deep neural network. Z_θ is a normalization constant depending on θ so as to guarantee $\int p_\theta(x)dx = 1$. However, since Z_θ involves all the data in the probability distribution and is difficult to solve, in order to make maximum likelihood training feasible, the likelihood-based generative model must restrict its model structure or approximate Z_θ, which is more computationally expensive. The good thing is that this problem is cleverly circumvented by the score function $s(x) = \nabla_x \log p(x)$ [39], which is the logarithmic gradient of the probability density and points to the gradient field in the direction of the fastest growth of the probability density function. According to $\nabla_x \log p_\theta(x) = -\nabla_x E_\theta(x) - \nabla_x \log Z_\theta = -\nabla_x E_\theta(x)$, the score function eliminates Z_θ by taking the derivative, which makes the solution easier.

Hyvarinen et al. [39] first propose the Score Matching(SM) method to solve the unstandardized statistical model by estimating the difference between $\nabla_x \log p_{\text{data}}(x)$ and $s_\theta(x) = \nabla_x \log p_\theta(x)$. Since the solution of the SM method involves the calculation of $\nabla_x s_\theta(x)$, i.e., the Hessian matrix about $\log p_\theta(x)$, which involves multiple backpropagation and is computationally intensive, Song et al. [40] propose the Sliced Score Matching(SSM) method on this basis, which projects the high-dimensional vector field of $\log p_\theta(x)$ onto the low-dimensional randomly sliced vector v of a simple distribution (e.g., multivariate standard Gaussian distribution, uniform distribution, etc.) for solution, and the vector problem is scalarized, requiring only one backpropagation, which greatly reduces the computational effort.

Since the SM method estimates $\nabla_x \log p_{\text{data}}(x)$ by solving for $s_\theta(x) = \nabla_x \log p_\theta(x)$, and the adversarial sample generation process precisely requires gradient information, the SM method can be applied. Compared with traditional adversarial sample attack methods that rely only on the decision boundary guidance of classifiers, our method considers the probability distribution of samples, breaks through the limitations of different structural classifiers, effectively improves the transferability, and gives a reasonable explanation from the mathematical theory and visualization perspectives.

3. Methodology

In this paper, we focus on exploring the probability distribution of samples, and estimate the gradient of the true probability density of samples by the SM method, so as to guide the source samples to move towards the probability distribution space of the target class to generate adversarial samples with higher transferability. The overview of our proposed SMBA framework is shown in Figure 2.

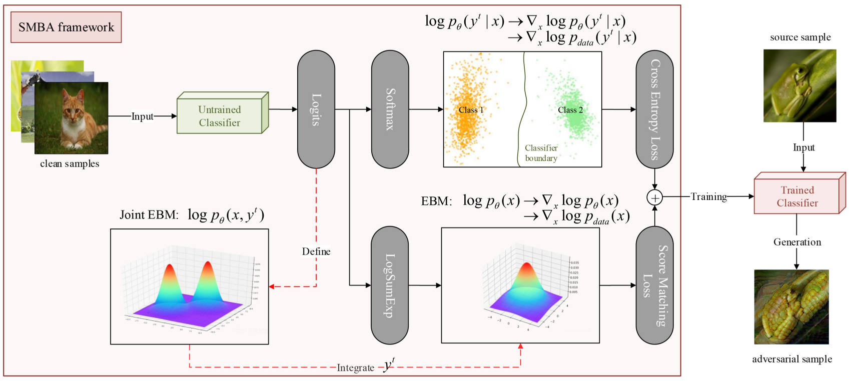

SMBA framework
$\log p_\theta(y^t \mid x) \rightarrow \nabla_x \log p_\theta(y^t \mid x)$
$\rightarrow \nabla_x \log p_{data}(y^t \mid x)$
source sample
clean samples
Input
Untrained Classifier
Logits
Softmax
Class 1
Class 2
Classifier boundary
Cross Entropy Loss
Input
Training
Trained Classifier
Generation
adversarial sample
Joint EBM: $\log p_\theta(x, y^t)$
Define
LogSumExp
EBM: $\log p_\theta(x) \rightarrow \nabla_x \log p_\theta(x)$
$\rightarrow \nabla_x \log p_{data}(x)$
Score Matching Loss
Integrate y^t

Figure 2. Overview of our proposed SMBA framework. After inputting clean samples, on the one hand, the Cross-Entropy Loss of the classifier is used to obtain the approximate value $\nabla_x \log p_\theta\left(y^t \mid x\right)$ of $\nabla_x \log p_{\text{data}}\left(y^t \mid x\right)$; on the other hand, The Joint EBM model is defined by the logits layer of the classifier, which is transformed into an EBM model after integrating y^t, and then the logarithmic gradient $\nabla_x \log p_{\text{data}}(x)$ of the sample probability density over the EBM is estimated using the SM method to obtain the approximation $\nabla_x \log p_\theta(x)$. Last, a gradient-aligned classifier is obtained by jointly training the Cross-Entropy Loss(CE Loss) with the Score matching Loss(SM Loss), and the adversarial samples can be generated.

3.1. Estimation Of The Logarithmic Gradient Of The True Conditional Probability Density

Given the input sample x and the corresponding label y, the classifier can be represented as $y = f(x, \theta)$. Let the total number of classifications be n and $f_\theta(x)[k]$ denote the Kth output of the logits layer of the classifier. The conditional probability density formula for predicting the label as y is:

$$p_\theta(y \mid x) = \frac{\exp\left(f_\theta(x)[y]\right)}{\sum_{k=1}^{n} \exp\left(f_\theta(x)[k]\right)} \tag{3}$$

Targeted attacks are committed to move in the direction of the gradient that reduces the classification loss or increases the probability of the target class label:

$$x' = x - \nabla_x L\left(f_\theta(x), y^t\right) = x + \nabla_x \log p_\theta\left(y^t \mid x\right) \tag{4}$$

The opposite is true for non-targeted attack:

$$x' = x + \nabla_x L\left(f_\theta(x), y\right) = x - \nabla_x \log p_\theta(y \mid x) \tag{5}$$

The subsequent derivation of the formula is based on the targeted attack, and the non-targeted attack is obtained in the same way. According to the idea that adversarial sample generation can be regarded as static SGLD sampling in probabilistic generative models, if the logarithmic gradient $\nabla \log p_{\text{data}}\left(x \mid y^t\right)$ of the sample true conditional probability density can be approximated using the classifier, then the ideal adversarial sample can be obtained according to Eq.(2).

Next derive the estimation method for $\nabla \log p_{\text{data}}\left(x \mid y^t\right)$. According to Bayes Theorem:

$$p_{\text{data}}\left(x \mid y^t\right) = \frac{p_{\text{data}}\left(x, y^t\right)}{p_{\text{data}}\left(y^t\right)} = \frac{p_{\text{data}}\left(y^t \mid x\right) \cdot p_{\text{data}}(x)}{p_{\text{data}}\left(y^t\right)} \tag{6}$$

After taking the logarithm of both sides, we have:

$$\log p_{\text{data}}\left(x \mid y^t\right) = \log p_{\text{data}}\left(y^t \mid x\right) + \log p_{\text{data}}(x) - \log p_{\text{data}}\left(y^t\right) \tag{7}$$

By eliminating the tail term by derivation we get:

$$\nabla_x \log p_{\text{data}}\left(x \mid y^t\right)=\nabla_x \log p_{\text{data}}\left(y^t \mid x\right)+\nabla_x \log p_{\text{data}}(x) \tag{8}$$

The first term on the right-hand side of which can be approximated by the regular gradient term $\nabla_x \log p_\theta\left(y^t \mid x\right)$ of the adversarial sample generated by the classifier. The second term is the logarithmic gradient of the input sample probability density. From the generative model point of view, we need to reconstruct a generative network (generally EBM) to solve the second term by the SM method.

The SM method requires the closer the score function $s_\theta(x)$ learned by the generative network to the logarithmic gradient of the sample probability density $\nabla_x \log p_{\text{data}}(x)$, and the mean squared error loss is used as a measure:

$$SM\_Loss=\frac{1}{2} E_{p_{\text{data}}(x)}\left[\left\|s_\theta(x)-\nabla_x \log p_{\text{data}}(x)\right\|_2^2\right] \tag{9}$$

Due to the difficulty in solving $p_{\text{data}}(x)$, after a theoretical derivation [39], Eq.(9) can be simplified to:

$$SM\_Loss=E_{p_{\text{dada}}(x)}\left[\operatorname{tr}\left(\nabla_x s_\theta(x)\right)+\frac{1}{2}\left\|s_\theta(x)\right\|_2^2\right] \tag{10}$$

where the unknown $p_{\text{data}}(x)$ is eliminated and the solution process only requires the score function $S_\theta(X)$ learned by the generative network.

Considering that $\operatorname{tr}\left(\nabla_x s_\theta(x)\right)$ involves a complex calculation of the Hessian matrix, it can be further simplified by the SSM method as follows:

$$SSM\_Loss=E_{p_v} E_{p_{\text{data}}(x)}\left[v^T \nabla_x\left(v^T \cdot s_\theta(x)\right)+\frac{1}{2}\left\|s_\theta(x)\right\|_2^2\right] \tag{11}$$

The simplified loss function simply needs to be supplied with a randomly sliced vector v of a simple distribution to approximate $\nabla_x \log p_{\text{data}}(x)$ with $S_\theta(x)$.

The problem now is that the classification model and EBM have different structures, and the SM and classification processes are independent of each other, so how to combine the classification model and EBM to construct a unified loss function to establish the constraint relationship is what we need to solve.

3.2. Transformation Of Classification Model To EBM

Inspired by the literature [41], we can transform the classification model into the EBM. The probability density function of the EBM is:

$$p_\theta(x)=\frac{\exp\left(-E_\theta(x)\right)}{Z(\theta)} \tag{12}$$

Consider the conditional probability density function for the universal form of the n classification problem as:

$$p_\theta(y \mid x)=\frac{\exp\left(f_\theta(x)[y]\right)}{\sum_{k=1}^n \exp\left(f_\theta(x)[k]\right)} \tag{13}$$

Now use the values $f_\theta(x)[y]$ of the logits layer of the classification model to define a Joint EBM :

$$p_\theta(x, y)=\frac{\exp\left(f_\theta(x)[y]\right)}{Z(\theta)} \tag{14}$$

According to the definition of EBM, it can be seen that $E_\theta(x, y)=-f_\theta(x)[y]$. By integrating over y, we can obtain:

$$p_\theta(x) = \sum_y p_\theta(x, y) = \frac{\sum_y \exp\left(f_\theta(x)[y]\right)}{Z(\theta)} \tag{15}$$

Taking the logarithm of both sides and deriving, we get:

$$\nabla_x \log p_\theta(x) = \nabla_x \left(-E_\theta(x)\right) = \nabla_x \log \sum_y p_\theta(x, y) = \nabla_x \log \sum_y \exp\left(f_\theta(x)[y]\right) \tag{16}$$

At this point, as long as the values $f_\theta(x)[y]$ of the logits layer of the classification model are obtained, the classification model can be directly used to replace the EBM for SM estimation of $\nabla_x \log p_{\text{data}}(x)$.

3.3. Generation Of Adversarial Samples On Gradient-Aligned Classifiers

The classifier can classify the source samples with high accuracy after CE loss training, and the gradient direction of the classifier can be aligned with the gradient direction of the true probability density logarithm of the source sample after SM estimation, so as to guide the adversarial sample generation process in a more directional way. By effectively combining the loss functions of both, the impact of parameter adjustment on the classification accuracy during gradient alignment can be reduced, and the final joint training objective is:

$$\theta^* = \arg\min_\theta \left[\text{ Loss } = CE\_Loss + \lambda \cdot SSM\_Loss\right] \tag{17}$$

where λ is the constraint coefficient. The gradient-aligned classifier can now be used to generate adversarial samples with high transferability according to Eq.(2) and (8).

4. Experiments And Results

In this section, the experimental setup is first described, followed by a comprehensive comparative analysis of the proposed method with existing adversarial sample attack methods, and finally the effectiveness of the model is demonstrated from a visualization perspective.

4.1. Experimental Settings

The dataset we take is ImageNet-1K [42], whose training set has a total of 1.3 million images in 1000 classes, and after preprocessing such as random cropping and flipping, the input size is transformed from uniformly $3 \times 224 \times 224$ pixels to the $[0, 1]$ interval with the batch size set to 16. In the joint training process, we choose ResNet-18 [43] as the surrogate model, which is trained according to Eq.(17), and optimized by SGD [44] optimizer with a learning rate of 0.001, and the constraint coefficient $\lambda = 5$. In the attack scenario, we select five advanced adversarial attack methods (PGD, MI-FGSM, DI2-FGSM, C&W, SGM [45]) to compare with our method. Five normal models with different structures (VGG-19 [46], ResNet-50 [43], DenseNet-121 [47], Inception-V3 [48], ViT-B/16 [49]) and three robust models (Adversarial Training [50], SIN [51], Augmix [52]) processed by adversarial training and data enhancement methods are selected as target models. The attacks are divided into non-targeted and targeted attacks. The maximum adversarial perturbation allowed by default is $\varepsilon = 16/255$, step size is $\eta = 2/155$, and iteration number is $N = 10$. Random 10,000 images correctly classified by the surrogate model on the validation set of ImageNet-1K are selected to generate adversarial samples for evaluation, and the evaluation metric is the attack success rate (1 - correct classification rate). To demonstrate the generality of our approach, experiments are also performed on the CIFAR-100 dataset [53]. Finally, in the visualization part, the perturbation strength is appropriately increased to generate more observable adversarial samples.In order to facilitate the description of the generation path and the attack process, and give a reasonable explanation, the Principal Component

Analysis(PCA) [54] method is adopted to reduce the embedding of the adversarial samples from high-dimensional features to two-dimensional feature space for observation.

4.2. Metrics Comparison

In this section, we firstly compare the performance of different methods for targeted and non-targeted attacks on different target models, Subsequently compare the performance of the different methods on different datasets.

4.2.1. Attack Target Model Experiments

As shown in Table 1 and Table 2, in attacking the normal model, the transferability of traditional methods is relatively low when either non-targeted or targeted attacks are performed. Especially from CNN network structures (VGG-19, ResNet-50, DenseNet-121, Inception-V3) to Vision Transformer network structure (ViT-B/16), in the case of PGD non-targeted attacks, for example, the transferability drops from a maximum of 45.05% to 3.25%. Fortunately, our SBMA method achieves the highest attack success rate compared to other methods and exhibits good transferability.

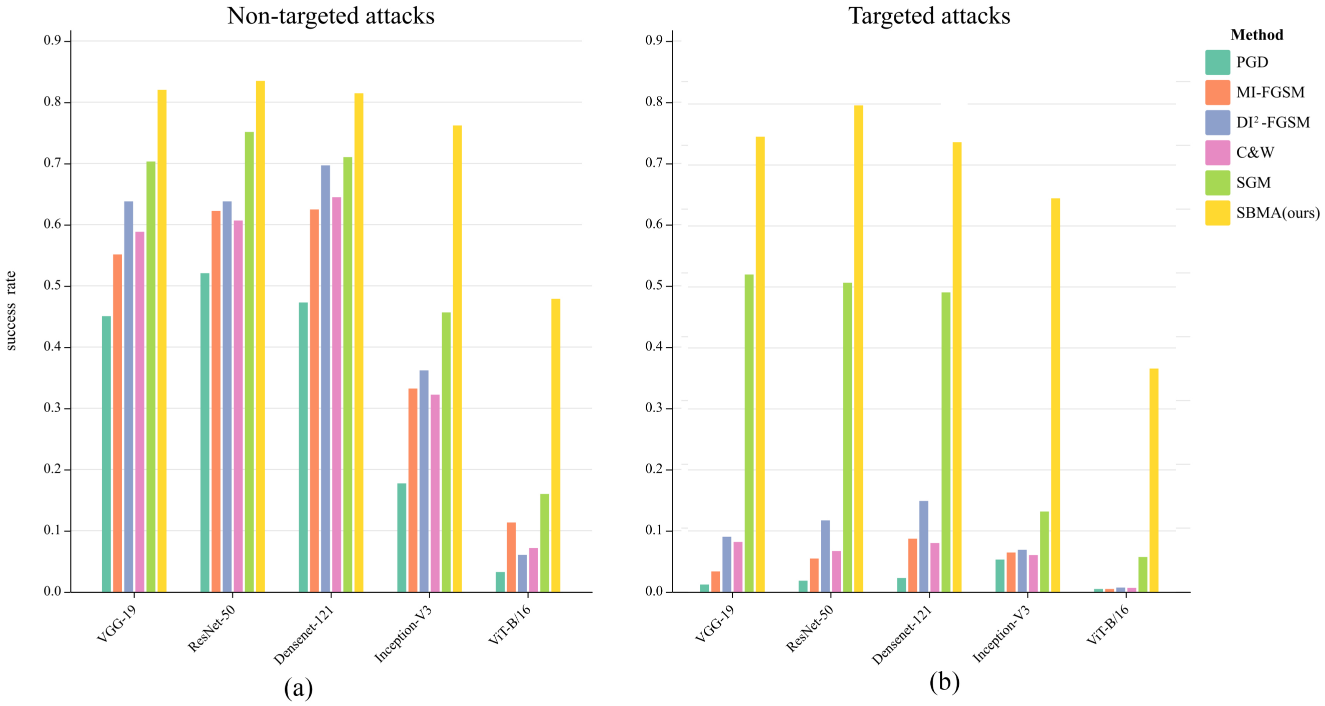
Table 1. Non-targeted attack experiments against normal models. The dataset is Image-Net-1K and the best results are in bold.

Surrogate Model	Attack Method	VGG-19	ResNet-50	DenseNet-121	Inception-V3	ViT-B/16
ResNet-18	PGD [9]	45.05%	52.07%	47.28%	17.73%	3.25%
	MI-FGSM [8]	55.13%	62.24%	62.48%	33.22%	11.34%
	DI^2-FGSM [10]	63.80%	63.80%	69.68%	36.19%	6.05%
	C&W [13]	58.83%	60.67%	64.48%	32.22%	7.17%
	SGM [45]	70.31%	75.14%	71.03%	45.66%	16.00%
	SBMA(ours)	**82.02%**	**83.48%**	**81.46%**	**76.20%**	**47.88%**

Table 2. Targeted attack experiments against normal models. The dataset is Image-Net-1K and the best results are in bold.

Surrogate Model	Attack Method	VGG-19	ResNet-50	DenseNet-121	Inception-V3	ViT-B/16
ResNet-18	PGD [9]	1.21%	1.84%	2.28%	5.30%	0.48%
	MI-FGSM [8]	3.36%	5.47%	8.72%	6.46%	0.48%
	DI^2-FGSM [10]	9.04%	11.73%	14.91%	6.90%	0.71%
	C&W [13]	8.19%	6.70%	8.01%	6.04%	0.66%
	SGM [45]	52.04%	50.69%	49.11%	13.18%	5.72%
	SBMA(ours)	**74.63%**	**79.77%**	**73.10%**	**64.54%**	**36.62%**

From Figure 3, we can see more intuitively that the transferability of the traditional methods decrease significantly when transforming from non-targeted attacks(subplot (a)) to targeted attacks(subplot (b)), while our SBMA method can still maintain a high transferability. This indicates that network models with different structures have a significant impact on the methods guided by the decision boundaries of the classifiers, but have less impact on the methods guided by the sample probability distributions.



(a)

(b)

Figure 3. Comparison of success rates when attacking normal models. subplot (a) represents non-targeted attacks and subplot (b) represents targeted attacks.

Table 3 and Figure 4 represent the comparison of the attacks on the normal model of DenseNet-121 and its robust models. From the results, we can see that the success rate of the PGD method and the FGSM series methods almost fail when transfer to attack the robust models, and only the SGM method maintains a relatively high success rate. In contrast, our SBMA method still maintains the highest success rate, although it also has some indicator drop. This shows that our approach, guided by the probability distribution of the samples, is able to achieve high transferability by ignoring the negative effects of classifier robustness improvement, while the rest of the methods clearly fail in the face of the robustness model.

Table 3. Targeted attack experiments against the robust models of DenseNet-121 with ImageNet-1K dataset, the best results are in bold.

Surrogate Model	Attack Method	Normal	Adversarial Training	SIN	Augmix
ResNet-18	PGD [9]	2.28%	0.59%	1.26%	1.21%
	MI-FGSM [8]	8.72%	1.83%	4.71%	4.54%
	DI^2-FGSM [10]	14.91%	3.60%	8.40%	8.10%
	C&W [13]	8.01%	1.64%	4.23%	4.08%
	SGM [45]	49.11%	9.22%	28.53%	27.50%
	SBMA(ours)	**73.10%**	**15.13%**	**60.27%**	**54.62%**

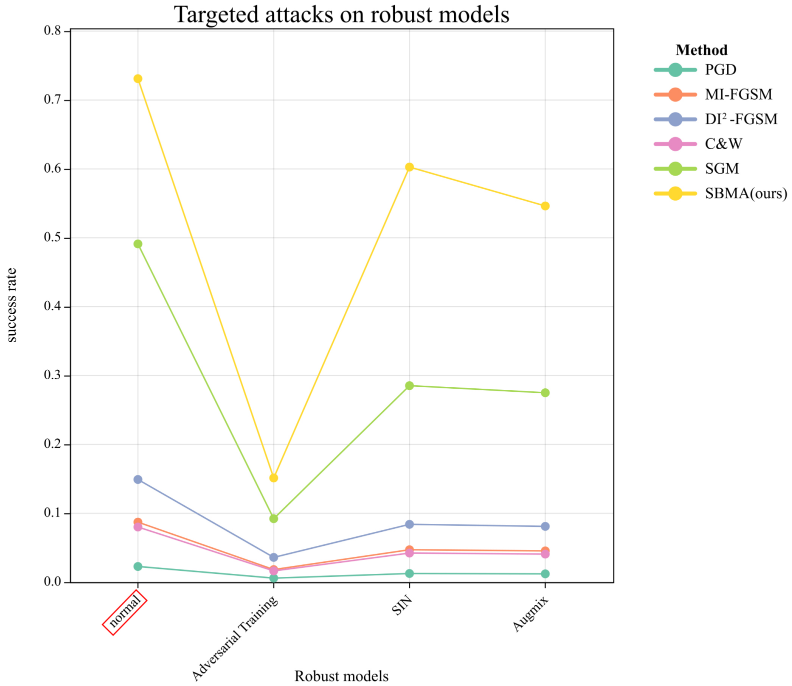


Figure 4. Comparison of the success rates of targeted attacks on the robust models, marked in red on x-axis is the normal model of DenseNet-121, and the rest are the DenseNet-121 models that have been trained by the robust methods.

4.2.2. Experiments On The CIFAR-100 Dataset

As shown in Table 4, when the dataset is replaced with CIFAR100 and the target attacks are implemented, we still get similar conclusions as in Table 2, indicating that our method is still effective in the face of different datasets. However, we found that all metrics decreased in varying degrees when compared to the results under the ImageNet-1K dataset, as shown in subplot (a) of Figure 5. The CIFAR-100 dataset is divided into 100 classes and the training set contains only 500 images of 3×32×32 for each class, which is fewer in number and smaller in size compared to ImageNet-1K. Considering that the estimation of the sample probability distribution requires more high quality of samples, we input ImageNet-1K images of different sizes and different numbers to validate our suspicion. The results are shown in Table 5, subplots (b) and (c) of Figure 5, where the more the number of input images and the larger the size, the higher the attack success rate is obtained.

Table 4. Targeted attack experiments against normal models. The dataset is CIFAR100, and the best results are in bold.

Surrogate Model	Attack Method	VGG-19	ResNet-50	DenseNet-121	Inception-V3	ViT-B/16
ResNet-18	PGD [9]	0.98%	1.56%	1.93%	4.49%	0.37%
	MI-FGSM [8]	2.84%	4.52%	7.39%	5.95%	0.40%
	DI^2-FGSM [10]	7.66%	9.94%	11.43%	5.85%	0.48%
	C&W [13]	6.93%	5.67%	6.78%	5.12%	0.56%
	SGM [45]	44.08%	40.80%	41.60%	11.16%	3.65%
	SBMA(ours)	**52.68%**	**51.31%**	**49.38%**	**41.56%**	**20.85%**

doi:10.20944/preprints202305.0872.v1

Table 5. Comparison results of input images of different sizes and different number under ImageNet-1K dataset. Surrogate model is ResNet-18 and target model is DenseNet-121.

Number of samples with 3 × 224 × 224		**Size of samples with 1300p**	
1300p	73.10%	3 × 224 × 224	73.10%
650p	62.69%	3 × 112 × 112	67.52%
300p	45.16%	3 × 64 × 64	58.35%
100p	21.08%	3 × 32 × 32	40.44%

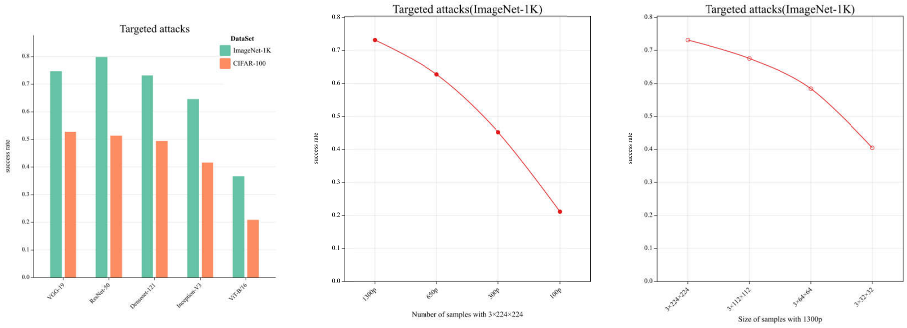


Figure 5. Subplot (a) represents the comparison of attack success rates with our SBMA method under different datasets; subplot (b) represents the comparison of attack success rates under ImageNet-1K dataset with input image size of 3 × 224 × 224 and different number of images; subplot (c) represents the comparison of attack success rates under ImageNet-1K dataset with input image number of 1300p and different sizes.

4.3. Visualization And Interpretability

In this section, the traditional PGD method is chosen for a visual comparison with our SMBA method, which is illustrated from both the perspective of generating adversarial samples and the corresponding feature space.

4.3.1. Generating Adversarial Samples

We appropriately increased the perturbation strength and observed the adversarial samples images and their perturbations generated by different methods from the perspective of targeted attacks. As shown in Figure 6 and Figure 7, the perturbations generated by the PGD method are irregular, while the SMBA method clearly transfers the tree frog (source sample) toward the semantic features of vine snake and corn (target classes), and the generated perturbation has the semantic features form of the target class. It shows that our method is indeed able to learn the semantic features of the target class.

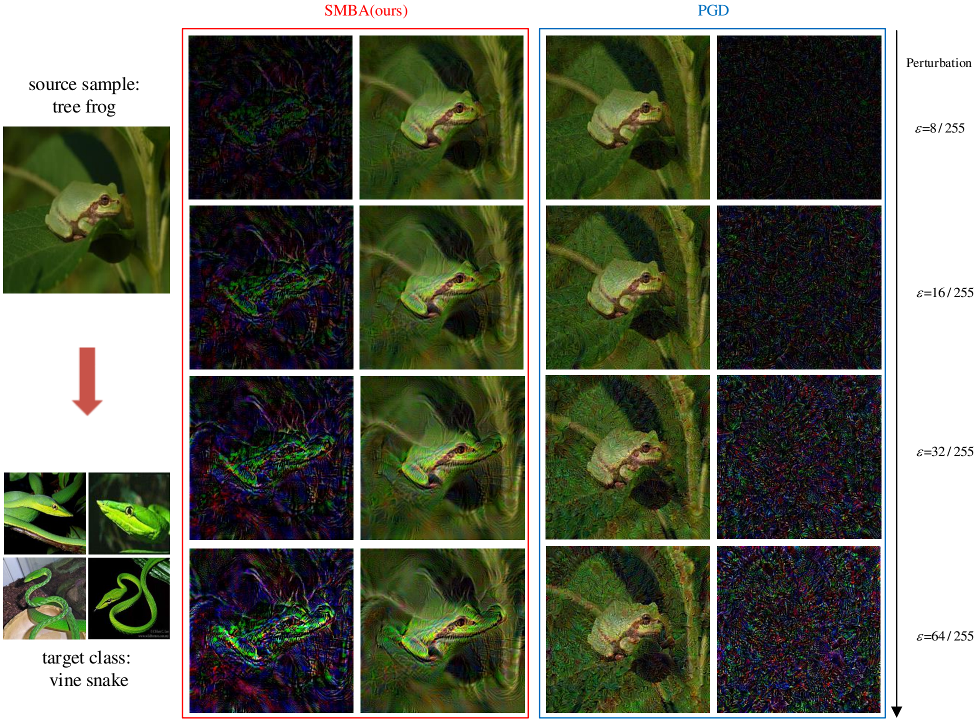


Figure 6. Example 1 of adversarial sample generation for targeted attacks. Source sample is tree frog, and target class is vine snake. The 1st and 2nd columns represent the perturbations(3 times the pixel difference between the adversarial sample and the source sample) and the images of adversarial samples generated by our SMBA methods. The 3rd and 4th columns represent the same content by the PGD methods. The perturbation intensity is increasing from top to bottom.

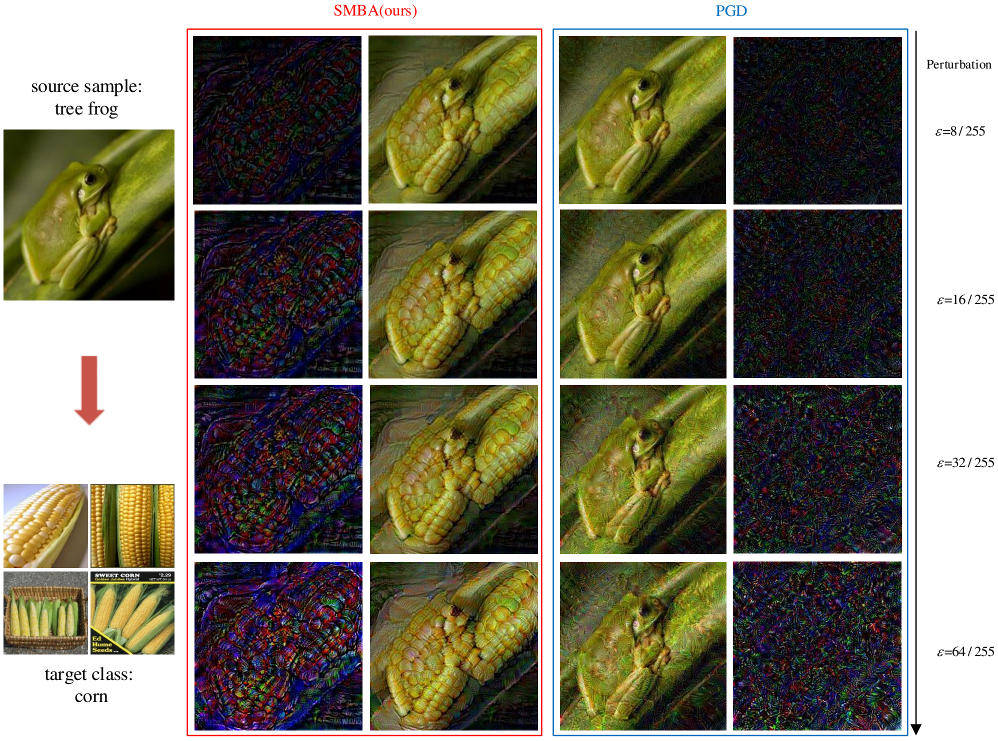


Figure 7. Example 2 of adversarial sample generation for targeted attacks. Source sample is tree frog, and target class is corn.

To further demonstrate the reliability of our SMBA method, we set the source samples as pure gray images and the target classes as multi-classes, as shown in Figure 8. Obviously, the adversarial samples generated by the PGD method are irregularly noisy, while the adversarial samples generated by our SMBA method can still learn the semantic features of the target classes clearly.

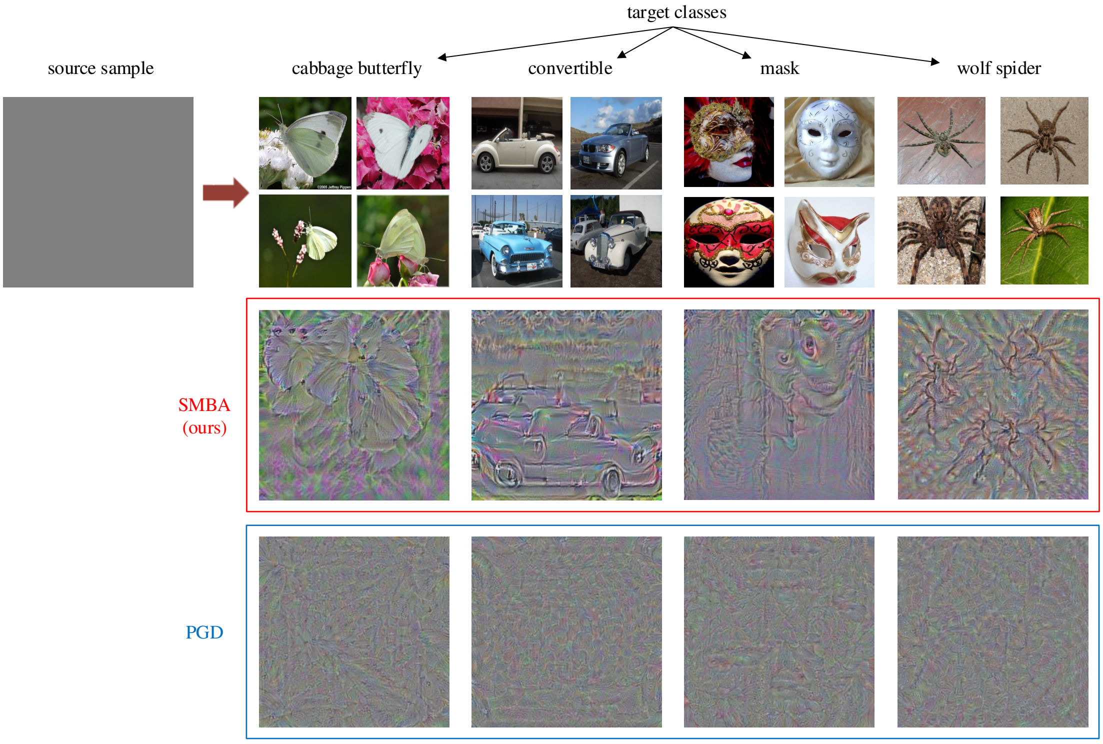


Figure 8. Perturbation imaging of targeted attacks. Source sample is a pure gray image with each pixel of 0.5 , the 1st row indicates multiple target classes, the 2nd and 3rd rows indicate the images of the adversarial samples generated by different methods.

4.3.2. Corresponding Feature Space

In the non-targeted attacks, we visualize the two-dimensional feature space distribution patterns of the source sample and the adversarial sample. As shown in Figure 9, when the adversarial samples generated by the non-targeted attacks on the surrogate model(ResNet-18) transfer to attack the target model(DenseNet-121), the black dashed circle area of subplot (a) shows a mixed state , which means that the adversarial samples generated by the PGD method are not completely removed from the feature space of the source samples. While the adversarial samples generated by the SMBA method have been completely removed from the feature space of the source samples along the black arrow of subplot (b). It indicates that our method can make the adversarial samples completely out of the original probability distribution space to enhance the transferability when performing non-targeted attacks.

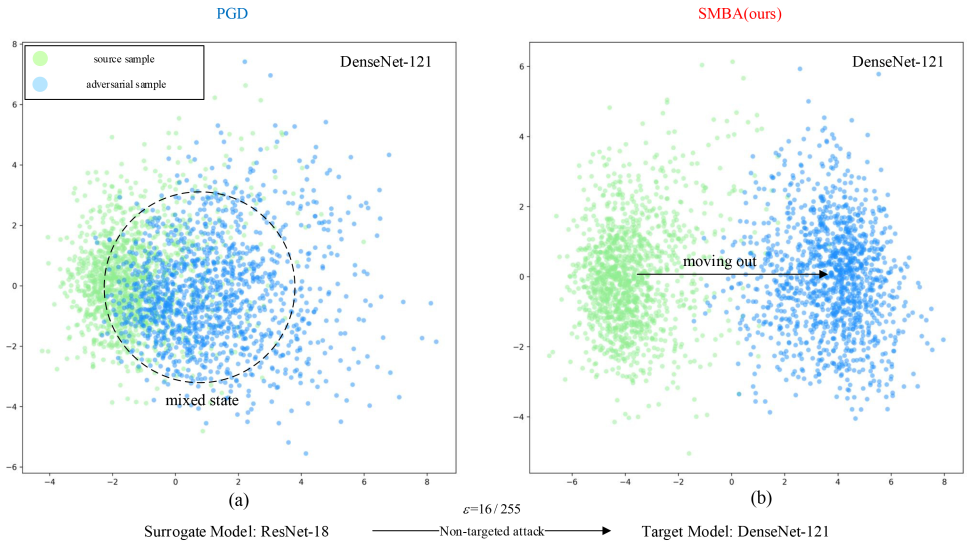


Figure 9. Comparison of the transferability of the adversarial samples generated by non-targeted attacks. We use the adversarial samples generated by non-targeted attacks on surrogate model (ResNet-18) to transfer to attack the target model (DenseNet-121). Subplot (a) represents the two-dimensional feature space distribution patterns of the source samples and the adversarial samples generated by our SMBA method, subplot (b) represents the same content by the PGD method, and the attack intensity is $\varepsilon = 16/255$.

In the targeted attacks, we visualize the wandering paths of the individual adversarial sample generated by different methods in the feature space. As shown in Figure 10, when the perturbation strength increases, for the surrogate model in subplot (a), the adversarial samples generated by different methods gradually move out of the feature space of the source class and wander toward and into the feature space of the target class, and finally the attacks are successful. For the target model in subplot (b), the adversarial samples generated by the PGD method cannot move out of the feature space of the source class (blue ① to ⑦) and cannot cross the decision boundary of the target model, thus the transferable attacks fail. Fortunately, the adversarial samples generated by the SMBA method can still move out of the feature space of the source class and move towards and into the feature space of the target class. The pink dashed circle in subplot (b) indicates the successful transferable attacks (red ⑤ to ⑦), which corresponds to the images with ′ ✓ ′ mark in (c) row, and we can see that the tree frog (source sample) has gradually acquired the semantic features form of the corn (target class). It indicates that our method can make the adversarial sample completely out of the original probability distribution space and wander toward and into the probability distribution space of the target class when conducting the targeted attacks, thus the success rate of transferability is higher.

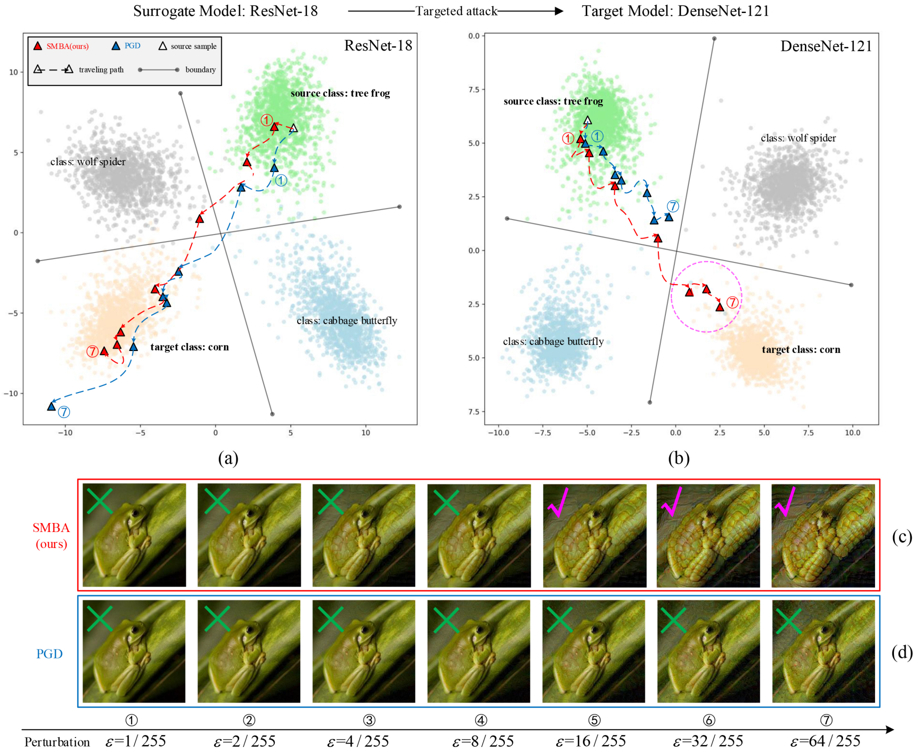


Figure 10. Comparison of the transferability of the adversarial samples generated by targeted attacks. With the perturbation strength increasing, subplot (a) and (b) represent the wandering paths(7 steps from ① to ⑦) in the two-dimensional feature space of the adversarial samples generated by different methods attacking the surrogate model (ResNet-18) and the target model (DenseNet-121). (c) and (d) rows represent the images of the adversarial samples generated by different methods under different perturbation strengths. The images in subplot(b) with the pink dashed circle indicate the successful transferable attacks, and their corresponding images are the images with ′ ✓ ′ mark in (c) row.

5. Conclusions

In this paper, we break through the limitation of traditional adversarial sample generation methods based on the decision boundary guidance of classifiers and reinterpret the generation mechanism of adversarial samples from the perspective of sample probability distribution. We find that if the adversarial samples are directed to move from the space of original probability distributions to the space of target probability distributions, the adversarial samples can learn the semantic features of the target samples, which can significantly improve the transferability and interpretability when faced with classifiers of different structures. Therefore, we propose a probability distribution-guided SMBA method, which uses the SM method to align the gradient of the classifier with the gradient of the sample probability distribution after transforming the classification model into an EBM model, so that the gradient of the classifier can be used to move the adversarial samples out of the original probability distribution and wander toward and into the target probability distribution. Extensive experiments demonstrate that our method shows good performance of transferability when faced with different datasets and models, and can give a reasonable explanation from the perspective of mathematical theory and feature space. Meanwhile, our findings also establish a bridge between probabilistic generative models and adversarial samples, providing a new entry angle for the study of adversarial samples and bringing new thinking to AI security. For future work, we will explore how to apply sample probability distribution estimation methods with non-gradient approximation to adversarial sample generation and attacks.

Author Contributions: Conceptualization, H.L., M.Y. and X.L.; methodology, H.L. and M.Y.; software, H.L. and X.L.; validation, M.Y., J.Z., S.L. and J.L.; formal analysis, H.L. and X.L.; investigation, H.H. and S.L.; data curation,

H.H. and J.L.; writing—original draft preparation, H.L., M.Y. and X.L.; writing—review and editing, H.L., M.Y. and X.L.; visualization, H.L.; supervision, J.Z.; project administration, J.Z.;funding acquisition, S.L. J.L. and X.L.; all authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China [grant numbers 62101571 and 61806215], Natural Science Foundation of Hunan [grant number 2021JJ40685], and Hunan Provincial Innovation Foundation for Postgraduate [grant number QL20220018].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Akhtar, N.; Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access* **2018**, *6*, 14410–14430.
2. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* **2013**.
3. Duan, R.; Ma, X.; Wang, Y.; Bailey, J.; Qin, A.K.; Yang, Y. Adversarial camouflage: Hiding physical-world attacks with natural styles. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1000–1008.
4. Zhang, Y.; Gong, Z.; Zhang, Y.; Bin, K.; Li, Y.; Qi, J.; Wen, H.; Zhong, P. Boosting transferability of physical attack against detectors by redistributing separable attention. *Pattern Recognition* **2023**, *138*, 109435.
5. Welling, M.; Teh, Y.W. Bayesian learning via stochastic gradient Langevin dynamics. Proceedings of the 28th international conference on machine learning (ICML-11), 2011, pp. 681–688.
6. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* **2014**.
7. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. In *Artificial intelligence safety and security*; Chapman and Hall/CRC, 2018; pp. 99–112.
8. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting adversarial attacks with momentum. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 9185–9193.
9. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* **2017**.
10. Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; Yuille, A.L. Improving transferability of adversarial examples with input diversity. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2730–2739.
11. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE, 2016, pp. 372–387.
12. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* **2013**.
13. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. 2017 ieee symposium on security and privacy (sp). Ieee, 2017, pp. 39–57.
14. Su, J.; Vargas, D.V.; Sakurai, K. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* **2019**, *23*, 828–841.
15. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical black-box attacks against machine learning. Proceedings of the 2017 ACM on Asia conference on computer and communications security, 2017, pp. 506–519.
16. Pengcheng, L.; Yi, J.; Zhang, L. Query-efficient black-box attack by active learning. 2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018, pp. 1200–1205.
17. Wu, W.; Su, Y.; Chen, X.; Zhao, S.; King, I.; Lyu, M.R.; Tai, Y.W. Boosting the transferability of adversarial samples via attention. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1161–1170.
18. Li, Y.; Bai, S.; Zhou, Y.; Xie, C.; Zhang, Z.; Yuille, A. Learning transferable adversarial examples via ghost networks. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 11458–11465.

19. Dong, Y.; Su, H.; Wu, B.; Li, Z.; Liu, W.; Zhang, T.; Zhu, J. Efficient decision-based black-box adversarial attacks on face recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7714–7722.
20. Hansen, N.; Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation* **2001**, *9*, 159–195.
21. Brunner, T.; Diehl, F.; Le, M.T.; Knoll, A. Guessing smart: Biased sampling for efficient black-box adversarial attacks. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4958–4966.
22. Shi, Y.; Han, Y.; Tian, Q. Polishing decision-based adversarial noise with a customized sampling. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1030–1038.
23. Rahmati, A.; Moosavi-Dezfooli, S.M.; Frossard, P.; Dai, H. Geoda: a geometric framework for black-box adversarial attacks. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 8446–8455.
24. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Communications of the ACM* **2020**, *63*, 139–144.
25. Xiao, C.; Li, B.; Zhu, J.Y.; He, W.; Liu, M.; Song, D. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610* **2018**.
26. Jandial, S.; Mangla, P.; Varshney, S.; Balasubramanian, V. Advgan++: Harnessing latent layers for adversary generation. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.
27. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2574–2582.
28. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1765–1773.
29. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2921–2929.
30. Joyce, J.M. Kullback-leibler divergence. In *International encyclopedia of statistical science*; Springer, 2011; pp. 720–722.
31. Wang, X.; Zhai, C.; Roth, D. Understanding evolution of research themes: a probabilistic generative model for citations. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 2013, pp. 1115–1123.
32. Alain, G.; Bengio, Y.; Yao, L.; Yosinski, J.; Thibodeau-Laufer, E.; Zhang, S.; Vincent, P. GSNs: generative stochastic networks. *Information and Inference: A Journal of the IMA* **2016**, *5*, 210–249.
33. Taketo, M.; Schroeder, A.C.; Mobraaten, L.E.; Gunning, K.B.; Hanten, G.; Fox, R.R.; Roderick, T.H.; Stewart, C.L.; Lilly, F.; Hansen, C.T. FVB/N: an inbred mouse strain preferable for transgenic analyses. *Proceedings of the National Academy of Sciences* **1991**, *88*, 2065–2069.
34. Germain, M.; Gregor, K.; Murray, I.; Larochelle, H. Made: Masked autoencoder for distribution estimation. International conference on machine learning. PMLR, 2015, pp. 881–889.
35. Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A.; others. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems* **2016**, *29*.
36. An, J.; Cho, S. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE* **2015**, *2*, 1–18.
37. Llorente, F.; Curbelo, E.; Martino, L.; Elvira, V.; Delgado, D. MCMC-driven importance samplers. *Applied Mathematical Modelling* **2022**, *111*, 310–331.
38. Du, Y.; Mordatch, I. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems* **2019**, *32*.
39. Hyvärinen, A.; Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research* **2005**, *6*.
40. Song, Y.; Garg, S.; Shi, J.; Ermon, S. Sliced score matching: A scalable approach to density and score estimation. Uncertainty in Artificial Intelligence. PMLR, 2020, pp. 574–584.
41. Grathwohl, W.; Wang, K.C.; Jacobsen, J.H.; Duvenaud, D.; Norouzi, M.; Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263* **2019**.

42. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
44. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* **2016**.
45. Wu, D.; Wang, Y.; Xia, S.T.; Bailey, J.; Ma, X. Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990* **2020**.
46. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
47. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
48. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. Proceedings of the AAAI conference on artificial intelligence, 2017, Vol. 31.
49. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; others. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* **2020**.
50. Salman, H.; Ilyas, A.; Engstrom, L.; Kapoor, A.; Madry, A. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems* **2020**, *33*, 3533–3545.
51. Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F.A.; Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231* **2018**.
52. Hendrycks, D.; Mu, N.; Cubuk, E.D.; Zoph, B.; Gilmer, J.; Lakshminarayanan, B. Augmix: A simple method to improve robustness and uncertainty under data shift. International conference on learning representations, 2020, Vol. 1, p. 6.
53. Krizhevsky, A.; Hinton, G.; others. Learning multiple layers of features from tiny images **2009**.
54. Shlens, J. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100* **2014**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.