ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230629

Improving Adversarial Transferability with Ghost Samples

Yi Zhao[a], Ningping Mou[a], Yunjie Ge[a] and Qian Wang[a;*]

[a]The Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, Hubei, China.

Abstract. Adversarial transferability presents an intriguing phenomenon, where adversarial examples designed for one model can effectively deceive other models. By exploiting this property, various transfer-based methods are proposed to conduct adversarial attacks without knowledge of target models, posing significant threats to practical black-box applications. However, these methods either have limited transferability or require high resource consumption. To bridge the gap, we investigate adversarial transferability from the optimization perspective and propose the *ghost sample attack (GSA)*. GSA improves adversarial transferability by alleviating the overfitting issue of adversarial examples on the surrogate model. Based on the insight that a slight shift of the adversarial example is similar to a minor change in the decision boundary, we aggregate gradients of perturbed adversarial copies (named ghost samples) to efficiently achieve a similar effect to calculating gradients of multiple ensemble surrogate models. Extensive experiments demonstrate that GSA achieves state-of-the-art adversarial transferability with restricted resources. On average, GSA improves the attack success rate by 4.8% on normally trained models compared to state-of-the-art attacks. Additionally, GSA reduces the computational cost by 62% compared with TAIG-R. When combined with other methods, GSA further improves transferability to 96.9% on normally trained models and 82.7% on robust models.

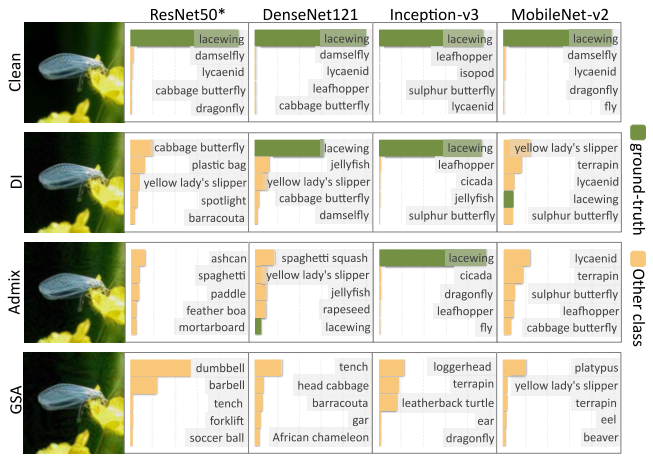


Figure 1: Comparison of the top-5 confidence distribution for GSA and two representative transfer-based attacks. All adversarial examples are crafted on ResNet50 with the maximum perturbation of $\epsilon = 16$ and the iteration of 10. GSA generates more transferable adversaries that successfully attack various black-box models. Best viewed in color and magnified.

1 Introduction

Adversarial examples [32], referring to malicious inputs with intentionally crafted imperceptible perturbations, have emerged as a significant security concern for deep neural networks (DNNs) in practical applications, such as autonomous driving [6] and face recognition [27]. Existing adversarial attacks perform exceptionally well in white-box scenarios where adversaries have full knowledge of the target model [19, 2]. However, most attacks are often less effective in black-box settings where adversaries have no knowledge of the target model, particularly when the target model has defense mechanisms [35, 40, 10]. In black-box settings, the most prevalent approaches are query-based and transfer-based. Query-based methods usually require a large number of queries on the target model to approximate its gradients [8, 15], making them hard to implement when the target model has query restrictions. While transfer-based methods provide a more practical way to conduct black-box attacks by using a surrogate model to generate highly transferable adversarial examples that can fool black-box models [4, 42]. Therefore, we focus on the transfer-based approach in this work.

Transfer-based adversarial attacks are designed by leveraging the capacity of adversarial examples crafted for one model to effectively deceive other models with different architectures and parameters. This property of adversarial transferability was first identified in traditional white-box attacks, but the limited transferability of these attacks makes them hard to implement in practice [19]. Recently, various methods have been proposed to improve adversarial transferability, such as advanced gradient calculation [4, 37], feature disruption[39, 14], input transformation [42, 5, 21, 38], *etc*. Despite the great efficiency of advanced gradient-based methods, their attack success rates are limited in black-box scenarios. Feature disruption methods can achieve superior transferability at the cost of more computational overhead. Input transformation methods may have achieved a relative balance between transferability and overhead. But this kind of method is less transferable to target models that are significantly different from the surrogate models or have defense mechanisms. We can see that existing methods often come with the trade-off of either having limited transferability in black-box scenarios or requiring excessive computational resources. There remains a

* Corresponding Author. Email: qianwang@whu.edu.cn.

gap in performance between white-box attacks and efficient transfer-based black-box attacks.

To bridge this gap, we propose an effective and efficient transfer-based attack named *ghost sample attack (GSA)*. It mitigates the model-dependence issue of adversarial examples, *i.e.*, overfitting the surrogate model, which may hinder the adversarial transferability. Heuristically, adversarial transferability occurs because neural networks tend to have similar decision boundaries in high-dimensional input spaces [36]. When the decision boundary of the target model is less similar to that of the surrogate model, most existing adversarial attacks, which are highly dependent on the decision surface of the surrogate model, fail to achieve high transferability. Therefore, we attempt to further improve adversarial transferability on diverse models by alleviating the dependency of adversarial examples on the surrogate decision boundary. A simple and intuitive way to achieve this goal is generating adversarial examples on several ensemble surrogate models. However, this method is hard to conduct in practice because of its high resource consumption. In GSA, we find an efficient alternative to mitigate the model-dependence issue of adversarial examples. Specifically, GSA generates perturbed copies of adversarial examples, referred to as ghost samples, in each iteration of the optimization. Based on the insight that a slight shift of the adversarial example is similar to a minor change in the decision boundary, we aggregate gradients of these ghost samples to efficiently achieve a similar effect to calculating gradients of multiple ensemble surrogate models, which leads to better transferability against diverse target models. To highlight the superior transferability of GSA, we illustrate the top-5 confidence distribution of an adversarial example generated by GSA and other attack methods in Figure 1.

Experimental results on the ImageNet dataset demonstrate that GSA can achieve state-of-the-art adversarial transferability with limited resources. It improves the average attack success rate on six normally trained models by 4.8% compared to state-of-the-art transfer-based attacks with similar resource consumption. GSA also reduces the computational cost by 62% compared with the state-of-the-art feature disruption methods (TAIG-R). Additionally, GSA can be easily integrated as a plug-and-play component and used in combination with other methods to further enhance adversarial transferability (96.9% on normally trained models and 82.7% on robust models). Our code is available at https://github.com/SincereJoy/GSA.git.

Our main contributions are summarized as follows:

- We propose the *ghost sample attack (GSA)* that leverages perturbed copies of adversarial examples, named ghost samples, to effectively alleviate the overfitting issue of adversarial examples on the surrogate model, thus improving the adversarial transferability.
- We employ the aggregated gradient of ghost samples as an efficient alternative to the gradient of a single adversarial example on multiple ensemble surrogate models, guiding optimization of the adversarial sample toward a more transferable direction.
- Extensive experiments demonstrate that GSA exceeds the performance of state-of-the-art transfer-based attacks by 4.8% on average, achieving a superior attack success rate with less resource consumption. When combined with other methods, GSA further improves the average attack success rate to 96.9% and 82.7% on normally trained models and robust models, respectively.

2 Related Work

Adversarial examples have drawn tremendous attention since research [32] revealed their severe hazards and transferability between different DNNs. To craft an adversarial example, the typical approach is to use the gradients of the model directly to optimize a standard objective function, such as the Basic Iterative Method (BIM) [19]. These white-box attack methods can serve as backends in transfer-based black-box attacks. Recently various kinds of methods are proposed to improve adversarial transferability. In this section, we discuss three categories of widely-used methods.

Advanced gradient calculation. Inspired by the optimization algorithms in the training process [24, 23], some works proposed advanced gradient-based methods, such as Momentum Iterative FGSM (MIFGSM) [4], Nesterov Iterative FGSM (NIFGSM) [21], and Variance tuning MIFGSM (VMIFGSM) [37]. These methods were dedicated to finding a better local optimum for the adversarial optimization problem. They generate adversarial examples totally based on the decision surface of the surrogate model. When the target model is largely different from the surrogate model, these methods would become less effective.

Feature disruption. Based on the fact that different DNNs learn similar features from the same sample [16, 7], some works utilize attention maps or other feature attribution techniques to disturb more transferable internal features. Feature importance-aware attack (FIA) [39] and random patch attack (RPA) [44] used masked images and gradient aggregation to highlight and disrupt essential object-aware features. TAIG [14] found highly transferable adversarial examples by Integrated Gradients [30]. However, feature disruption-based methods usually achieve satisfying attack performance at the cost of huge computational overhead.

Input transformation. Some studies leveraged input transformation to promote adversarial transferability. Diverse input method (DI) [42] randomly resized the input with a fixed probability. Translation-invariant attack (TI) [5] approximated the ensemble gradient of translated images by convolving the gradient of the untranslated image. Scale-invariant attack (SI) [21] optimized the adversarial perturbations over the scale copies of the input. Admix [38] applied a specially designed Mixup to transform the input image with a small proportion of images randomly sampled from other categories. However, existing input transformation-based attacks show limited transferability when attacking models with diverse architecture and advanced defense mechanisms.

To achieve high transferability with a low cost, we investigate adversarial transferability from the perspective of optimization and propose a novel input transformation-based attack. Instead of finding a better local optimum for the adversarial optimization problem on the surrogate model, we focus on mitigating the overfitting issue of adversarial examples on the surrogate model and draw on techniques for improving model generalization, *e.g.*, regularization and data augmentation, to improve the adversarial transferability.

3 Methodology

In this section, we first introduce the preliminaries of adversarial attacks. Then we explain our motivation from the optimization perspective. Finally, we describe the details of our proposed method and provide a transferability analysis.

3.1 Preliminaries

Let $\mathcal{X}$ be the set of input images under consideration for a classification task and $\mathcal{Y}$ be the set of output labels. Given a black-box target model $f_t(\boldsymbol{x};\theta_t) : \boldsymbol{x} \in \mathcal{X} \mapsto y \in \mathcal{Y}$ with unknown parameters

θ_t and a white-box surrogate model $f_s(\boldsymbol{x};\theta_s) : \boldsymbol{x} \in \mathcal{X} \mapsto y \in \mathcal{Y}$, transfer-based adversarial attacks aim to craft an adversarial example $\boldsymbol{x}^{adv}$ against f_s with the perturbation limited by the l_p-norm magnitude ϵ, *i.e.*, $\|\boldsymbol{x}^{adv} - \boldsymbol{x}\|_p \leq \epsilon$, which is then transferred to the target model to cause misclassification, *i.e.*, $f_t(\boldsymbol{x}^{adv};\theta_t) \neq y$.

For the white-box surrogate model f_s, we can formulate the adversarial attack as the following optimization problem:

$$\arg\max_{\boldsymbol{x}^{adv}} J(\boldsymbol{x}^{adv}, y), \quad s.t. \|\boldsymbol{x}^{adv} - \boldsymbol{x}\|_p \leq \epsilon, \tag{1}$$

where $J(\cdot,\cdot)$ is the loss function, and ϵ is the magnitude of adversarial perturbations. To align with previous methods, $p = \infty$ is adopted to measure the distance between $\boldsymbol{x}^{adv}$ and $\boldsymbol{x}$ in this work.

3.2 Motivation

Previous transfer-based attacks either have limited transferability or require high resource consumption. New insights on improving adversarial transferability with limited resources need to be discovered to bridge the performance gap between white-box attacks and efficient transfer-based attacks.

In this work, we investigate adversarial transferability from the perspective of optimization. Similar to training a model, generating an adversarial example can also be regarded as an optimization problem [4, 21]. In the model training process, we iteratively update the model parameters with gradients computed through backpropagation on the training data to minimize the loss function. Similarly, when generating an adversarial example, we optimize the adversarial example with backpropagated gradients generated by the model to maximize the loss. In the context of transfer-based attacks, the surrogate model can be seen as the training data on which we "*train*" the adversarial examples. Also, the target model can be treated as the out-of-distribution domain where we want the "*trained*" adversarial examples to generalize. Therefore, the transferability of adversarial examples can be seen as an analogue to the generalizability of models, *i.e.*, adversarial examples transferring from a surrogate model to target models can be considered similar to models generalizing from the training data to out-of-distribution domains.

Based on the analogy between adversarial transferability and model generalization, we attempt to draw inspiration from techniques designed for improving model generalization. Among various methods that benefit model generalization, regularization [29, 43] and data augmentation [17] are two categories of commonly used approaches. Researches have shown that some data augmentation methods also have an effect of regularization, and Gaussian noise data augmentation is a particularly effective one [1, 43]. Therefore, we try to integrate noise data augmentation methods into the adversarial optimization process, which is detailed in the subsequent sections of the paper.

3.3 Ghost Sample Attack

As described above, the surrogate model can be seen as data on which we "*train*" the adversarial example. Based on this, training a model with noise data augmentation corresponds to generating adversarial examples on multiple ensemble surrogate models with slightly different decision boundaries. Considering model ensemble attacks require huge memory and computational resources, we attempt to transform the ensemble of models into an ensemble of perturbed adversarial examples. Figure 2 presents a toy example to illustrate how this works.

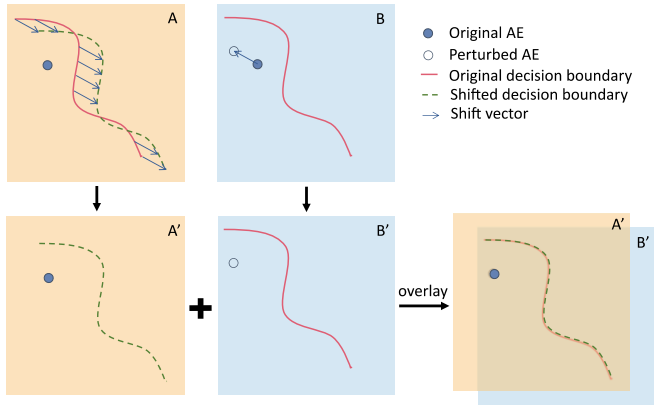


Figure 2: Schematic diagrams of the equivalence between adversarial example shift and decision boundary shift.

In Figure 2, diagram A depicts the change in decision boundary, *i.e.*, the original decision boundary (the red curve) slightly shifts to the new boundary (the dashed green curve) with a vector $\vec{a}$. Diagram B presents that the original adversarial example shifts to the perturbed one with a vector $\vec{b}$, where $\vec{b} = -\vec{a}$. Diagrams A′ and B′ show the results of A and B, respectively. By overlaying A′ and B′, we can see that the relative position of the adversarial example and the decision boundary is the same, which indicates that adversarial example shift and boundary shift result in the same gradient direction (relative position). Therefore, the aggregated gradient of perturbed adversarial copies, referred to as ghost samples, could be considered as an alternative to the gradient of an adversarial example toward multiple ensemble models with slightly different decision boundaries. Based on this, we can replace the ensemble of models with perturbed adversarial examples to improve adversarial transferability.

Given the adversarial example $\boldsymbol{x}_t^{adv}$ in the t-th iteration, we randomly generate $N-1$ perturbed ghost samples within the neighborhood region of $\boldsymbol{x}_t^{adv}$ under $N(\boldsymbol{x}_t^{adv}, \sigma^2 I)$ distribution. Then we calculate the aggregation of gradients as the following weighted sum:

$$\boldsymbol{g}_{t+1} = \sum_{i=0}^{N-1} w_i \nabla_{\boldsymbol{x}_{t,i}^{gst}} J(\boldsymbol{x}_{t,i}^{gst}, y), \tag{2}$$

where $\boldsymbol{x}_{t,0}^{gst} = \boldsymbol{x}_t^{adv}$, $\boldsymbol{x}_{t,i}^{gst} (i > 0)$ is the i-th ghost sample of $\boldsymbol{x}_t^{adv}$, w_i is the pre-defined weight for the corresponding gradient, and y is the ground-truth label of $\boldsymbol{x}$. In practice, we set all $w_i = \frac{1}{N}$ to give all ghost samples equal importance in determining the optimization direction since they are randomly sampled. Additionally, the experimental results presented in Section 4.6 show that keeping the original sample's weight the same as that of the ghost samples is beneficial. Therefore we simplify Equation 2 to Equation 3 and summarize the algorithm of GSA in Algorithm 1.

$$\boldsymbol{g}_{t+1} = \frac{1}{N} \sum_{i=0}^{N-1} \nabla_{\boldsymbol{x}_{t,i}^{gst}} J(\boldsymbol{x}_{t,i}^{gst}, y). \tag{3}$$

3.4 Transferability Analysis

Before conducting extensive experiments, we first analyze the loss landscape of GSA to study whether GSA could benefit adversarial transferability. Researchers have shown the relationship between the

Algorithm 1 Ghost Sample Attack

Input: A classifier f; a clean image $\boldsymbol{x}$ with ground-truth label y.
Parameter: The magnitude of the perturbation ϵ; the number of iteration T; the update step size α; the number of ghost samples N; the standard deviation of Gaussian noise σ.
Output: An adversarial example $\boldsymbol{x}^{adv}$.

Let $\boldsymbol{g}_0 = 0$; $\boldsymbol{x}_0^{adv} = \boldsymbol{x}$.
for $t = 0$ to $T-1$ **do**
  Let $\boldsymbol{x}_{t,0}^{gst} = \boldsymbol{x}_t^{adv}, \boldsymbol{g}_{t+1} = 0$.
  Sample $N-1$ ghost samples $\boldsymbol{x}_{t,\cdot}^{gst}$ under the distribution of $\mathcal{N}(\boldsymbol{x}_t^{adv}, \sigma^2\boldsymbol{I})$.
  Get the loss function $J(\boldsymbol{x}; y)$.
  for $i = 0$ to $N-1$ **do**
    Calculate the gradient by $\boldsymbol{g}_{t+1}^i = \nabla_{\boldsymbol{x}_{t,i}^{gst}} J(\boldsymbol{x}_{t,i}^{gst}, y)$.
    Aggregate the gradient by $\boldsymbol{g}_{t+1} = \boldsymbol{g}_{t+1} + \frac{1}{N}\boldsymbol{g}_{t+1}^i$.
  end for
  Update $\boldsymbol{x}_{t+1}^{adv}$ by applying the gradient sign:

$$\boldsymbol{x}_{t+1}^{adv} = \boldsymbol{x}_t^{adv} + \alpha \cdot sign(\boldsymbol{g}_{t+1}).$$

end for
return $\boldsymbol{x}^{adv} = \boldsymbol{x}_T^{adv}$

loss landscape and attack transferability [3, 25]. A flatter loss landscape means that the loss function has a relatively small gradient magnitude over a larger region of the input space. This makes the loss function relatively insensitive to small changes in the input, allowing the attacker to generate effective adversarial examples across different models with similar loss landscapes.

We follow the previous work [25] that visualizes the loss flatness around adversarial examples by plotting the loss change when moving the adversarial example along a random direction with different magnitudes. The loss flatness indicates whether $\boldsymbol{x}^{adv}$ locates at a flat local region where the points in the vicinity of $\boldsymbol{x}^{adv}$ have similar loss values as $\boldsymbol{x}^{adv}$. With a low loss flatness value, the erroneous prediction of $\boldsymbol{x}^{adv}$ is unlikely to be affected by the slight change in decision boundary or mild deviations of $\boldsymbol{x}^{adv}$ (usually caused by preprocess-based defense mechanisms), thus achieving better transferability. The implementation details of loss flatness are provided in Appendix A[1].

As shown in Figure 3, the loss flatness value of GSA is significantly lower than that of IFGSM, DI, and Admix. When combined with the MIFGSM method [4] (denoted as MI), the difference between DI and Admix becomes obscure, while the superiority of GSA remains clear. These results verify that adversarial examples generated by GSA locate at a smoother loss landscape and are resistant to changes in the decision boundary caused by model variance and sample deviation caused by defense mechanisms.

4 Experiments

4.1 Setup

Dataset. We conduct our experiments on the widely-used ImageNet-compatible dataset[2] that contains 100.00 images provided by the NIPS 2017 adversarial competition.

[1] The appendix is available at https://drive.google.com/file/d/1jcbRxoBOJA-0AgraWfJ4RI-67GAjpgnH/view?usp=sharing
[2] https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition/dataset

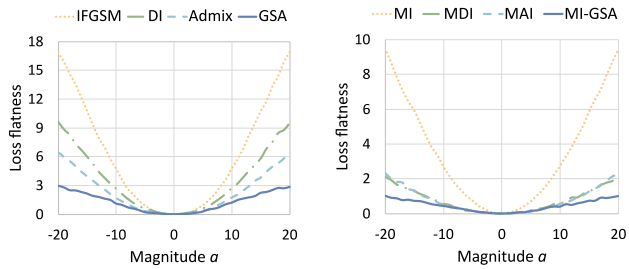


Figure 3: The loss flatness visualization of various attacks on the surrogate model ResNet50. MDI represents the combination of MI and DI. MAI represents the combination of MI and Admix.

Models. We use ResNet50 [9], VGG16 [28], and Inception-v3 [31] as surrogate models to generate adversarial examples. Due to the page limit, experiment results on VGG16 and Inception-v3 are provided in Appendix B.1. For target models, we considered six normally trained models (*e.g.*, DenseNet121 [13], VGG16, Inception-v3, MobildeNet-v2 [26], SENet154 [12], and PNASNet-5-Large [22]), two ensemble adversarially trained models (*e.g.*, adv-Inception-v3 [18] and ens-adv-Inception-ResNet-v2 [35]), three ViT models (*e.g.*, PiT-S [11], CaiT-S [34], and DeiT-B [33]), and five models with advanced defense mechanisms.

Baseline Methods. We take IFGSM [19] as the basic attack and further consider the competitive input transformation-based methods DI [42], TI [5], SI [21], and Admix [38], and their combinations as baselines. To further illustrate the effectiveness and efficiency of GSA, we also compare GSA with state-of-the-art methods from other categories. We take a feature disruption method TAIG-R [14] and an advanced gradient-based method RAP [25] as additional baselines.

Implementation Details. We adopt both Cross-Entropy (CE) loss and logit loss to conduct untargeted attacks. We also extend GSA to targeted attacks. Following the previous work [25], We set the l_∞ magnitude of perturbation $\epsilon = 16/255$, the number of iterations $T = 10$, and the step size $\alpha = 2/255$. Experimental results with different ϵ are shown in Appendix B.2. We adopt the decay factor $\mu = 1.0$ for MI, the transformation probability $p = 0.5$ for DI, and the Gaussian kernel size of 5×5 for TI, the number of copies $m = 5$ for SI. For Admix, we follow its original settings: the number of copies $m_1 = 5$ and the number of random samples $m_2 = 3$ with $\eta = 0.2$. To ensure fair comparisons, we set the sampling points of TAIG-R as 15. The number of ghost samples is also set to 15, and the standard deviation of Gaussian noise $\sigma = 0.1$. We set a random seed as 1234 in our experiments to guarantee reproducibility. We conduct all experiments on a Ubuntu 18.04.1 server with an NVIDIA GeForce RTX 3090 GPU.

4.2 Evaluation on Normally Trained Models

Single Methods. We first evaluate the attack success rates of GSA and baselines on normally trained models. Specifically, we craft adversarial examples on ResNet50 and test them on six normally trained target models with diverse architectures. Previous works have shown that the commonly-used cross-entropy (CE) loss tends to encounter the vanishing gradient problem [20, 45], *i.e.*, the gradients of the input with respect to the loss function become very small as the number of gradient calculations increases. On the other hand, the logit loss has been found to be more resistant to the vanishing gradient problem and has a superior performance in transferability [45]. Therefore, we use logit loss in addition to the commonly adopted CE

Table 1: Attack success rates (%) on normally trained models. The symbol $*$ indicates the surrogate model used to generate adversarial examples. The attack success rate of the surrogate model is not included in the average attack success rate. The highest values of each column are marked in bold.

Loss	Method	ResNet50*	DenseNet121	VGG16	Inception-v3	MobileNet	SENet154	PNASNet	Average
CE	IFGSM	**100.0**	77.5	73.2	34.8	69.8	49.1	37.7	57.0
	DI	**100.0**	86.6	91.0	48.6	83.3	65.0	61.0	72.6
	TI	**100.0**	83.4	76.3	41.1	73.2	54.4	46.5	62.5
	SI	**100.0**	90.7	79.9	61.2	79.4	54.8	46.1	68.7
	Admix	**100.0**	94.1	90.4	**73.2**	87.7	67.9	59.6	78.8
	GSA	**100.0**	**95.6**	**93.8**	**73.2**	**93.4**	**73.7**	**70.8**	**83.4**
logit	IFGSM	**100.0**	77.1	73.8	35.5	70.5	46.7	32.8	56.1
	DI	**100.0**	86.1	92.0	50.1	82.8	67.0	60.3	73.1
	TI	**100.0**	80.8	77.5	42.5	73.8	53.6	40.9	61.5
	SI	**100.0**	92.9	85.9	66.3	86.6	64.2	52.3	74.7
	Admix	**100.0**	97.3	92.0	76.8	93.6	75.2	64.7	83.3
	GSA	**100.0**	**97.5**	**96.4**	**79.3**	**95.8**	**82.7**	**77.1**	**88.1**

Table 2: Inter-category comparison results on normally trained models. The symbol $*$ indicates the surrogate model used to generate adversarial examples.

Method	Iter	Inception-v3	ResNet50	DenseNet121	VGG16
TAIG-R	10	99.2*	43.0	51.3	48.1
RAP	400	**100.0***	62.1	60.8	65.9
GSA	10	**100.0***	**62.4**	**65.4**	**69.1**
TAIG-R	10	**83.3**	99.0*	94.2	95.6
RAP	400	57.2	**100.0***	91.9	92.9
GSA	10	79.3	**100.0***	**97.5**	**96.4**

loss. Table 1 shows the attack success rates (*i.e.*, the proportion of adversarial examples misclassified by the corresponding target model) of GSA and six baseline methods on six normally trained models, using a ResNet50 surrogate model and $\epsilon = 16/255$. We can observe that GSA achieves higher attack success rates on all target models compared to all baselines, using both CE and logit loss. Among the baseline methods, Admix has the best adversarial transferability, and GSA outperforms Admix with an average improvement of 4.6% and 4.8% using CE and logit loss, respectively.

To further demonstrate the effectiveness of GSA, we compared it with other state-of-the-art methods, including TAIG-R, a feature disruption-based method, and RAP, an advanced gradient-based method. Typically, feature disruption-based methods require high computational costs, and TAIG-R is no exception because it involves the resource-consuming calculation of Integrated Gradients. Although advanced gradient-based methods are generally more efficient, RAP requires hundreds of iterations to converge, making it computationally expensive.

Table 2 presents the attack success rates of TAIG-R, RAP, and GSA using Inception-v3 and ResNet50 surrogate models and $\epsilon = 16/255$. When using Inception-v3 surrogate model, GSA outperforms TAIG-R by a significant margin of 14.1%$\sim$21% with the same number of optimization iterations. Furthermore, GSA with only 10 iterations outperforms RAP with 400 iterations by an average margin of 2.6%. When using ResNet50 surrogate model, GSA achieves comparable attack success rates with TAIG-R and outperforms RAP by an average of 10.4%. Moreover, the less satisfying performance of GSA when targeting Inception-v3 may be caused by the special inception module, which uses a combination of convolutional kernels with different sizes and pooling operations to extract features at multiple scales. The fusion of features in Inception models may enhance their robustness against output-level attacks, which generate adversarial examples with respect to the model's outputs. On the other hand, feature disruption attacks like TAIG-R, which aims to disrupt features in the intermediate layers, have inherent advantages in attacking Inception models. Overall, these results demonstrate that GSA is not only superior to other input transformation methods (intra-category methods), but it can also outperform inter-category methods that have high computational costs.

Combinational Methods. Previous studies have demonstrated that the combination of baselines could further enhance adversarial transferability [21, 38]. We also test the attack success rates of GSA incorporated with the other attacks. Specifically, we consider combinations of MI, DI, and TI (denoted as MTDI). Additionally, we include SI and Admix in these combinations, for example, MTDSI represents MTDI combined with SI, and MTDAI represents MTDI combined with Admix. SI and Admix cannot be combined together because SI is a special case of Admix. Moreover, we don't consider the combination of TAIG-R and RAP since their resource consumption will multiply when integrated with input transformation methods. Table 3 reports attack success rates of combined methods on six normally trained models.

Our results show that when using VGG16 as the target model, MTDAI slightly outperforms MTDAI-GSA. However, MTDAI-GSA has higher attack success rates on other target models and in the average results. Similar patterns can also be observed in the comparison of MTDSI and MTDSI-GSA when attacking SENet154 and PNASNet. We suppose the combined GSA occasionally gets worse because of the special structure of the target model. In general, GSA outperforms other combined baselines on the average attack success rate of various models. Besides, by comparing MTDI-GSA with MTDSI-GSA and MTDAI-GSA, we also find that Admix does not provide any improvement to MTDI-GSA on average, and SI even diminishes the effectiveness of MTDI-GSA. On the other hand, GSA can bring additional improvement on the basis of MTDSI and MTDAI, where the average attack success rate is already above 95%. In general, GSA can serve as a plug-and-play component to effectively improve the adversarial transferability of existing attacks.

4.3 *Evaluation on Robust Models*

To evaluate the effectiveness of our method against robust models, we consider two ensemble adversarially trained models, and four models with advanced defense mechanisms, *i.e.*, Feature Denoising (FD) [41], DeepAugment (DA) [10], HGD [20] (rank-1 submission in the NIPS 2017 defense competition), and R&P [40] (rank-2 submission in the NIPS 2017 defense competition). Previous works have

Table 3: Attack success rates (%) of combined methods on normally trained models. The symbol $*$ indicates the surrogate model used to generate adversarial examples. The attack success rate of the surrogate model is not included in the average attack success rate.

Method	ResNet50*	DenseNet121	VGG16	Inception-v3	MobileNet	SENet154	PNASNet	Average
MTDI	99.8	96.1	96.1	79.8	93.8	86.8	84.4	89.5
MTDI-GSA	**100.0**	**99.3**	**98.7**	**95.0**	**97.9**	**95.0**	**95.4**	**96.9**
MTDSI	**100.0**	99.2	97.2	95.8	97.0	**91.4**	**94.1**	95.8
MTDSI-GSA	**100.0**	**99.5**	**97.5**	**98.3**	**98.2**	90.9	93.4	**96.3**
MTDAI	**100.0**	99.2	**98.1**	96.5	97.0	92.6	94.1	96.3
MTDAI-GSA	**100.0**	**99.6**	97.9	**98.3**	**98.3**	**92.9**	**94.5**	**96.9**

Table 4: Attack success rates (%) on robust models. The highest values of each column are marked in bold.

Method	Inc-v3$_{adv}$	IncRes-v2$_{ens}$	FD	DA	HGD	R&P	PiT-S	CaiT-S	DeiT-B	Average
MTDI	68.9	53.8	54.5	68.6	69.6	40.5	54.7	45.9	41.2	55.3
MTDI-GSA	**90.4**	**87.9**	**59.4**	**92.0**	**91.0**	**83.7**	**79.0**	**75.1**	**68.9**	**80.8**
MTDSI	87.8	82.5	56.7	85.1	89.8	73.9	72.2	64.1	57.5	74.4
MTDSI-GSA	**92.8**	**91.1**	**60.7**	**92.6**	**94.3**	**89.4**	**77.4**	**72.7**	**64.8**	**81.8**
MTDAI	92.5	85.7	56.8	87.6	93.2	78.5	76.5	70.7	61.2	78.1
MTDAI-GSA	**94.8**	**91.3**	**60.5**	**92.9**	**95.8**	**88.9**	**79.2**	**74.9**	**65.9**	**82.7**

shown that ViTs have superior adversarial robustness, and transferring adversarial attacks from CNN models to ViTs is challenging. To assess the effectiveness of GSA in this context, we also conducted experiments on three ViT models.

Table 4 illustrates the attack success rates of combined methods when attacking the robust models. The results indicate that combining the baseline methods with GSA can significantly improve attack success rates on all robust models and increase the average results by 4.6%∼25.5%. Additionally, MTDI-GSA outperforms MTDSI and MTDAI in terms of the average attack success rates by 6.4% and 2.7%, respectively. We also find that GSA is good at breaking input-level defenses, *e.g.*, DA, HGD, and R&P, but is relatively less effective when attacking models with feature-level defenses, such as FD. Although ViTs show relatively better robustness than CNN models, GSA can still enhance the transferability on ViT target models by an average improvement of 16.3%. These results indicate that GSA has superior adversarial transferability when attacking robust models, making it practical and effective for a wide range of applications.

4.4 Evaluation in the Targeted Attack Scenario

While GSA is originally designed for untargeted attacks, it can be adapted for targeted attacks by simply modifying the objective function to maximize the logit output of the desired target class. This is done by replacing the ground-truth label with the target label and taking the negative of the loss function. For instance, the logit loss for the targeted attack is $J_{logit}(\boldsymbol{x}, y_t) = l_{y_t}(\boldsymbol{x})$, where $l_{y_t}(\cdot)$ denotes the logit output with respect to the target class y_t. Then we optimize the adversarial example to maximize the logit output with respect to the target class through the gradient descent process.

Table 5 illustrates the targeted attack success rates of adversarial examples generated on Resnet50 with $\epsilon = 16/255$ and 100.0 iterations. We can see that GSA outperforms all baseline methods by a significant margin in the targeted setting, reaching an average targeted attack success rate of 78.8%. These results demonstrate that GSA can be easily extended to targeted attacks and achieve high targeted attack success rates, making it a versatile method for both untargeted and targeted attack scenarios.

4.5 Computational and Time Cost

To demonstrate the efficiency of GSA, we conducted a quantitative and qualitative analysis of its cost compared to Admix and TAIG. For a fair comparison, we set $N = 15$ for GSA, the number of sampling points $n = 15$ for TAIG-R, $m_1 = 3, m_2 = 5$ for Admix, and iteration $T = 10$ for all three methods. These settings allow an equal number of forward and backward calculations among the three methods. Therefore, the efficiency difference is a result of the extra operations used by the three methods.

First, we analyze the computational cost qualitatively. The computation costs of forward and backward operations are equal in these three methods since we set $T \times m_1 \times m_2 = T \times n = T \times N = 15T$. The extra costs of TAIG-R, Admix, and GSA are caused by Integrated Gradients calculation C_{IG}, Mixup C_{mixup}, and perturbing operation $C_{perturb}$, respectively. Mixup and perturbing are widely-used data augmentation techniques, which only bring a little additional overhead. However, the calculation of Integrated Gradient is known to be computationally expensive, resulting in the high overhead of TAIG-R.

We also evaluate the average time cost of generating a single adversarial example for each of the three methods. It costs TAIG-R 1.42s to generate one adversarial example, Admix 0.57s, and GSA 0.56s. In summary, GSA has a similar time cost to Admix and reduces 62% of the time cost compared with TAIG-R. Although Admix has a similar overhead to GSA, it is observed to have lower adversarial transferability. Overall, GSA is able to achieve advanced adversarial transferability at a relatively low overhead, making it a cost-effective choice for the transfer-based black-box attack.

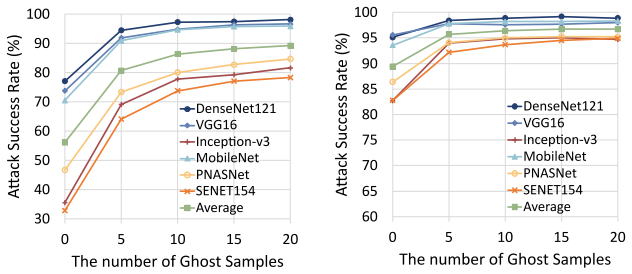
4.6 Ablation Study

We conduct an ablation study on the hyper-parameters, *i.e.*, the number of ghost samples N, the aggregation weight of the original adversarial example w_0, *etc*. All adversarial examples are generated on ResNet50 with 10 iterations.

The Number of Ghost Samples. Figure 4 illustrates how the attack success rate varies on six normally trained models as the number of ghost samples (N) increases from 0 to 20 in increments of 5. When $N = 0$, GSA reverts to the corresponding baseline methods. The

Table 5: Targeted attack success rates (%) on normally trained models. The symbol $*$ indicates the surrogate model used to generate adversarial examples. The attack success rate of the surrogate model is not included in the average attack success rate.

Method	ResNet50*	DenseNet121	VGG16	Inception-v3	MobileNet	SENet154	PNASNet	Average
IFGSM	**100.0**	55.0	58.4	31.1	59.9	37.1	20.8	43.7
DI	**100.0**	76.5	93.1	38.6	74.2	64.8	56.6	67.3
TI	**100.0**	60.1	64.1	34.8	62.1	45.0	28.0	49.0
SI	**100.0**	78.1	70.5	50.4	73.7	50.8	35.1	59.8
Admix	**100.0**	84.1	77.0	59.1	80.3	58.6	43.1	67.0
GSA	**100.0**	**90.5**	**90.8**	**65.3**	**88.4**	**74.4**	**63.2**	**78.8**



(a) GSA (b) MTDI-GSA

Figure 4: Attack success rates on normally trained models with various numbers of ghost samples.

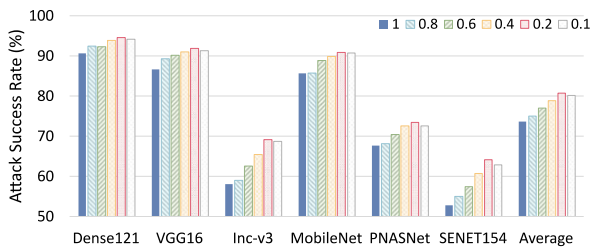


Figure 5: Attack success rates on normally trained models with different aggregation weights w_0.

results reveal that GSA outperforms the baselines by a noticeable margin with just 5 ghost samples. As N increases, the attack success rates continue to rise but eventually plateau when N is greater than 15. We set $N = 15$ in default to ensure better performance and fair comparison between GSA and Admix. Generally, the number of ghost samples can be adjusted in different scenarios to balance adversarial transferability and resource consumption.

The Aggregation Weight. Given the random nature of ghost samples, it's reasonable to assign equal weight to them. In this analysis, we primarily focus on the weight of the original adversarial example w_0. To examine this, we fix $N = 5$ and test the attack success rates of GSA on six normally trained models with $w_0 = 0.1, 0.2, 0.4, 0.6, 0.8, 1$. As shown in Figure 5, GSA achieves the highest attack success rates when $w_0 = 0.2$ ($w_0 = \frac{1}{N}$) on all target models, indicating that assigning equal weights to the original adversarial example and ghost samples results in better performance. From the optimization perspective, any ghost sample with a larger weight will have a more important role in determining the optimization direction, making the adversarial example depends more on the corresponding decision boundary. Under the condition that we cannot determine which decision boundary is better, it is best to treat them equally.

Other Hyper-parameters. We also evaluate the influence of the surrogate model and the magnitude of perturbations ϵ. As for the surrogate model, we additionally consider VGG16 and Inception-v3. GSA outperforms the state-of-the-art attacks by an average attack success rate of 2.3% on VGG16 and 13.4% on Inc-v3. The detailed evaluation results are presented in Appendix B.1. We also evaluate the performance of GSA and the baseline methods with smaller magnitudes of perturbations ($\epsilon = 4/255$ and $\epsilon = 8/255$) to show the effectiveness of GSA in stealthiness-sensitive scenarios. Results show that GSA can generate effective adversarial examples with smaller magnitudes of perturbations and has the advantage of being stealthy while maintaining a high attack success rate. The evaluation results of different ϵ are provided in Appendix B.2.

5 Conclusion

In this paper, we propose a novel transfer-based attack called *ghost sample attack (GSA)*, which improves adversarial transferability by alleviating the overfitting issue of adversarial examples on the surrogate model. By considering adversarial transferability as an analogue to model generalization, we draw inspiration from techniques used to improve model generalization and integrate noise data augmentation into the adversarial optimization process. Specifically, GSA utilizes the aggregated gradient of perturbed adversarial copies as an efficient alternative to the gradient of a single adversarial example attacking multiple ensemble surrogate models. Extensive evaluations show that GSA outperforms state-of-the-art transfer-based attacks in terms of adversarial transferability while maintaining a relatively low overhead (62% less than TAIG-R). When combined with other methods, GSA achieves an impressive average attack success rate of 96.9% against normally trained models and 82.7% against robust models. We believe that GSA could serve as a cost-effective approach to evaluate model robustness and inspire the security improvements of practical black-box applications.

Acknowledgements

This work was partially supported by the National Key R&D Program of China (2020AAA0107701), and the NSFC under Grants U20B2049 and U21B2018.

References

[1] Chris M. Bishop, 'Training with noise is equivalent to tikhonov regularization', *Neural Computation*, **7**(1), 108–116, (1995).
[2] Nicholas Carlini and David Wagner, 'Towards evaluating the robustness of neural networks', in *Proc. of IEEE Symposium on Security and Privacy*, pp. 39–57, (2017).
[3] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli, 'Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks', in *Proc. of 28th USENIX Security Symposium*, pp. 321–338, (2019).
[4] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li, 'Boosting adversarial attacks with momentum', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193, (2018).

[5] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu, 'Evading defenses to transferable adversarial examples by translation-invariant attacks', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4312–4321, (2019).

[6] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song, 'Robust physical-world attacks on deep learning visual classification', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, (2018).

[7] Aditya Ganeshan, Vivek BS, and R Venkatesh Babu, 'Fda: Feature disruptive attack', in *Proc. of IEEE International Conference on Computer Vision*, pp. 8069–8079, (2019).

[8] Yiwen Guo, Ziang Yan, and Changshui Zhang, 'Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks', in *Proc. of Advances in Neural Information Processing Systems*, (2019).

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, (2016).

[10] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al., 'The many faces of robustness: A critical analysis of out-of-distribution generalization', in *Proc. of IEEE International Conference on Computer Vision*, pp. 8340–8349, (2021).

[11] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh, 'Rethinking spatial dimensions of vision transformers', in *Proc. of the IEEE International Conference on Computer Vision*, pp. 11936–11945, (2021).

[12] Jie Hu, Li Shen, and Gang Sun, 'Squeeze-and-excitation networks', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, (2018).

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, 'Densely connected convolutional networks', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, (2017).

[14] Yi Huang and Adams Wai-Kin Kong, 'Transferable adversarial attack based on integrated gradients', in *Proc. of International Conference on Learning Representations*, (2021).

[15] Andrew Ilyas, Logan Engstrom, Anish Athalye, and J Lin, 'Black-box adversarial attacks with limited queries and information', in *Proc. of International Conference on Machine Learning*, pp. 2137–2146, (2018).

[16] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen, 'Feature space perturbations yield more transferable adversarial examples', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7066–7074, (2019).

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, 'Imagenet classification with deep convolutional neural networks', *Communications of ACM*, **60**(6), 84–90, (2017).

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio, 'Adversarial machine learning at scale', *arXiv:1611.01236*, (2016).

[19] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio, 'Adversarial examples in the physical world', in *Artificial Intelligence Safety and Security*, 99–112, (2018).

[20] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu, 'Defense against adversarial attacks using high-level representation guided denoiser', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1787, (2018).

[21] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft, 'Nesterov accelerated gradient and scale invariance for adversarial attacks', in *Proc. of International Conference on Learning Representations*, (2019).

[22] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy, 'Progressive neural architecture search', in *Proc. of European Conference on Computer Vision*, pp. 19–34, (2018).

[23] Yurii Nesterov, 'A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2)', in *Doklady an ussr*, volume 269, pp. 543–547, (1983).

[24] Boris T Polyak, 'Some methods of speeding up the convergence of iteration methods', *Ussr Computational Mathematics and Mathematical Physics*, **4**(5), 1–17, (1964).

[25] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu, 'Boosting the transferability of adversarial attacks with reverse adversarial perturbation', *arXiv:2210.05968*, (2022).

[26] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, 'Mobilenetv2: Inverted residuals and linear bottlenecks', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, (2018).

[27] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter, 'Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition', in *Proc. of ACM SIGSAC Conference on Computer and Communications Security*, pp. 1528–1540, (2016).

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

[29] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, 'Dropout: a simple way to prevent neural networks from overfitting', *The Journal of Machine Learning Research*, **15**(1), 1929–1958, (2014).

[30] Mukund Sundararajan, Ankur Taly, and Qiqi Yan, 'Axiomatic attribution for deep networks', in *Proc. of International Conference on Machine Learning*, pp. 3319–3328, (2017).

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, 'Rethinking the inception architecture for computer vision', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, (2016).

[32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, 'Intriguing properties of neural networks', in *Proc. of International Conference on Learning Representations*, (2014).

[33] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou, 'Training data-efficient image transformers & distillation through attention', in *Proc. of the International Conference on Machine Learning*, pp. 10347–10357, (2021).

[34] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou, 'Going deeper with image transformers', in *Proc. of the IEEE International Conference on Computer Vision*, pp. 32–42, (2021).

[35] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel, 'Ensemble adversarial training: Attacks and defenses', in *Proc. of International Conference on Learning Representations*, (2018).

[36] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel, 'The space of transferable adversarial examples', *arXiv:1704.03453*, (2017).

[37] Xiaosen Wang and Kun He, 'Enhancing the transferability of adversarial attacks through variance tuning', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1924–1933, (2021).

[38] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He, 'Admix: Enhancing the transferability of adversarial attacks', in *Proc. of IEEE International Conference on Computer Vision*, pp. 16138–16147, (2021).

[39] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren, 'Feature importance-aware transferable adversarial attacks', in *Proc. of IEEE International Conference on Computer Vision*, pp. 7639–7648, (2021).

[40] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille, 'Mitigating adversarial effects through randomization', in *Proc. of International Conference on Learning Representations*, (2018).

[41] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He, 'Feature denoising for improving adversarial robustness', in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 501–509, (2019).

[42] 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 *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, (2019).

[43] Xiaojun Xu, Jacky Y Zhang, Evelyn Ma, Hyun Ho Son, Sanmi Koyejo, and Bo Li, 'Adversarially robust models may not transfer better: Sufficient conditions for domain transferability from the view of regularization', in *Proc. of International Conference on Machine Learning*, pp. 24770–24802, (2022).

[44] Yaoyuan Zhang, Yu-an Tan, Tian Chen, Xinrui Liu, Quanxin Zhang, and Yuanzhang Li, 'Enhancing the transferability of adversarial examples with random patch', in *Proc. of International Joint Conference on Artificial Intelligence*, pp. 1672–1678, (2022).

[45] Zhengyu Zhao, Zhuoran Liu, and Martha Larson, 'On success and simplicity: A second look at transferable targeted attacks', in *Proc. of Advances in Neural Information Processing Systems*, pp. 6115–6128, (2021).