2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.00397

Adversarial Robustness via Random Projection Filters

Minjing Dong, Chang Xu[*]
School of Computer Science, University of Sydney
{mdon0736@uni, c.xu@}.sydney.edu.au

Abstract

Deep Neural Networks show superior performance in various tasks but are vulnerable to adversarial attacks. Most defense techniques are devoted to the adversarial training strategies, however, it is difficult to achieve satisfactory robust performance only with traditional adversarial training. We mainly attribute it to that aggressive perturbations which lead to the loss increment can always be found via gradient ascent in white-box setting. Although some noises can be involved to prevent attacks from deriving precise gradients on inputs, there exist trade-offs between the defense capability and natural generalization. Taking advantage of the properties of random projection, we propose to replace part of convolutional filters with random projection filters, and theoretically explore the geometric representation preservation of proposed synthesized filters via Johnson-Lindenstrauss lemma. We conduct sufficient evaluation on multiple networks and datasets. The experimental results showcase the superiority of proposed random projection filters to state-of-the-art baselines. The code is available on GitHub.

1. Introduction

Although Deep Neural Networks (DNNs) have become a popular technique in various scientific fields [16,46,47,50], the vulnerability of DNNs reveals the high risk of deployment in real scenarios, especially under the attack of adversarial examples [11, 28]. Some tiny and imperceptible perturbations to network inputs could result in the major changes of outputs, which can be easily crafted through various adversarial attack strategies [1, 6, 11]. Adversarial attacks could be generally categorized into two streams, the white-box and black-box attacks. In black-box setting, the attackers have no knowledge of victim models but can estimate the strong perturbation via surrogate models or huge number of queries [15, 19]. In white-box setting, the attackers have full knowledge of victim model, including the model parameters, network architecture, and inference strategy [28, 39]. Since the gradients of victim models can be directly fetched, the crafted adversarial examples are more aggressive and the performance under white-box attacks is one of the key criteria of robustness evaluation.

Seeking adversarial robust networks becomes a key challenge when it comes to the deployment of DNNs. One of the most popular and effective techniques is adversarial training [28], which arguments the training data with adversarial examples within a fixed perturbation size. With the involvement of adversarial examples, DNNs are optimized to preserve their outputs for perturbed samples within the ℓ_p ball of all training input data. However, due to the increasingly advanced attack techniques, it is difficult for existing adversarially trained networks to achieve satisfactory robustness against all potential attacks. Furthermore, the training on stronger adversarial examples could hurt the natural generalization of models [52], and there exists a trade-off between robustness and accuracy [51].

Besides the traditional adversarial training, the utilization of randomization in adversarial robustness has been proven effective. For example, Liu *et al*. [25] propose to inject noise which is sampled from Gaussian distribution to the inputs of convolution layers. Some theoretical analyses have shown that randomized classifiers can easily outperform deterministic ones in defending against adversarial attacks [32, 33]. We mainly attribute the improvement of randomization in adversarial robustness evaluation to the fusion of features with noises, which prevents white-box attackers from obtaining the precise gradients of loss with respect to the inputs. Although the involvement of noise in the networks can be an effective defense mechanism, the design of noises, such as the way of injection, the magnitude of noise, *etc*., can also significantly influence the natural generalization of networks in practice. The trade-offs between the adversarial robustness and optimization difficulty are always ignored in the randomized techniques, which limits their superiority to deterministic models.

In this paper, we introduce randomness into deep neural networks with the help of random projection filters. Random projection is a simple yet effective technique

[*]Corresponding author.

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.00397

for dimension reduction, which can approximately preserve the pairwise distance between any two data points from a higher-dimensional space in the projected lower-dimensional space under certain conditions. The theoretical and empirical advantages offered by random projection thus inspire a new way to explore the potential of noise injection with better trade-offs in Convolutional Neural Networks (CNNs). We propose to *partially replace the convolutional filters with the random projection filters*. Theoretically, we extend the scope of Johnson-Lindenstrauss Lemma [41] to cover the convolutions, where partial convolutional filters are randomly sampled from a zero-mean Gaussian distribution. Pairwise example distance can also be approximately preserved under the new convolutions defined by random projection filters, if the number of random projection filters is lower bounded in terms of the weight norm of the remaining convolutional filters. Motivated by these observations, we introduce a simple and efficient defense scheme via the proposed Random Projection Filters (RPF). As parameters of random projection filters are randomly sampled during forwarding, the attackers have no knowledge of upcoming sampled parameters even if in white-box attack settings. The effectiveness of proposed RPF is verified via extensive empirical evaluations in our experiments.

2. Related Work

2.1. Adversarial Attacks

The adversarial examples are first revealed by [39], in which Szegedy *et al*. demonstrated the vulnerability of DNNs to the perturbed inputs within a ℓ_p ball. To explore the vulnerability of DNNs, various attacks have been developed [3, 6, 12, 28]. Adversarial attacks can be generally categorized into white-box attacks and black-box attacks. [12]. In white-box setting, the attackers have access to all the information of victim models, such as the model parameters and structure. Since the gradient information can be fetched, most white-box attacks utilize gradients to obtain the perturbations on the inputs which maximize the loss function. Goodfellow *et al*. introduce an efficient yet effective attack method via the sign of gradients, named Fast Gradient Sign Method (FGSM) [12]. Kurakin *et al*. proposed to adopt basic iterative method for FGSM, which achieves a higher attack success rate [24]. Projection Gradient Descent (PGD) proposed to randomly initialize the adversarial examples within the ℓ_p ball [28]. Carlini and Wagner (CW) attack proposed to treat adversarial attack as a constrained optimization problem [3]. Some feature-disruption-based attacks are introduced [20, 48]. Dong *et al*. explored various momentum-based iterative attack algorithm and proposed Momentum Iterative Fast Gradient Sign Method (MI-FGSM), which showed that the momentum term in the iterations can stabilize the updating direction and prevent local maxima [9]. In black-box setting, the attackers have no access to the information of victim models. One of the sub-categories of black-box attack is query-based methods where the perturbations can be approximated via huge number of queries [1, 4, 13]. However, massive queries can be easily detected in real scenarios, which motivates some works to focus on efficiency [37, 38]. Another sub-category lies in transfer-based methods where the attacks have full access to a surrogate model and aims at generating adversarial examples with higher transferability [26, 31, 42]. Although transfer-based methods dismiss the massive queries, they can hardly achieve satisfactory attack success rate on robust models. Recently, an ensemble of multiple attacks is introduced [6, 27]. In Auto Attack [6], four different diverse attacks including both white-box and black-box attacks are utilized in a specific order, which achieves state-of-the-art attacking performance. Due to its superior performance, Auto Attack is currently one of the most important criteria of network adversarial robustness evaluation.

2.2. Adversarial Defense

Defending adversarial attacks becomes a crucial problem which has attracted increasing attention [5, 8]. The main stream of defence mechanisms lies in the adversarial training and it remains relatively resistant to most existing attacks. Vanilla adversarial training strategy simply takes adversarial examples as training data to form a min-max game during optimization [28]. There exist a large number of variants of adversarial training algorithms, which improve the adversarial robustness performance [35, 36, 45]. Zhang *et al*. introduced friendly adversarial training which adopts early-stopped PGD attack to improve natural generalization of networks [52]. Rice *et al*. explored the importance of early-stopping strategy in adversarial training [35]. TRADES was introduced to achieve better trade-offs between adversarial robustness and natural accuracy [51]. Besides traditional adversarial training, there exist randomized techniques for adversarial robustness [7, 17, 21]. Liu *et al*. proposed to inject random noises before the convolutional layers, which forms a noisy model to defend against adversarial examples [25]. Pinot *et al*. provided theoretical evidence that deterministic classifier can hardly ensure optimal robustness against all potential adversarial attacks and a mixture of classifiers can offer better robustness [32]. Fu *et al*. proposed to utilize random bits for adversarial defense [10]. Although the methods of noise injection can be diverse, how the noise injection influence the natural generalization as well as convergence of networks has not been well explored, which could constrained the robustness.

2.3. Random Projection

Random projection is a classic technique in dimensionality reduction [2]. Through controlling the distribution and dimensionality of random projection matrices, the pairwise distances between any two data points can be preserved after the projection, which is stated in Johnson-Lindenstrauss lemma [41]. Furthermore, the projection is achieved by a simple linear transformation via random projection matrices, whose entities are sampled from a predefined distribution, random projection is both efficient and effective in practice. Due to its effectiveness, some work proposed to incorporate random projection into DNNs [30, 50]. Different from previous work, we propose to treat random projection as a noise injection method, which performs a strong defense scheme against adversarial attacks.

3. Methodology

3.1. Preliminaries

Adversarial Training Given a classifier f with parameters θ which maps the input image $\mathcal{X} \in \mathbb{R}^D$ to the logits $f_\theta(\mathcal{X}) \in \mathbb{R}^C$ where D and C denote the dimension of original image and number of classes respectively, the adversarial example $\mathcal{X}^{adv} = \mathcal{X} + \delta$ is defined as

$$\max_{\mathcal{X}^{adv}} \mathcal{L}(f_\theta(\mathcal{X}^{adv}), y), \text{ s.t. } \|\mathcal{X}^{adv} - \mathcal{X}\|_p \leq \epsilon, \tag{1}$$

where $\mathcal{L}(,)$ denotes the loss function (*e.g.* cross-entropy loss), ϵ denotes the maximum perturbation size and y denotes the ground truth label. In adversarial training strategy, the adversarial examples are generated and fed to the classifier to form a min-max optimization as

$$\min_\theta \max_{\mathcal{X}^{adv}} \mathcal{L}(f_\theta(\mathcal{X}^{adv}), y), \text{ s.t. } \|\mathcal{X}^{adv} - \mathcal{X}\|_p \leq \epsilon. \tag{2}$$

Random Projection The random projection is a linear transformation from D dimensions to D' dimensions via a random matrix $R \in \mathbb{R}^{D \times D'}$ where each entry is drawn from an independent identically distributed (i.i.d.) Gaussian distribution $\mathcal{N}(0, 1)$ and the columns are normalized to have unit lengths. Given data point $x \in \mathbb{R}^D$, the random projected data point $x' \in \mathbb{R}^{D'}$ can be derived as $x' = xR$. In CNNs, we can simply replace the filter parameters with the i.i.d. zero-mean Gaussian weights.

3.2. Random Projection Filters

White-box attacks have full access to network including the parameters and architecture and it is difficult for networks to defend against various white-box attacks since adversarial perturbations can be easily found via gradient ascent. Thus, to prevent attackers from deriving precise gradient on input image, we propose to involve some noises during the network inference. However, the magnitude of noise and the manner of the noise involvement can significantly influence the optimization, which implies that a careful design of noise is necessary to achieve a better trade-offs between adversarial robustness and optimization difficulty.

Motivated by the distance preservation of random projection, we propose to incorporate random projection into CNNs to achieve a better trade-offs. The core idea of random projection mainly lies in the Johnson-Lindenstrauss lemma which states that a projection of data points of high dimension to an appropriate lower dimensional space can preserve the distances among the data points. By definition, given a linear mapping $F : \mathbb{R}^D \rightarrow \mathbb{R}^{D'}$ and a set of data points $\mathbb{X}$ with size of m, for $D' > 8(ln\ m)/\epsilon^2$

$$(1-\epsilon)\|x_i - x_j\|^2 \leq \|F(x_i) - F(x_j)\|^2 \leq (1+\epsilon)\|x_i - x_j\|^2, \tag{3}$$

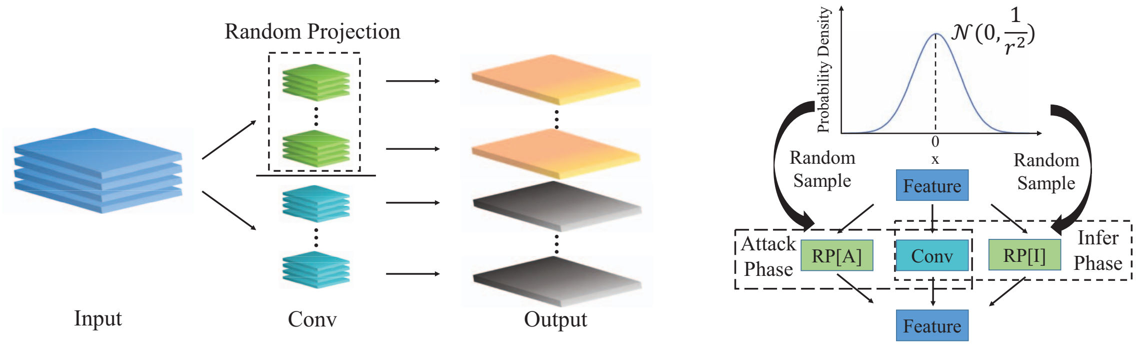
for all x_i, x_j in $\mathbb{X}$. Since convolution is a linear mapping, Johnson-Lindenstrauss lemma holds for CNNs. According to Eq. 3, the dimension of projected space plays an important role in random projection. Intuitively, a higher ratio of random projection in CNNs could make it difficult for white-box attackers to obtain adversarial perturbations, however, it also brings huge noise to network optimization. Thus, to balance these trade-offs, we propose to replace a bunch of the convolutional filters in CNN layers with random projection, as shown in Figure 1 (a). Formally, given the input feature $x \in \mathbb{R}^{n \times n \times d}$ where n and d denote the size and dimension of feature respectively, and a single filter of CNN $F \in \mathbb{R}^{r \times r \times d}$ where r denotes the kernel size, the output z is given by

$$z(p, q) = F * [x]^r_{p,q} = \sum_{i=0}^{r}\sum_{j=0}^{r}\sum_{k=0}^{d} F(i, j, k) \cdot x(p+i, q+j, k), \tag{4}$$

where $[x]^r_{p,q}$ denotes the subarea of x for convolutional operation with row from p to $p + r - 1$ and column from q to $q + r - 1$. For a convolutional layer which contains N filters $F_1, \ldots, F_N$, we divide these filters into two parts. We denote $F_1, \ldots, F_{N_r}$ as the random projection filters with parameters randomly sampled from a zero-mean Gaussian distribution, and denote $F_{N_r+1}, \ldots, F_N$ as the traditional convolutional filters with trainable parameters. The output z can be formulated as

$$\begin{aligned} &z(p, q) = \left[F_{1,\ldots,N_r} * [x]^r_{p,q}, F_{N_r+1,\ldots,N} * [x]^r_{p,q}\right], \\ &\text{where } F_1, \ldots, F_{N_r} \sim \mathcal{N}(0, \sigma^2), \end{aligned} \tag{5}$$

where $\left[,\right]$ denotes the concatenation and σ^2 denotes the variance of random projection filters. With Eq. 5, the trade-offs between adversarial defense capability and network optimization difficulty can be explored via adjusting N_r. The Johnson-Lindenstrauss lemma in CNN layers has been studied in [30]. In this work, under mild assumptions, we further generalize it to the random projection scenario where only N_r output features are derived via random projection while the others via optimized convolutional filters. Since batch normalization layers with affine transformation are widely adopted in CNNs, we assume that the inputs to



(a). Random Projection Filters

(b). Defense with Random Gaussian Matrix

Figure 1. An overview of proposed with Random Projection Filters with defense scheme. Part of the filters in convolutional layers are replaced by random projection, whose weight is randomly sampled from a Gaussian distribution. RP[A] and RP[I] denote the sampled Gaussian matrix of random projection filters during attack and infer phase respectively.

the filters follows Gaussian distribution with mean of β and variance of γ^2 where β and γ are the affine parameters of batch normalization layers. We also assume that the variance of trainable filters $F_{N_{r+1}}, \ldots, F_N$ is same as the one of random projection filters $F_1, \ldots, F_{N_r}$.

Theorem 1. *Let $x, y \in \mathbb{R}^{n \times n \times d}$ be the input to the filters, which follow Gaussian distribution $x, y \sim \mathcal{N}(\beta, \gamma^2)$. Consider we have N filters $F_1, \ldots, F_N \in \mathbb{R}^{r \times r \times d}$, in which $F_1, \ldots, F_{N_r}$ denote the random projection matrices where all the entries are drawn from i.i.d. $\mathcal{N}(0, \frac{1}{r^2})$ while $F_{N_{r+1}}, \ldots, F_N$ denote the trainable parameters of convolutional layer with mean of μ and variance of $\frac{1}{r^2}$ where r denotes the kernel size. We assume that*

$$\max_{i,j} \|[x]_{ij}^r\| \leq R, \quad \max_{i,j} \|[y]_{ij}^r\| \leq R, \quad \max_i \|F_i\| \leq W, \tag{6}$$

and we denote $K = n^2 max\{\frac{C_0^2 R^2}{r^2}, (r^2 d\beta\mu + C_0 W\gamma)^2\}$ and $D = \mu^2\beta^2 n^2 r^4 d^2$. Then the probability that the distance between x, y cannot be preserved after convolutional operation F can be upper bounded as

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{l=1}^{N}\langle F_l * x, F_l * y\rangle - \langle x, y\rangle\right| \geq \epsilon\right) \leq \delta, \ \textit{for } \delta > 0 \textit{ and}$$
$$N_r > \begin{cases} \frac{(D-\epsilon)N + K log\frac{2Cn^2}{\delta}}{D}, & \textit{if } \frac{\epsilon - \frac{N-N_r}{N}D}{K} \leq \frac{(\epsilon - \frac{N-N_r}{N}D)^2}{K^2} \\ \frac{(D-\epsilon)N + NK\sqrt{log\frac{2CNn^2}{\delta}}}{D}, & \textit{otherwise} \end{cases} \tag{7}$$

where C and C_0 are absolute constants.

In Theorem 1, we define the distance between two data points x, y as $\langle x, y\rangle$ and the geometric representation preservation as the scenario where the sum of absolute differences between $\langle F_l * x, F_l * y\rangle$ and $\langle x, y\rangle$ can be bounded by ϵ. The proof mainly utilizes the Bernstein's inequality [41] and the detailed proof is provided in the supplementary material. In Eq. 7, we can see that the probability of breaking geometric representation preservation can be upper bounded by δ if an appropriate N_r is selected for this convolutional layer. It indicates a lower bound of the number of random projection filters. Since our objective is the better trade-offs between network optimization difficulty and defense capability, we propose to reduce the number of random projection filters N_r while meeting the constraint in Eq. 7 so that geometric representation preservation holds and the noises introduced by random projection filters do not damage the convergence and performance of networks. Besides the constants, the lower bound of N_r is dominated by $K = n^2 max\{\frac{C_0^2 R^2}{r^2}, (r^2 d\beta\mu + C_0 W\gamma)^2\}$. In practice, the maximum Euclidean norm of input subareas R can be well-controlled due to batch normalization layers while the weight norm of trainable parameters $F_{N_{r+1}}, \ldots, F_N$ cannot. Thus, to reduce the burden of N_r, we propose to impose a larger weight decay to the $F_{N_{r+1}}, \ldots, F_N$, which minimizes W to relieve the constraint in Eq. 7. Thus, the objective in Eq. 2 can be reformulated as

$$\begin{aligned} &\min_\theta \max_{\mathcal{X}^{adv}} \mathcal{L}(f_\theta(\mathcal{X}^{adv}), y) + \alpha\|F_{N_r+1}, \ldots, F_N\|, \\ &\text{s.t. } \|\mathcal{X}^{adv} - \mathcal{X}\|_p \leq \epsilon, \end{aligned} \tag{8}$$

where α denotes the hyperparameter of weight decay.

3.3. Adversarial Training with Random Projection

Existing white-box attacks can easily discover an aggressive perturbation δ for a fixed network f via gradient ascent, however, it is difficult for the generated adversarial example $x' = x + \delta$ to attack another network f' successfully [40, 43]. Since the parameters of random projection filters $F_1, \ldots, F_{N_r}$ are randomly sampled from $\mathcal{N}(0, \frac{1}{r^2})$, we individually sample parameters for random projection filters during attacking and inference phase, which is denoted as $F_{1:N_r}[A]$ and $F_{1:N_r}[I]$ respectively. Considering the partial derivative of output feature z with respect to the

Algorithm 1 Adversarial Training with Random Projection

```
Input: Network with random projection filters $f_\theta$; Number of random projection filters $N_r$; Weight decay of random projection filters $\alpha$; Perturbation size $\epsilon$; Attack step size $\eta$; Attack iterations $t$; Training set $\{\mathcal{X}, \mathcal{Y}\}$;
while not converge do
    Sample a batch of data $\{x, y\}_{i=1}^{n}$ from $\{\mathcal{X}, \mathcal{Y}\}$;
    for $F$ with random projection filters do
        $F_1, \ldots, F_{N_r} \sim \mathcal{N}(0, \frac{1}{r^2})$
    end for
    Random initialize adversarial perturbation $\delta$;
    for $i \leftarrow 1$ to $t$ do
        $\delta = \delta + \eta \cdot \mathrm{sign}(\nabla_x \mathcal{L}(f_\theta(x^{adv}), y)$;
        Clip $x^{adv}$ = $\mathrm{Clip}_x^{\epsilon}\{x + \delta\}$;
    end for
    for $F$ with random projection filters do
        $F_1, \ldots, F_{N_r} \sim \mathcal{N}(0, \frac{1}{r^2})$;
    end for
    $\theta = \theta - \nabla_\theta \Big( \mathcal{L}(f_\theta(x^{adv}), y) + \alpha \| F_{N_r+1}, \ldots, F_N \| \Big)$;
end while
```

input adversarial feature x^{adv}

$$\frac{\partial z(p, q, 1 : N_r)}{\partial x^{adv}(p+i, q+j, k)} = F_{1:N_r}[A](i, j, k) \neq F_{1:N_r}[I](i, j, k), \tag{9}$$

which indicates that the difference between $F_{1:N_r}[A]$ and $F_{1:N_r}[I]$ can significantly influence the gradient of adversarial examples. $f_{\theta[A]}$ denotes the parameters to be optimized in a network with the random projection filters $F_{1:N_r}[A]$, while $f_{\theta[I]}$ corresponds to the parameters to be optimized in one network with $F_{1:N_r}[I]$. The mix-max optimization in Eq. 2 can be reformulated as

$$\begin{aligned} &\min_{\theta[I]} \max_{\mathcal{X}^{adv}} \mathcal{L}(f_{\theta[A]}(\mathcal{X}^{adv}), y) + \alpha \| F_{N_r+1}, \ldots, F_N \|, \\ &\text{s.t. } \|\mathcal{X}^{adv} - \mathcal{X}\|_p \leq \epsilon, \end{aligned} \tag{10}$$

where adversarial examples have been produced from a network with random projection filters $F_{1:N_r}[A]$ and then the adversarial examples are used to train a network with random projection filters $F_{1:N_r}[I]$. The details of adversarial training with random projection is shown in Algorithm 1.

With the involvement of random projection in convolutional filters and corresponding adversarial training strategy in Algorithm 1, CNNs can perform a strong defense during inference phase, which is illustrated in Figure 1 (b). Considering a white-box attack which has access to the current sampled random projection parameters $F_{1:N_r}[A]$ and generates adversarial example $\mathcal{X}^{adv}$ of $f_{\theta[A]}$ successfully, $F_{1:N_r}[A]$ is re-sampled and becomes $F_{1:N_r}[I]$ during evaluation so that $\mathcal{X}^{adv}$ can hardly be generalized to $f_{\theta[I]}$. Together with the fact that Theorem 1 holds for random Gaussian matrix sampling strategy, RPF achieves better tradeoffs between clean accuracy and adversarial robustness.

4. Experiments

4.1. Experimental Setup

Datasets Following previous work [28, 35], we include multiple datasets in our evaluation, including CIFAR-10/100 and ImageNet. CIFAR-10 and CIFAR-100 datasets [23] have 10 and 100 categories respectively. Each of them contains $60K$ color images with size of 32×32, including $50K$ training images and $10K$ validation images. ImageNet dataset [16] contains $1.2M$ training images and $50k$ testing images with size of 224×224 from 1000 categories.

Models Note that our proposed RPF can be easily applied to any CNN-based models via partially replacing CNN filters with random projection ones. Thus, we evaluate the performance of RPF on several widely compared models in the field of adversarial robustness, including ResNet-18 [23] and WideResnet-34-10 [49] on CIFAR-10/100 as well as ResNet-50 [23] on ImageNet.

Training Strategy We follow the protocol of state-of-the-art adversarial training strategy [35] to setup our experiments on CIFAR-10/100. We train the network for 200 epochs with a batch size of 128 via SGD with momentum of 0.9. The learning rate is set to 0.1 and the weight decay is set to 5×10^{-4}. We use a piecewise decay learning rate scheduler with a decay factor of 0.1 at 100 and 150 epoch. For adversarial example generation, PGD-10 is used with the a maximum perturbation size $\epsilon = 8/255$. The step size of PGD is set to $2/255$. On ImageNet, we train the network for 90 epochs with a batch size of 1024 via SGD with momentum of 0.9. The learning rate is set to 0.02 and the weight decay is set to 1×10^{-4}. We use a cosine learning rate scheduler. For adversarial example generation, PGD-2 is used with the a maximum perturbation size $\epsilon = 4/255$.

Attacks For the adversarial robustness evaluation of proposed RPF, we conduct extensive experiments on various attacks, including Fast Gradient Sign Method (FGSM) [39], Projected Gradient Descent (PGD) [28], CW attack [3], Momentum-based Iterative Fast Gradient Sign Method (MIFGSM) [9], DeepFool [29] and Auto Attack [6]. We follow the standard protocol [22] to setup the attacks. The maximum perturbation size ϵ is set to $8/255$ for FGSM, PGD, MIGFSM, and Auto Attack. The step size is set to $2/255$ for PGD and MIGFSM, and the steps are 20 and 5 for PGD and MIGFSM respectively. For CW attack, the learning rate is set to 0.01 with 1000 steps. For DeepFool, the steps are set to 50 with an overshoot of 0.02. On ImageNet, ϵ is set to $4/255$ with steps of 10 and 50.

Baselines We include extensive baselines for comparison. We compare RPF with some randomize techniques,

Table 1. The comparison with noise injection techniques with ResNet-18 on CIFAR-10 and CIFAR-100.

Dataset	Method	Clean	FGSM	PGD^{20}	CW	MIFGSM	DeepFool	AutoAttack
CIFAR-10	AT [35]	81.84	56.70	52.16	78.46	54.96	0.35	47.69
	Additive [25]	81.24	59.19	57.61	80.84	57.83	73.44	62.25
	Multiplicative	83.16	61.92	59.49	82.80	59.48	78.28	63.78
	RPF(Ours)	**83.79**	**62.71**	**61.27**	**83.60**	**60.72**	**79.43**	**64.38**
CIFAR-100	AT [35]	55.81	31.33	28.71	50.94	30.26	0.79	24.48
	Additive [25]	53.34	31.72	31.13	52.50	31.16	46.76	36.37
	Multiplicative	54.52	34.09	32.58	54.61	32.45	50.90	38.13
	RPF (Ours)	**56.88**	**37.67**	**37.37**	**56.59**	**35.31**	**54.39**	**42.88**

Table 2. Adversarial robustness evaluation of randomized techniques with WideResNet on CIFAR-10.

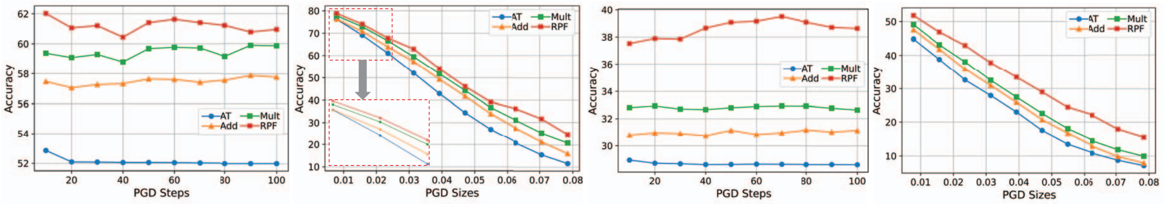
Method	FGSM	PGD^{20}	MIFGSM	AA
AT [35]	60.65	55.06	58.47	52.24
Random Bit [10]	57.95	53.96	56.32	53.30
Additive [25]	62.36	58.47	60.58	60.55
Multiplicative	62.01	57.48	59.79	57.99
RPF (Ours)	**63.95**	**63.71**	**60.77**	**68.71**

such as additive noise [25] and random bits [10]. We also include another strong baseline for comparison which replaces the additive noise with the multiplicative noise. In addition, some other defense techniques are also involved in our comparison, including RobustWRN [18], AWP [44], SAT [45], LLR [34], and RobNet [14].

4.2. Results on CIFAR

To demonstrate the effectiveness of proposed RPF, we first perform six different attacks. Besides the deterministic classifier with adversarial training denoted as AT, we include the additive noise injection. We follow the setting of [25] to conduct noise injection where some sampled noises are added to the input of convolution layers. Furthermore, we also construct another stronger baseline, namely multiplicative noise injection, which simply fuses the feature maps via multiplying noises. Additive noises are sampled from a standard Gaussian distribution $\mathcal{N}(0,1)$ while multiplicative noises are sampled from $\mathcal{N}(1,1)$. Although these baselines are simple, they can achieve satisfactory adversarial robustness in our defense scheme. The comparison results are shown in Table 1. All the baselines are adversarially trained with the same setting. On CIFAR-10, the additive noise injection can achieve 57.61% robust accuracy under PGD^{20} attack and 59.19% under FGSM attack. Compared with deterministic AT baseline, additive noise injection improves the baseline by 5.45% under PGD^{20} attack and 2.49% under FGSM attack, which demonstrates that the randomized techniques can improve the adversarial robustness against current popular white-box attacks. Besides additive noise injection, the multiplicative noise injection baseline also shows superiority to AT baseline. Comparing additive and multiplicative noise injections, the multiplicative one has better performance. For example, multiplicative noise achieves 82.80% robust accuracy under CW attack with a gap of 1.96% an 63.78% under Auto Attack with a gap of 1.53%, which indicates that the noise injected to CNNs as well as the method of injection play important roles in the adversarial robustness. The natural accuracy decrement of additive noise injection also implies that there exists a trade-offs between natural generalization of network and adversarial robustness if randomized techniques are utilized. Thus, RPF is introduced to tackle this problem via the involvement of random projection. With the assistance of the geometric representation preservation property, our algorithm can achieve better trade-offs than these baselines. To illustrate, on ResNet-18, our proposed RPF achieves 83.79% natural accuracy, 61.27% robust accuracy under PGD^{20} attack, 79.43% under DeepFool attack, and 64.38% under Auto Attack, with a obvious gap between RPF and all the baselines. Under 6 different attacks, our proposed RPF achieves the best performance in all the scenarios as well as the highest clean accuracy. This superiority can also be generalized to CIFAR-100. RPF improves the robust accuracy of AT baseline by 18.40% under Auto Attack, 8.66% under PGD^{20} attack, and 5.05% under MIFGSM attack. Furthermore, RPF achieves a clean accuracy of 56.88%, which improves all the noise injection techniques by a considerable gap. On the contrary, the superiority of both additive and multiplicative noises to AT baseline becomes much slighter, which highlights the necessity of proposed RPF as a more advanced noise injection techniques for adversarial robustness.

We also provide adversarial robustness evaluation with WideResNet-34-10 on CIFAR-10, and compare the results with other randomized baselines. The comparison is presented in Table 2. Similar to the results on ResNet-18, our proposed RPF achieves strong adversarial robustness. Under powerful Auto Attack, RPF remains a accuracy of 68.71% with a gap of 16.47 to deterministic AT classifier and a gap of 8.16% to the additive noise injection. Compared to other randomized techniques, such as random bit, RPF also show clear advantage, with a robust accuracy of 63.71% under PGD^{20} compared to 53.93% in random bits. The extensive experimental results under various attacks



(a) CIFAR-10 PGD Steps (b) CIFAR-10 PGD Sizes (c) CIFAR-100 PGD Steps (d) CIFAR-100 PGD Sizes

Figure 2. The evaluation of stronger PGD attacks with ResNet-18 on CIFAR-10 and CIFAR-100.

Table 3. Comparison with SOTA defense algorithms on CIFAR-10 and ImageNet.

Method	CIFAR-10		ImageNet	
	PGD^{20}	AA	PGD^{10}	PGD^{50}
Overfit [35]	55.06	52.24	39.85	39.19
RobustWRN [18]	59.13	52.48	31.14	-
AWP [44]	58.14	54.04	-	-
RobNet [14]	52.74	-	37.16	37.15
Random Bit [10]	53.96	53.30	42.88	42.72
SAT [45]	56.01	51.83	-	42.30
LLR [34]	54.24	-	-	47.00
RPF (Ours)	**63.71**	**68.71**	**56.56**	**55.41**

with multiple models on different datasets show strong empirical evidence that our proposed RPF can achieve superior adversarial robustness.

4.3. Evaluate with Stronger Attacks

Besides the standard evaluation of adversarial robustness under various attacks in Table 1, we also provide the performance of AT baseline, noise injection baselines, and our algorithm under stronger attacks. The evaluation is conducted on CIFAR-10 and CIFAR-100, as shown in Figure 2. We mainly consider two scenarios, including the PGD attacks with more steps and the PGD attacks with larger maximum perturbation size ϵ. Specifically, we consider the attacking scenario of PGD^{10}, . . . , PGD^{100}, and $\epsilon \in [2/255, 20/255]$. The randomized techniques are insensitive to the increasing PGD steps, as illustrated in Figure 2 (a) and (c). Different from the deterministic AT classifier whose robust accuracy is inversely proportional to the number of steps, all the noise injection based method still have chance to achieve a relatively higher robust accuracy even under PGD^{100}. Among all the techniques, RPF achieves the best performance under different PGD steps. Taking PGD size into consideration, all the robust methods have a large drop with the increment of perturbation size since the search space of adversarial perturbations becomes much larger. The results are illustrated in Figure 2 (b) and (d). Compared with baselines, RPF shows more robust performance under larger perturbation sizes. On CIFAR-10, RPF achieves 78.73% accuracy under PGD with $\epsilon = 2/255$ and 24.53% under PGD with $\epsilon = 20/255$, where the drop is 54.31%. Additive noise injection has a drop of $60.29\% (76.37\% \rightarrow 16.08\%)$ and multiplicative noise injection has a drop of $56.92\% (77.80\% \rightarrow 20.88\%)$. Similarly, on CIFAR-100, the drop of RPF is 36.22%, 39.63% for additive noise injection, and 39.16% for multiplicative noise injection, where RPF has a much smaller accuracy drop. Thus, among all the noise injection techniques, RPF performs better resistance against stronger PGD attacks. In all the scenarios including PGD steps and sizes, RPF consistently achieves the best robust accuracy, which highlights the superior defense capability of RPF.

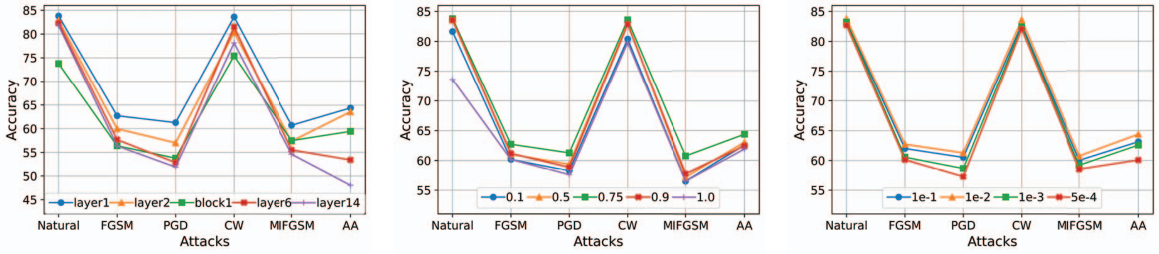
4.4. State-of-the-art Comparison

We further provide the comparison with state-of-the-art defense techniques to demonstrate the effectiveness of proposed RPF. We consider two popular benchmarks which are widely compared, including WideResNet-34-10 (WRN-34-10) on CIFAR-10 and ResNet-50 on ImageNet. For evaluation, we select PGD^{20} and Auto Attack on CIFAR. On ImageNet, we report the robust accuracy under PGD^{10} and PGD^{50} attacks. For baselines Overfit [35] and Random bit [10], we reproduce the results with the official implementation. For the rest results, we cite them from the original paper. Compared with these baselines, RPF performs much stronger defense. RPF achieves 5.57% higher than AWP with PGD^{20} and 15.41% higher than Random Bits with Auto Attack on CIFAR-10. Similarly, RPF achieves 19.4% higher than RobNet with PGD^{10} and 8.41% higher than LLR with PGD^{50} on ImageNet. We mainly attribute the success of RPF to the theoretical-guided design of randomized techniques. Note that RPF can be easily integrated into other state-of-the-art defense techniques to further improve the performance since RPF is orthogonal to other baselines and there is no extra parameters involved.

4.5. Ablation Study

During the setup of random projection filters, multiple components could influence the performance, including the location of random projection filters in the network, the ratio of random projection, and the weight decay of the other convolution filters. We further provide more empirical evidence to verify the observations in Theorem 1.

Random Projection Filters Location We first replace a specific convolution layer with the one with random projection filters in different locations of network. Taking ResNet-18 as an example, we propose to apply random projection



(a) Location of RPF

(b) Ratio of RPF

(c) Weight Decay of Conv

Figure 3. Ablation studies of Random Projection Filters on location, ratio, and weight decay.

in different layers or entire block and the number of random projection filters is set to 48. The natural accuracy and adversarial robustness under different scenarios are shown in Figure 3 (a). Considering the replacement of an entire block, the injected noise could be redundant and overwhelm the natural generalization, which makes both natural and robust accuracy drop to some extent. RPF on the first layer achieves a natural accuracy of 83.79% and robust accuracy of 61.27% under PGD attack while the natural accuracy drops to 73.76% and robust accuracy to 53.80% with RPF on the first block. Thus, we propose to apply random projection to a specific layer in the network. There exist a clear tendency that the robust accuracy decreases if we deploy random projection filters in the deeper layers. For example, the robust accuracy is 63.57% under Auto Attack with RPF on the 2^{nd} layer while 53.39% on the 6^{th} layer and 48.06% on the 14^{th} layer. According to Eq. 7, the geometric representation preservation holds if the number of random projection filters N_r is large than the term which is proportional to the total number of filters N. When it goes deeper in ResNet-18, N keeps increasing, which requires larger N_r to guarantee the bound in the deeper layers where the redundant random projection filters could hurt the trade-offs between robustness and natural generalization. Thus, we apply random projection filters to the first layer in our experiments to achieve better trade-offs.

Ratio of Random Projection Filters We then explore how the ratio of random projection filters in the first layer of ResNet-18 influence the adversarial robustness. The results are shown in Figure 3 (b). According to Eq. 7, the number of random projection filters N_r is required to be sufficient, however, directly setting $N_r \approx N$ could involves redundant noise to the network. For illustration, applying RPF with a ratio of 0.75 can achieve the natural accuracy of 83.79% and robust accuracy of 61.27% under PGD attack. On the contrary, the natural accuracy becomes 73.60 with a RPF ratio of 1.0 due to the redundant random projection filters, and the robust accuracy becomes 58.26% under PGD attack with a RPF ratio of 0.1 due to the insufficient random projection filters. Thus, the empirical observations of the RPF ratio is consistent with the analysis of Theorem 1.

Weight Norm Study According to Eq. 7, the requirement of N_r can be further relieved via the reduction of weight norm of convolution filters of that layer besides the random projection, which motivates us to adjust the weight decay of these convolution filters via α in Eq. 8. We further provide empirical evidence that the weight norm could play an important role in the our defense scheme through setting different weight decays for the trainable parameters in the first layer of ResNet-18, as shown in Figure 3 (c). The traditional weight decay is set to 5×10^{-4} for ResNet-18 on CIFAR-10, however, it cannot achieve satisfactory performance, with 60.04% robust accuracy under Auto Attack. On the contrary, the variants with larger weight decays, such as 1×10^{-2} or 1×10^{-1}, can achieve 64.38% and 63.12% respectively, which empirically verify the effectiveness of Eq. 8 and the correctness of Theorem 1.

5. Conclusion

In this paper, we propose to utilize random projection as the noise injection to perform a randomized defense technique against adversarial examples. Through the generalization of Johnson-Lindenstrauss lemma to the scenario where partial convolution filters can replaced by random projection, we theoretically show the correlations between weight norm, the number of random projection filters, and the total number of filters. Based on these observations, we introduce Random Projection Filters as a strong defense scheme. Through sufficient evaluation with various models and datasets, we present the superiority of proposed algorithm to other baselines.

Acknowledgments

This work was supported in part by the Australian Research Council under Project DP210101859 and the University of Sydney Research Accelerator (SOAR) Prize. The authors acknowledge the use of the National Computational Infrastructure (NCI) which is supported by the Australian Government, and accessed through the NCI Adapter Scheme and Sydney Informatics Hub HPC Allocation Scheme. The AI training platform supporting this work were provided by High-Flyer AI. (Hangzhou High-Flyer AI Fundamental Research Co., Ltd.)

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, page 484–501, Berlin, Heidelberg, 2020. Springer-Verlag. 1, 2

[2] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, 2001. 2

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 2, 5

[4] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018. 2

[5] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019. 2

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *CoRR*, abs/2003.01690, 2020. 1, 2, 5

[7] Minjing Dong, Xinghao Chen, Yunhe Wang, and Chang Xu. Random normalization aggregation for adversarial defense. In *Advances in Neural Information Processing Systems*. 2

[8] Minjing Dong, Yanxi Li, Yunhe Wang, and Chang Xu. Adversarially robust neural architectures. *arXiv preprint arXiv:2009.00902*, 2020. 2

[9] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 2, 5

[10] Yonggan Fu, Qixuan Yu, Meng Li, Vikas Chandra, and Yingyan Lin. Double-win quant: Aggressively winning robustness of quantized deep neural networks via random precision training and inference. In *International Conference on Machine Learning*, pages 3492–3504. PMLR, 2021. 2, 6, 7

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014. 1

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[13] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pages 2484–2493. PMLR, 2019. 2

[14] Minghao Guo, Yuzhe Yang, Rui Xu, and Ziwei Liu. When NAS meets robustness: In search of robust architectures against adversarial attacks. *CoRR*, abs/1911.10695, 2019. 6, 7

[15] Yiwen Guo, Ziang Yan, and Changshui Zhang. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE, June 2016. 1, 5

[17] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 588–597, 2019. 2

[18] Hanxun Huang, Yisen Wang, Sarah Monazam Erfani, Quanquan Gu, James Bailey, and Xingjun Ma. Exploring architectural ingredients of adversarially robust deep neural networks. *CoRR*, abs/2110.03825, 2021. 6, 7

[19] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2137–2146. PMLR, 10–15 Jul 2018. 1

[20] Nathan Inkawhich, Kevin Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. *Advances in Neural Information Processing Systems*, 33:20791–20801, 2020. 2

[21] Ahmadreza Jeddi, Mohammad Javad Shafiee, Michelle Karg, Christian Scharfenberger, and Alexander Wong. Learn2perturb: an end-to-end feature perturbation learning to improve adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1241–1250, 2020. 2

[22] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020. 5

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[24] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018. 2

[25] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018. 1, 2, 6

[26] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016. 2

[27] Ye Liu, Yaya Cheng, Lianli Gao, Xianglong Liu, Qilong Zhang, and Jingkuan Song. Practical evaluation of adversarial robustness via adaptive auto attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15105–15114, 2022. 2

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2, 5

[29] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 5

[30] Ido Nachum, Jan Hazla, Michael Gastpar, and Anatoly Khina. A johnson-lindenstrauss framework for randomly initialized CNNs. In *International Conference on Learning Representations*, 2022. 3

[31] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016. 2

[32] Rafael Pinot, Raphael Ettedgui, Geovani Rizk, Yann Chevaleyre, and Jamal Atif. Randomization matters how to defend against strong adversarial attacks. In *International Conference on Machine Learning*, pages 7717–7727. PMLR, 2020. 1, 2

[33] Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[34] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. *Advances in Neural Information Processing Systems*, 32, 2019. 6, 7

[35] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020. 2, 5, 6, 7

[36] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019. 2

[37] Xuxiang Sun, Gong Cheng, Hongda Li, Lei Pei, and Junwei Han. Exploring effective data for surrogate training towards black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15355–15364, 2022. 2

[38] Xuxiang Sun, Gong Cheng, Lei Pei, and Junwei Han. Query-efficient decision-based attack via sampling distribution reshaping. *Pattern Recognition*, 129:108728, 2022. 2

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1, 2, 5

[40] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017. 4

[41] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018. 2, 3, 4

[42] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1924–1933, 2021. 2

[43] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1924–1933, June 2021. 4

[44] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 6, 7

[45] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020. 2, 6, 7

[46] Xingyi Yang, Zhou Daquan, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1

[47] Xingyi Yang, Jingwen Ye, and Xinchao Wang. Factorizing knowledge in neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 73–91. Springer, 2022. 1

[48] Yunrui Yu, Xitong Gao, and Cheng-Zhong Xu. Lafeat: Piercing through adversarial defenses with latent features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5735–5745, 2021. 2

[49] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. 5

[50] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019. 1, 3

[51] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019. 1, 2

[52] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pages 11278–11287. PMLR, 2020. 1, 2