Adversarial Robustness via Random Projection Filters: Supplementary Material

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

1. Proof of Theorem 1

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 $F_1, \dots, F_N \in \mathbb{R}^{r\times r\times d}$, in which $F_1, \dots, F_{N_r}$ denote the random projection matrices where all the entries are drawn from i.i.d. $\mathcal{N}(0, \frac{1}{r^2})$ while $F_{N_{r+1}}, \dots, F_N$ 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*

$$\max_{i,j} \|[x]^r_{ij}\| \le R, \quad \max_{i,j} \|[y]^r_{ij}\| \le R, \quad \max_i \|F_i\| \le W, \tag{1}$$

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

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

where C and C_0 are absolute constants.

Proof. We consider a single filter in convolution layer $F \in \mathbb{R}^{r\times r\times d}$ with mean of μ and variance of $\sigma^2 = \frac{1}{r^2}$ and the input $x, y \sim \mathcal{N}(\beta, \gamma^2)$. For simplicity. We denote $k = r \times r \times d$ and $\mathbb{Z}_n$ as the set of $\{0, \dots, n-1\}$. We first prove the following simple results: Let $u, v \in \mathbb{R}^{r\times r\times d}$ and $Z_1 = u^T F, Z_2 = v^T F$, then we have

$$\begin{aligned}
&\mathbb{E}[FF^T] = cov(F) + E[F]E[F]^T = \sigma^2 I + \mu^2, \\
&\mathbb{E}[Z_1 \cdot Z_2] = u^T \mathbb{E}[FF^T]v = \mu^2 \cdot \sum u \cdot \sum v + \sigma^2\langle u, v\rangle,
\end{aligned} \tag{3}$$

where I denotes identity matrix. Now we replace u and v with $[x]^r_{i,j}$ and $[y]^r_{i,j}$ respectively. Given the fact that $\langle x, y\rangle = \frac{1}{r^2}\sum_{i,j\in\mathbb{Z}_n}\langle [x]^r_{i,j}, [y]^r_{i,j}\rangle$, the expectation of the dot product of two filter output can be written as

$$\begin{aligned}
\mathbb{E}[\langle F * x, F * y\rangle] &= \sum_{i,j\in\mathbb{Z}_n} \mathbb{E}[\langle F, [x]^r_{i,j}\rangle \cdot \langle F, [y]^r_{i,j}\rangle] \\
&= \sum_{i,j\in\mathbb{Z}_n} \mu^2 \sum^k [x]^r_{i,j} \cdot \sum^k [y]^r_{i,j} + \sigma^2\langle [x]^r_{i,j}, [y]^r_{i,j}\rangle \\
&= \langle x, y\rangle + \sum_{i,j\in\mathbb{Z}_n} \mu^2 \cdot k^2 \cdot \beta^2
\end{aligned} \tag{4}$$

*Corresponding author.

Similarly, we consider a single random projection filter $F \in \mathbb{R}^{r\times r\times d}$ with zero mean and variance of $\frac{1}{r^2}$.

$$\mathbb{E}[\langle F * x, F * y\rangle] = \sum_{i,j\in\mathbb{Z}_n} \mathbb{E}[\langle F, [x]^r_{i,j}\rangle \cdot \langle F, [y]^r_{i,j}\rangle] = \sum_{i,j\in\mathbb{Z}_n} \frac{1}{r^2}\langle [x]^r_{i,j}, [y]^r_{i,j}\rangle = \langle x, y\rangle \tag{5}$$

For simplicity, we denote $X_{ijl} = \langle F_l, [x]^r_{ij}\rangle$ and $Y_{ijl} = \langle F_l, [y]^r_{ij}\rangle$. Now we consider all the filters including random projection filters $F_1, \ldots, F_{N_r}$ and convolutional filters $F_{N_{r+1}}, \ldots, F_N$. The probability that the absolute difference between the inputs and outputs is large than ϵ can be derived as

$$\begin{aligned}
&\mathbb{P}\left(\left|\frac{1}{N}\sum_{l=1}^{N}\langle F_l * x, F_l * y\rangle - \langle x, y\rangle\right| \geq \epsilon\right)\\
=&\mathbb{P}\left(\left|\frac{1}{N}\left(\sum_{l=1}^{N}(\langle F_l * x, F_l * y\rangle - \mathbb{E}[\langle F_l * x, F_l * y\rangle]) + (\sum_{l=1}^{N-N_r}\sum_{i,j\in\mathbb{Z}_n}\mu_l^2 k^2\beta^2)\right)\right| \geq \epsilon\right)\\
\leq&\mathbb{P}\left(\left|\frac{1}{N}\sum_{l=1}^{N}(\langle F_l * x, F_l * y\rangle - \mathbb{E}[\langle F_l * x, F_l * y\rangle])\right| + \left|\frac{1}{N}\sum_{l=1}^{N-N_r}\sum_{i,j\in\mathbb{Z}_n}\mu_l^2 k^2\beta^2\right| \geq \epsilon\right)\\
=&\mathbb{P}\left(\left|\frac{1}{N}\sum_{l\in[N];i,j\in\mathbb{Z}_n}\langle X_{ijl}, Y_{ijl}\rangle - \mathbb{E}[\langle X_{ijl}, Y_{ijl}\rangle]\right| \geq \epsilon - \frac{N-N_r}{N}\mu^2\beta^2 n^2 k^2\right)\\
\leq&\mathbb{P}\left(\frac{1}{N}\sum_{i,j\in\mathbb{Z}_n}\left|\sum_{l\in[N]}\langle X_{ijl}, Y_{ijl}\rangle - \mathbb{E}[\langle X_{ijl}, Y_{ijl}\rangle]\right| \geq \epsilon - \frac{N-N_r}{N}\mu^2\beta^2 n^2 k^2\right)\\
\leq&\mathbb{P}\left(\frac{n^2}{N}\max_{i,j\in\mathbb{Z}_n}\left|\sum_{l\in[N]}\langle X_{ijl}, Y_{ijl}\rangle - \mathbb{E}[\langle X_{ijl}, Y_{ijl}\rangle]\right| \geq \epsilon - \frac{N-N_r}{N}\mu^2\beta^2 n^2 k^2\right)\\
\leq&\sum_{i,j\in\mathbb{Z}_n}\mathbb{P}\left(\frac{n^2}{N}\left|\sum_{l\in[N]}\langle X_{ijl}, Y_{ijl}\rangle - \mathbb{E}[\langle X_{ijl}, Y_{ijl}\rangle]\right| \geq \epsilon - \frac{N-N_r}{N}\mu^2\beta^2 n^2 k^2\right)
\end{aligned} \tag{6}$$

For the convolutional filters, $X_{ijl} = \langle F_l, [x]^r_{ij}\rangle$ and $Y_{ijl} = \langle F_l, [y]^r_{ij}\rangle$ are linear combination of i.i.d. Gaussian RVs since $x, y \sim \mathcal{N}(\beta, \gamma^2)$. Thus, X_{ijl} and Y_{ijl} are sub-Gaussian RVs with mean of $\beta k\mu$ and variance of $\gamma^2\|F_l\|^2$. The sub-gaussian norm of $\langle F_l, [x]^r_{ij}\rangle$ can be computed as

$$\left\|\langle F_l, [x]^r_{ij}\rangle\right\|_{\psi_2} = \left\|X_{ijl}\right\|_{\psi_2} = \left\|\beta k\mu + \gamma^2\|F_l\|^2 z\right\|_{\psi_2} \leq \left\|\beta k\mu\right\|_{\psi_2} + \left\|\gamma\|F_l\| z\right\|_{\psi_2} \leq k\beta\mu + C_0 W\gamma \tag{7}$$

and $\|\langle F_l, [y]^r_{ij}\rangle\|_{\psi_2} = \|Y_{ijl}\|_{\psi_2} \leq k\beta\mu + C_0 W\gamma$ where C_0 denotes an absolute constant. According to the product of sub-Gaussians property and centering property [8], we have $\|XY\|_{\psi_1} \leq \|X\|_{\psi_2}\|Y\|_{\psi_2}$ and $\|X - \mathbb{E}[X]\|_{\psi_1} \leq C\|X\|_{\psi_1}$. Thus, we have

$$\|X_{ijl}Y_{ijl} - \mathbb{E}[X_{ijl}Y_{ijl}]\|_{\psi_1} \leq (k\beta\mu + C_0 W\gamma)^2. \tag{8}$$

Similarly, for the random projection filters, we have $\|\langle F_l, [x]^r_{ij}\rangle\|_{\psi_2} = \|X_{ijl}\|_{\psi_2} \leq \frac{C_0 R}{r}$ and $\|\langle F_l, [y]^r_{ij}\rangle\|_{\psi_2} = \|Y_{ijl}\|_{\psi_2} \leq \frac{C_0 R}{r}$. According to the product of sub-Gaussians property and centering property, we have

$$\|X_{ijl}Y_{ijl} - \mathbb{E}[X_{ijl}Y_{ijl}]\|_{\psi_1} \leq C_0^2\nu^2 R^2, \tag{9}$$

According to Bernstein's inequality for sub-exponentials, let $X_1, \ldots, X_N$ be independent zero-mean sub-exponential RVs. Then, for all $t \geq 0$

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{i=1}^{N}X_i\right| \geq t\right) \leq 2exp\Big\{-min\Big\{\frac{t^2}{K^2}, \frac{t}{K}\Big\}\cdot c\cdot N\Big\}, \tag{10}$$

where $K = \max_i \|X_i\|_{\psi_1}$ and $c > 0$ is an absolute constant.

Together with results in Eq. 8 and Eq. 9, the probability in Eq. 6 can be bounded as

$$\begin{aligned}
&\sum_{i,j\in\mathbb{Z}_n} \mathbb{P}\left(\frac{n^2}{N}\left|\sum_{l\in[N]}\langle X_{ijl}, Y_{ijl}\rangle - \mathbb{E}[\langle X_{ijl}, Y_{ijl}\rangle]\right| \geq \epsilon - \frac{N-N_r}{N}\mu^2\beta^2 n^2 r^4 d^2\right) \\
&\leq 2n^2 exp\Big\{-min\Big\{\frac{(\epsilon - \frac{N-N_r}{N}\mu^2\beta^2 n^2 r^4 d^2)^2}{n^4 max\{C_0^4\nu^4 R^4, (k\beta\mu + C_0 W\gamma)^4\}}, \frac{\epsilon - \frac{N-N_r}{N}\mu^2\beta^2 n^2 r^4 d^2}{n^2 max\{C_0^2\nu^2 R^2, (k\beta\mu + C_0 W\gamma)^2\}}\Big\}\cdot c\cdot N\Big\}
\end{aligned} \tag{11}$$

We denote $K = n^2 max\{C_0^2\nu^2 R^2, (k\beta\mu + C_0 W\gamma)^2\}$ and $D = \mu^2\beta^2 n^2 r^4 d^2$. If $\frac{\epsilon - \frac{N-N_r}{N}D}{K} \leq \frac{(\epsilon - \frac{N-N_r}{N}D)^2}{K^2}$, we have

$$\begin{aligned}
&\delta > 2cn^2 exp\Big\{-\frac{\epsilon N - D(N-N_r)}{K}\Big\} \\
&log\frac{\delta}{2cn^2} > -\frac{(\epsilon - D)N + DN_r}{K} \\
&Klog\frac{2cn^2}{\delta} < (\epsilon - D)N + DN_r \\
&N_r > \frac{(D-\epsilon)N + Klog\frac{2cn^2}{\delta}}{D}
\end{aligned} \tag{12}$$

Similarly, if $\frac{\epsilon - \frac{N-N_r}{N}D}{K} > \frac{(\epsilon - \frac{N-N_r}{N}D)^2}{K^2}$, we have

$$N_r > \frac{(D-\epsilon)N + NK\sqrt{log\frac{2cNn^2}{\delta}}}{D} \tag{13}$$

□

2. Multiple Runs

We provide the results of multiple runs of proposed random projection filters as well as additive and multiplicative noise injection with ResNet-18 on CIFAR-10. Our proposed RPF consistently achieves the best performance.

Table 1. The evaluation results of 5 runs.

Method	Clean	FGSM	PGD^{20}	CW	MIFGSM	DeepFool	AutoAttack
Add [5]	81.09	59.51	57.46	80.83	57.64	73.56	62.23
	81.02	59.24	57.84	80.77	57.49	73.61	62.02
	81.49	59.88	57.25	80.90	57.83	73.65	62.10
	80.94	59.23	57.83	81.36	57.86	73.57	62.11
	81.24	59.19	57.61	80.84	57.83	73.44	62.25
Add [5] Avg	81.16	59.41	57.60	80.94	57.73	73.57	62.14
Multi	82.91	61.89	59.77	82.70	59.96	78.49	63.96
	82.76	61.89	59.43	82.77	59.54	78.98	64.03
	83.08	61.77	59.05	82.73	59.36	78.52	63.94
	82.74	61.98	59.34	82.27	59.37	78.70	64.04
	83.16	61.92	59.49	82.80	59.48	78.28	63.78
Multi Avg	82.93	61.89	59.42	82.65	59.54	78.59	63.95
RPF	83.75	62.87	60.75	83.62	60.59	78.96	64.71
	83.48	63.19	60.88	83.63	60.39	79.74	64.72
	83.73	62.87	60.89	83.62	61.94	79.71	64.29
	83.80	61.95	62.12	83.34	61.57	79.31	65.06
	83.79	62.71	61.27	83.60	60.72	79.43	64.38
RPF(Ours) Avg	**83.72**	**62.72**	**61.19**	**83.56**	**61.04**	**79.43**	**64.63**

Table 2. Evaluation of black-box attacks.

	C-10		C-100			C-10		C-100	
Attack	AT	RPF	AT	RPF	Attack	AT	RPF	AT	RPF
Square	53.64	**76.56**	29.57	**48.21**	Pixle	8.21	**44.84**	1.16	**23.39**

3. Evaluation of Black-box Attacks

We evaluate RPF under black-box attacks Square [1] and Pixle [6] with ResNet-18 on CIFAR-10 and CIFAR-100. Query number is set to 5000 in Square and the maximum patch size is 10×10 in Pixle. The advantage of RPF over AT can be found in Table 2 where RPF achieves better robust accuracy in all the scenarios.

4. Evaluation on More Models, Norms, and Defense Techniques.

We apply RPF on different models including densenet121, squeezenet, and vgg. We also include evaluation on different normalizations including instance norm and layer norm [2, 7]. Furthermore, we include MART+RPF in our evaluation [9]. Our proposed RPF shows consistent improvements in all the scenarios, as shown in Table 3.

Table 3. Results with ResNet-18 on CIFAR-10.

Setting	Method	Clean	FGSM	PGD	MIFGSM	AA
DenseNet	AT	82.94	59.36	55.32	57.67	51.83
	RPF	**85.19**	**60.90**	**57.00**	**58.92**	**59.91**
SqueezeNet	AT	76.71	51.95	47.29	49.91	42.06
	RPF	**82.66**	**64.59**	**62.99**	**60.83**	**69.06**
Vgg16 BN	AT	79.30	53.87	48.40	51.62	44.17
	RPF	**82.41**	**61.92**	**61.09**	**61.40**	**64.41**
IN	AT	81.05	52.13	42.96	48.50	39.82
	RPF	**84.00**	**56.67**	**49.46**	**52.36**	**52.46**
LN	AT	78.07	52.91	45.57	50.30	41.35
	RPF	**82.38**	**57.42**	**50.73**	**53.80**	**54.12**
Defense	MART	77.35	56.04	52.22	54.65	45.55
	RPF	**82.11**	**62.65**	**60.40**	**60.97**	**64.46**

5. Comparisons with Noise Injection Techniques

Different from [3, 4] which utilize additive noises, RPF replaces partial filters with random projection to form concatenate noise. Following the same setting in [4], we apply RPF on ResNet-20/32/44/56. RPF performs better than PNI [3] and Learn2Perturb [4] with relatively large margins, as shown in Table 4.

Table 4. Comparison with other noise injection techniques.

Method	R20		R32		R44		R56	
	FGSM	PGD	FGSM	PGD	FGSM	PGD	FGSM	PGD
PNI	54.40	45.90	51.50	43.50	55.80	48.50	53.90	46.30
Learn2Perturb	58.41	51.13	59.94	54.62	61.32	54.62	61.53	54.62
RPF	**63.27**	**60.94**	**62.52**	**60.78**	**63.39**	**62.47**	**62.30**	**60.97**

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. 4

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] 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. 4

[4] 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. 4

[5] 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. 3

[6] Jary Pomponi, Simone Scardapane, and Aurelio Uncini. Pixle: a fast and effective black-box attack based on rearranging pixels. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2022. 4

[7] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

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

[9] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020. 4