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 \( F_1, \ldots, F_N \in \mathbb{R}^{r \times r \times d} \), in which \( F_1, \ldots, F_N \) denote the random projection matrices where all the entries are drawn from i.i.d. \( \mathcal{N}(0, \frac{1}{T}) \) while \( F_{N+1}, \ldots, F_N \) denote the trainable parameters of convolutional layer with mean of \( \mu \) and variance of \( \frac{1}{T^2} \) where \( T \) denotes the kernel size. We assume that

\[
\max_{i,j} \| [x]_{i,j}^r \| \leq R, \quad \max_{i,j} \| [y]_{i,j}^r \| \leq R, \quad \max_{i} \| F_i \| \leq W, \quad (1)
\]

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

\[
\mathbb{P}\left( \left| \frac{1}{N} \sum_{i=1}^{N} \langle F_i \ast x, F_i \ast y \rangle - \langle x, y \rangle \right| \geq \epsilon \right) \leq \delta, \quad \text{for} \quad \delta > 0 \quad \text{and} \quad N_r > \begin{cases} \frac{(D-\epsilon)N+K\log\frac{2C_n}{\epsilon}}{D}, & \text{if} \quad \frac{\epsilon - \frac{N}{N_r} D}{\frac{N}{N_r}} \leq \frac{(\epsilon - \frac{N}{N_r} D)^2}{K^2} \\ \frac{(D-\epsilon)N+KD\log\frac{2C_n}{\epsilon}}{D}, & \text{otherwise} \end{cases} \quad (2)
\]

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 \( \mu \) and variance of \( \sigma^2 = \frac{1}{T^2} \) 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 a set of \( \{0, \ldots, n-1\} \). 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

\[
\mathbb{E}[FF^T] = \text{cov}(F) + E[F]E[F]^T = \sigma^2 I + \mu^2, \\
\mathbb{E}[Z_1 \cdot Z_2] = u^T \mathbb{E}[FF^T] v = \mu^2 \cdot \sum u \cdot \sum v + \sigma^2 \langle u, v \rangle, \quad (3)
\]

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

\[
\mathbb{E}[\langle F \ast x, F \ast y \rangle] = \sum_{i,j \in \mathbb{Z}_n} \mathbb{E}[\langle F, [x]_{i,j}^r \rangle \cdot \langle F, [y]_{i,j}^r \rangle] \\
= \sum_{i,j \in \mathbb{Z}_n} \mu^2 \sum_{i,j \in \mathbb{Z}_n} [x]_{i,j}^r \cdot [y]_{i,j}^r + \sigma^2 \langle [x]_{i,j}^r, [y]_{i,j}^r \rangle \\
= \langle x, y \rangle + \sum_{i,j \in \mathbb{Z}_n} \mu^2 \cdot [k^2] \cdot \beta^2 \quad (4)
\]

*Corresponding author.*
According to Bernstein’s inequality for sub-exponentials, let
\[
\mathbb{E}[\{F \ast x, F \ast y\}] = \sum_{i,j \in \mathbb{Z}_n} \mathbb{E}[\{F, [x]_{i,j}^r\} \cdot \{F, [y]_{i,j}^r\}] = \sum_{i,j \in \mathbb{Z}_n} \frac{1}{r^2} \{[x]_{i,j}^r, [y]_{i,j}^r\} = \langle x, y \rangle
\]  
(5)

For simplicity, we denote \(X_{ijl} = \langle F_l, [x]_{ij}^r \rangle\) and \(Y_{ijl} = \langle F_l, [y]_{ij}^r \rangle\). 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 larger than \(\epsilon\) can be derived as
\[
P\left(\frac{1}{N} \sum_{l=1}^{N} \{F_l \ast x, F_l \ast y\} - \langle x, y \rangle \geq \epsilon\right) \leq \mathbb{P}\left(\frac{1}{N} \sum_{l=1}^{N} \frac{1}{N} \sum_{i,j \in \mathbb{Z}_n} \{F_l \ast x, F_l \ast y\} - \mathbb{E}[\{F_l \ast x, F_l \ast y\}] \geq \epsilon\right) \leq \frac{1}{N} \sum_{i,j \in \mathbb{Z}_n} \frac{1}{N} \sum_{l=1}^{N} \{F_l \ast x, F_l \ast y\} - \mathbb{E}[\{F_l \ast x, F_l \ast y\}] \geq \epsilon
\]  
(6)

For the convolutional filters, \(X_{ijl} = \langle F_l, [x]_{ij}^r \rangle\) and \(Y_{ijl} = \langle F_l, [y]_{ij}^r \rangle\) 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 \(\{F_l, [x]_{ij}^r\}\) can be computed as
\[
\|\{F_l, [x]_{ij}^r\}\|_{\psi_2} = \|X_{ijl}\|_{\psi_2} = \|\beta k \mu + \gamma^2 \|F_l\|^2\|z\|_{\psi_2} \leq \|\beta k \mu\|_{\psi_2} + \|\gamma\|\|F_l\|\|z\|_{\psi_2} \leq k \beta \mu + C_0 W \gamma
\]  
(7)

and \(\|\{F_l, [y]_{ij}^r\}\|_{\psi_2} = \|X_{ijl}\|_{\psi_2} \leq k \beta \mu + C_0 W \gamma\) 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
\[
\|X_{ijl}Y_{ijl} - \mathbb{E}[X_{ijl}Y_{ijl}]\|_{\psi_1} \leq (k \beta \mu + C_0 W \gamma)^2.
\]  
(8)

Similarly, for the random projection filters, we have \(\|\{F_l, [x]_{ij}^r\}\|_{\psi_2} = \|X_{ijl}\|_{\psi_2} \leq C_0 R \frac{t}{r}\) and \(\|\{F_l, [y]_{ij}^r\}\|_{\psi_2} = \|Y_{ijl}\|_{\psi_2} \leq \frac{C_0 R}{r}\). According to the product of sub-Gaussians property and centering property, we have
\[
\|X_{ijl}Y_{ijl} - \mathbb{E}[X_{ijl}Y_{ijl}]\|_{\psi_1} \leq C_0 \nu^2 \frac{R^2}{r^2},
\]  
(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 \(t \geq 0\)
\[
P\left(\frac{1}{N} \sum_{i=1}^{N} X_i \geq t\right) \leq 2exp\left\{-\min\left\{\frac{t^2}{K^2}, \frac{t}{K}\right\} \cdot c \cdot N\right\},
\]  
(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

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

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 \( \frac{\epsilon - N - N_r}{K} \leq \left( \frac{\epsilon - N - N_r}{K^2} \right)^2 \), we have

\[
\delta > 2cn^2 \exp \left\{ - \frac{\epsilon N - D(N - N_r)}{K} \right\}
\]

\[
\log \frac{\delta}{2cn^2} > - \frac{(\epsilon - D)N + DN_r}{K} \cdot \frac{K}{\epsilon} \cdot \frac{2cn^2}{\delta} > (\epsilon - D)N + DN_r
\]

\[
N_r > \frac{(D - \epsilon)N + K \log \frac{2cn^2}{\delta}}{D} \tag{12}
\]

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

\[
N_r > \frac{(D - \epsilon)N + NK \log \frac{2cn^2}{\delta}}{D} \tag{13}
\]

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Clean</th>
<th>FGSM</th>
<th>PGD</th>
<th>CW</th>
<th>MIFGSM</th>
<th>DeepFool</th>
<th>AutoAttack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [5]</td>
<td>81.09</td>
<td>59.51</td>
<td>57.46</td>
<td>80.83</td>
<td>57.64</td>
<td>73.56</td>
<td>62.23</td>
</tr>
<tr>
<td>Add [5] Avg</td>
<td>81.16</td>
<td>59.41</td>
<td>57.60</td>
<td>80.94</td>
<td>57.83</td>
<td>73.57</td>
<td>62.14</td>
</tr>
<tr>
<td>Multi</td>
<td>82.91</td>
<td>61.89</td>
<td>59.77</td>
<td>82.70</td>
<td>59.96</td>
<td>78.49</td>
<td>63.96</td>
</tr>
<tr>
<td>Multi Avg</td>
<td>82.93</td>
<td>61.89</td>
<td>59.42</td>
<td>82.65</td>
<td>59.54</td>
<td>78.59</td>
<td>63.95</td>
</tr>
<tr>
<td>RPF</td>
<td>83.75</td>
<td>62.87</td>
<td>60.75</td>
<td>83.62</td>
<td>60.59</td>
<td>78.96</td>
<td>64.71</td>
</tr>
<tr>
<td>RPF(Ours) Avg</td>
<td>83.72</td>
<td>62.72</td>
<td>61.19</td>
<td>83.56</td>
<td>61.04</td>
<td>79.43</td>
<td>64.63</td>
</tr>
</tbody>
</table>
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 $10 \times 10$ in Pixle. The advantage of RPF over AT can be found in Table 2 where RPF achieves better robust accuracy in all the scenarios.

Table 2. Evaluation of black-box attacks.

<table>
<thead>
<tr>
<th>Attack</th>
<th>C-10 AT</th>
<th>C-10 RPF</th>
<th>C-100 AT</th>
<th>C-100 RPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square</td>
<td>53.64</td>
<td>76.56</td>
<td>29.57</td>
<td>48.21</td>
</tr>
<tr>
<td>Pixle</td>
<td>8.21</td>
<td>44.84</td>
<td>1.16</td>
<td>23.39</td>
</tr>
</tbody>
</table>


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.

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

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>R20 FGSM</th>
<th>R20 PGD</th>
<th>R32 FGSM</th>
<th>R32 PGD</th>
<th>R44 FGSM</th>
<th>R44 PGD</th>
<th>R56 FGSM</th>
<th>R56 PGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNI</td>
<td>54.40</td>
<td>45.90</td>
<td>51.50</td>
<td>43.50</td>
<td>55.80</td>
<td>48.50</td>
<td>53.90</td>
<td>46.30</td>
</tr>
<tr>
<td>Learn2Perturb</td>
<td>58.41</td>
<td>51.13</td>
<td>59.94</td>
<td>54.62</td>
<td>61.32</td>
<td>54.62</td>
<td>61.53</td>
<td>54.62</td>
</tr>
<tr>
<td>RPF</td>
<td>63.27</td>
<td>60.94</td>
<td>62.52</td>
<td>60.78</td>
<td>63.39</td>
<td>62.47</td>
<td>62.30</td>
<td>60.97</td>
</tr>
</tbody>
</table>
References


