Supplementary Material for "Frequency Domain Model Augmentation for Adversarial Attack"

Yuyang Long[1], Qilong Zhang[1], Boheng Zeng[1], Lianli Gao[1], Xianglong Liu[2], Jian Zhang[3], and Jingkuan Song[1*]

[1] Center for Future Media, University of Electronic Science and Technology of China
[2] Beihang University [3] Hunan University
yuyang.long@outlook.com,{qilong.zhang,boheng.zeng}@std.uestc.edu.cn,
lianli.gao@uestc.edu.cn,xlliu@buaa.edu.cn,jianzh@hnu.edu.cn,
jingkuan.song@gmail.com

In this supplementary material, we provide: (A) theoretical proof of our proposed method, (B) discussion on hyper-parameters for our proposed S^{2}I-FGSM, (D) discussion on spatial domain transformation analysis and visualizations for spectrum transformation images.

A Proof

Proposition 1. *Our proposed spectrum transformation can generate diverse spectrum saliency maps and thus simulate diverse substitute models.*
Proof. According to *Lagrange's mean value theorem*:

$$\frac{\partial J(\boldsymbol{x_1}, y; \phi)}{\partial \boldsymbol{x_1}} = \frac{\partial J(\boldsymbol{x_2}, y; \phi)}{\partial \boldsymbol{x_2}} + \boldsymbol{K}, \tag{1}$$

where $\boldsymbol{K} = \frac{\partial^2 J(\boldsymbol{\zeta}, y; \phi)}{\partial \boldsymbol{\zeta}^2}(\boldsymbol{x_1} - \boldsymbol{x_2}), \boldsymbol{\zeta} \in [\boldsymbol{x_2}, \boldsymbol{x_1}]$.

Without spectrum transformation function $\mathcal{T}(\cdot)$, spectrum saliency map:

$$\boldsymbol{S}_\phi = \frac{\partial J(\mathcal{D}_\mathcal{I}(\mathcal{D}(\boldsymbol{x})), y; \phi)}{\partial \mathcal{D}(\boldsymbol{x})}, \tag{2}$$

after applying our proposed spectrum transformation function $\mathcal{T}(\cdot)$, the resulting spectrum saliency map:

$$\boldsymbol{S}'_\phi = \frac{\partial J(\mathcal{T}(\boldsymbol{x}), y; \phi)}{\partial \mathcal{D}(\boldsymbol{x})}, \tag{3}$$

where $\mathcal{T}(\boldsymbol{x}) = \mathcal{D}_\mathcal{I}((\mathcal{D}(\boldsymbol{x}) + \mathcal{D}(\boldsymbol{\xi})) \odot \boldsymbol{M})$

Let $\boldsymbol{D_1}$ denotes $\frac{\partial J(\mathcal{D}_\mathcal{I}(\mathcal{D}(\boldsymbol{x})), y; \phi)}{\partial \mathcal{D}_\mathcal{I}(\mathcal{D}(\boldsymbol{x}))}$ and $\boldsymbol{D_2}$ denotes $\frac{\partial \mathcal{D}_\mathcal{I}(\mathcal{D}(\boldsymbol{x}))}{\partial \mathcal{D}(\boldsymbol{x})}$, then $\boldsymbol{S}_\phi = \boldsymbol{D_1}\boldsymbol{D_2}$ (according to chain rule). After applying $\mathcal{T}(\cdot)$ to $\boldsymbol{x}$, resulting spectrum saliency map $\boldsymbol{S}'_\phi$ can be expressed as:

$$\boldsymbol{S}'_\phi = \boldsymbol{D}'_\boldsymbol{1}\boldsymbol{D}'_\boldsymbol{2} \odot \boldsymbol{M}, \tag{4}$$

*Corresponding author

where

$$\boldsymbol{D_1'} = \frac{\partial J(\mathcal{D}_{\mathcal{I}}(\mathcal{D}(\boldsymbol{x}+\boldsymbol{\xi})\odot\boldsymbol{M}), y; \phi)}{\partial \mathcal{D}_{\mathcal{I}}(\mathcal{D}(\boldsymbol{x}+\boldsymbol{\xi})\odot\boldsymbol{M})}, \tag{5}$$

$$\boldsymbol{D_2'} = \frac{\partial \mathcal{D}_{\mathcal{I}}(\mathcal{D}(\boldsymbol{x}+\boldsymbol{\xi})\odot\boldsymbol{M})}{\partial(\mathcal{D}(\boldsymbol{x}+\boldsymbol{\xi})\odot\boldsymbol{M}))}. \tag{6}$$

Based on Eq. 1, we can formally formulate $\boldsymbol{S_\phi'}$ to be:

$$\begin{aligned} \boldsymbol{S_\phi'} &= (\boldsymbol{D_1}+\boldsymbol{K_1})(\boldsymbol{D_2}+\boldsymbol{K_2})\odot\boldsymbol{M}, \\ &= (\boldsymbol{S_\phi}+\boldsymbol{K'})\odot\boldsymbol{M}, \end{aligned} \tag{7}$$

where $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$ are two specific matrices, and $\boldsymbol{K'} = \boldsymbol{D_1K_2}+\boldsymbol{D_2K_1}+\boldsymbol{K_1K_2}$. Eq. 7 clearly demonstrates that our proposed transformation $\mathcal{T}(\cdot)$ is capable of simulating a different spectrum saliency map.

B On the Hyper-Parameters Settings

We first study the influence of the hyper-parameters(*i.e.*, standard deviation (std) σ of noise $\boldsymbol{\xi}$, tuning factor ρ of matrix $\boldsymbol{M}$, number N of spectrum transformations) for the proposed Spectrum Simulation Attack method.

B.1 On the Standard Deviation σ of Noise $\boldsymbol{\xi}$

In Figure 1, we report the attack success rates of S^2I-FGSM for different std σ. Adversarial examples are crafted via Inc-v3 with $N = 20$ and $\rho = 0.5$. Particularly, $\sigma = 0$ means no noise is added to the input. A first glance shows that for normally trained models, the attack success rates increase gradually as σ increases and then tend to decrease when σ exceeds 16. Also when $\sigma = 16$, the defense models can achieve relatively high attack success rates. Therefore, we set $\sigma = 16$ in our paper.

B.2 On the Tuning Factor ρ of Matrix $\boldsymbol{M}$

In this section, we study the effect of tuning factor ρ for our S^2I-FGSM in Figure 2. Adversarial examples are crafted via Inc-v3 with $N = 20$ and $\sigma = 16$. Particularly, $\rho = 0$ means there is no tuning on the spectrum. Similarly, as ρ increases, the degree of spectrum transformation becomes stronger and the attack success rates gradually increase and peak at $\rho = 0.5$. If we continue to increase ρ (*i.e.* $\rho > 0.5$), the attack success rates will decrease which may be attributed to the excessive spectrum transformation. To achieve better transferability, we choose $\rho = 0.5$ in our paper.

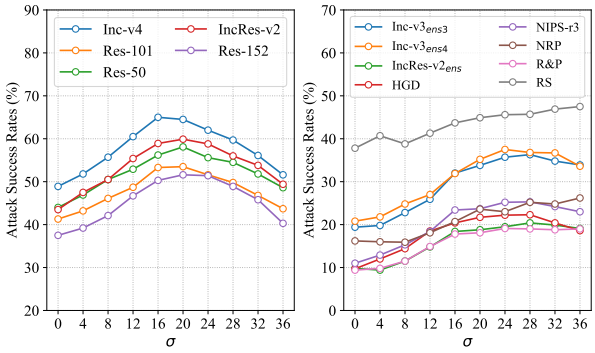

Fig. 1: The attack success rates (%) of S^{2}I-FGSM on normally trained and defense models w.r.t. the std σ of ξ. Adversarial examples are generated via Inc-v3. **Left**: The results for fooling normally trained models. **Right**: The results for fooling defense models.

B.3 On the Number N of Spectrum Transformations.

In this section, we study the effect of number N of spectrum transformations for our S^{2}I-FGSM in Figure 3. Adversarial examples are crafted via Inc-v3 with $\rho = 0.5$ and $\sigma = 16$. As shown in Figure 3, when $N = 1$, our method performs only one spectrum transformation and achieves the lowest transferability. As N increases, the transferability of adversarial examples is significantly enhanced at first, and turns to increase slowly after N exceeds 20. It also demonstrates that our spectrum transformation can effectively narrow the gap between the substitute model and victim model. It is worth noting that larger N implies expensive computational overhead, as we need more forward and backward propagation for gradient computation at each iteration. To balance the transferability and computational overhead, we choose $N = 20$ in our paper.

C Time Analysis of DCT/IDCT

In our experiments, we directly apply DCT/IDCT on the full image which is a time-consuming operation. Therefore, in this section we analyze the time consumption of DCT/IDCT. In Tab.1 we show the average time of an adversarial example generated by S^{2}I-FGSM and the average time of DCT/IDCT among it. For example, let IncRes-v2 be the substitute model, S^{2}I-FGSM takes an average of 3.78s to produce an adversarial example, of which DCT/IDCT takes up 0.58s (only accounts for 15.3% of all overheads). The experiment is conducted on RTX 3090 GPUs.

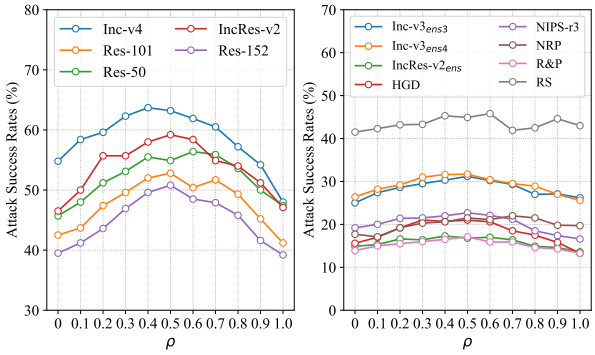

Fig. 2: The attack success rates (%) of S^{2}I-FGSM on normally trained and defense models w.r.t. the tuning factor ρ. Adversarial examples are generated via Inc-v3. **Left**: The results for fooling normally trained models. **Right**: The results for fooling defense models.

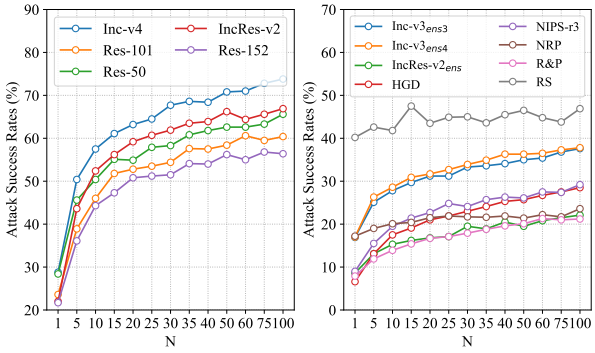

Fig. 3: The attack success rates (%) of S^{2}I-FGSM on normally trained and defense models w.r.t. the number N of spectrum transformations. Adversarial examples are generated via Inc-v3. **Left**: The results for fooling normally trained models. **Right**: The results for fooling defense models.

Table 1: The average time (s) of generating an adversarial example on Inc-v3, Inc-v4, IncRes-v2 and Res-152, respectively. The left side of slash indicates the time of DCT/IDCT and right side indicates the time of S^{2}I-FGSM.

	Inc-v3	Inc-v4	IncRes-v2	Res-152
Time	0.60/1.89	0.61/2.85	0.58/3.78	0.61/3.05

D Additional Results

D.1 Spatial Domain Transformation Analysis

In this section, we further validate our point that analysis on spatial domain cannot well reflect the gap between models. To support our point, we first define spatial saliency map $\hat{\boldsymbol{S}}_{\boldsymbol{\phi}}$ as:

$$\hat{\boldsymbol{S}}_{\boldsymbol{\phi}} = \frac{\partial J(\boldsymbol{x}, y; \phi)}{\partial \boldsymbol{x}}, \tag{8}$$

which is similar to our proposed spectrum saliency map $\boldsymbol{S}_{\boldsymbol{\phi}}$ in Eq. 4. Then we flip the image horizontally (spatial domain transformation) and analyze their spatial saliency map and frequency saliency map. As shown in Figure 4, although spatial saliency maps between raw image and fliped image vary greatly, the changes in frequency spectrum and frequency saliency map (an indicator reflecting the characteristics of models) are small. Thus, analysis on spatial domain is unreliable and can hardly reflect the gap between models.

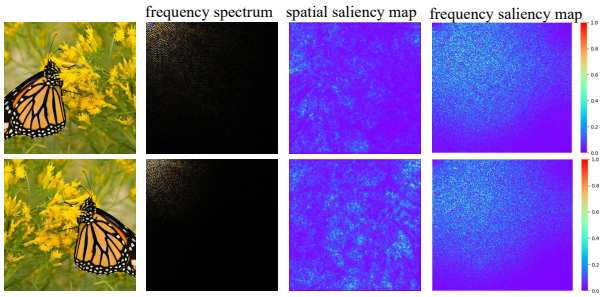

Fig. 4: Visualization for frequency spectrum, spatial saliency map, and frequency saliency map. Top raw corresponds to raw image, and bottom row corresponds to spatial domain transformed image. This result demonstrates that analysis on spatial domain is unreliable.

D.2 Spectrum Transformation Images

To better understand the process of our method, we visualize the outputs of spectrum transformation. Specifically, we perform several spectrum transformations on input images and show the resulting spectrum transformation outputs in Figure 5. This figure shows that spectrum transformation just modifies colors of image and does not change its semantic information.

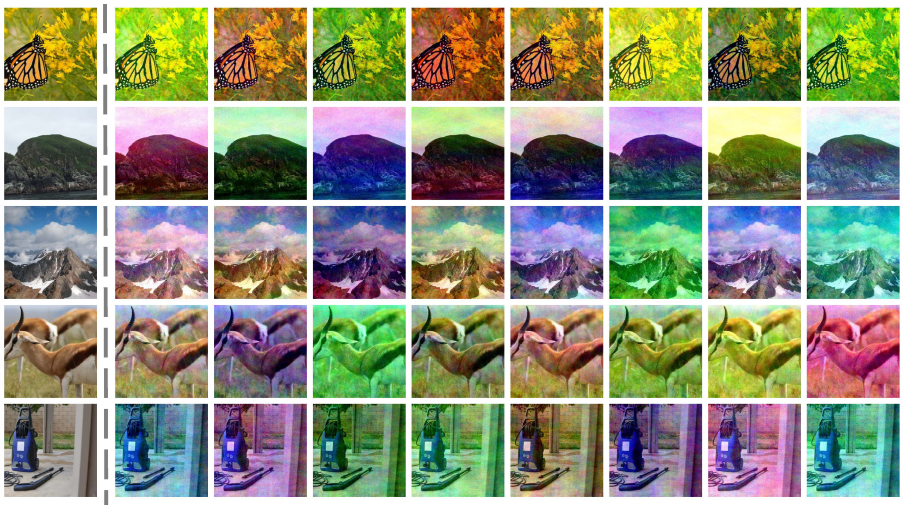

Fig. 5: Visualization for the spectrum transformation outputs (right columns) w.r.t. raw input images (left column). This result shows that spectrum transformation just modifies colors of image and does not change its semantic information.