Phase-shifted Adversarial Training (Supplementary Material)

Yeachan Kim[1] **Seongyeon Kim**[*2] **Ihyeok Seo**[*3] **Bonggun Shin**[*4]

[1]Deargen Inc., Seoul, Republic of Korea
[2]School of Mathematics, Korea Institute for Advanced Study, Seoul, Republic of Korea
[3]Department of Mathematics, Sungkyunkwan University, Suwon, Republic of Korea
[4]Deargen USA Inc., Atlanta, GA

1 FILTERING METHOD FOR FREQUENCY ANALYSIS

Motivated by the examination of F-principle [Xu et al., 2020], we use the filtering method to analyze the behavior of the neural networks in adversarial training. The idea is to split the frequency domain into two parts, i.e., low-frequency and high-frequency parts. However, the Fourier transform for high-dimensional data requires high computational costs and large memory footprints. As an alternative, we use the Fourier transform of a Gaussian function $\hat{G}$.

Let the original dataset be $\{x_j, y_j\}_{j=0}^{N-1}$, and the network output for x_j be $\mathcal{T}_j$. The low frequency part of the training dataset can be derived by

$$y_j^{low,\delta} = \frac{1}{C_j} \sum_{m=0}^{N-1} y_m G^{\delta}(x_j - x_m) \tag{1.1}$$

where $C_j = \sum_{m=0}^{N-1} G^{\delta}(x_j - x_m)$ is a normalization factor, and δ is the variance of the Gaussian function (we fix δ to 3). The Gaussian function can be represented as

$$G^{\delta}(x_j - x_m) = \exp(-|x_j - x_m|^2/(2\delta)). \tag{1.2}$$

Then, the high-frequency part can be derived by $y_j^{high,\delta} \triangleq y_j - y_j^{low,\delta}$. We also compute the frequency components for the networks, i.e, $\mathcal{T}_j^{low,\delta}$, $\mathcal{T}_j^{high,\delta}$ by replacing y_j with the outputs of networks, i.e., $\mathcal{T}_j$. Lastly, we calculate the errors to quantify the convergence in terms of low- and high-frequency.

$$e_{low} = \left(\frac{\sum_j \left| y_j^{low,\delta} - \mathcal{T}_j^{low,\delta} \right|^2}{\sum_j \left| y_j^{low,\delta} \right|^2} \right)^{\frac{1}{2}} \tag{1.3}$$

$$e_{high} = \left(\frac{\sum_j \left| y_j^{high,\delta} - \mathcal{T}_j^{high,\delta} \right|^2}{\sum_j \left| y_j^{high,\delta} \right|^2} \right)^{\frac{1}{2}} \tag{1.4}$$

2 ITERATIVE-VERSION OF PHASEAT

For training efficiency, we design PhaseAT as a non-iterative method based on the FGSM perturbation [Wong et al., 2020]. To confirm the effect of stronger attacks in the training process of PhaseAT, we additionally introduce an iterative version of PhaseAT. Since PhaseAT is not closely related to the perturbation generation, we replace the FGSM perturbation with the perturbation generated from PGD [Madry et al., 2018]. The overall algorithm is shown in Algorithm-1.

[*] Co-corresponding authors who equally contributed to this work.

Algorithm 1 Phase-shifted Adversarial Training (Iterative version)

```
Require: Training epochs $T$, Dataset size $N$, PGD steps $P$, Perturbation size $\epsilon$, Perturbation step $\alpha$, Trainable networks $\mathcal{T}$,
    Cosine similarity function $CS(\cdot,\cdot)$
for $t = 1 ... T$ do
    for $j = 1 ... N$ do
        $\delta$ = Uniform$(-\epsilon, \epsilon)$
        for $k = 1 ... P$ do                                    ▷ Multiple updates of perturbations
            if $j$ % 2 == 0 then                                ▷ Alternate training on mini-batches
                $\delta = \delta + \alpha\cdot$ sign$(\nabla_\delta \ell(\mathcal{T}(x_j + \delta), y_j))$
            else
                $\delta = \delta + \alpha\cdot$ sign$(\nabla_\delta \ell(\mathcal{T}_0(x_j + \delta), y_j))$
            end if
            $\delta$ = max(min$(\delta, \epsilon)$, $-\epsilon$)
        end for
        $\theta = \theta - \nabla_\theta[\ell(\mathcal{T}(x_j + \delta), y_j) + CS(\mathcal{T}(x_j + \delta), \mathcal{T}_0(x_j + \delta))]$
    end for
end for
```

3 DETAILS ABOUT EVALUATION

3.1 ATTACK CONFIGURATION

In our work, we mainly adopt the projected gradient descent (PGD) [Madry et al., 2018] and auto-attack (AA) [Croce and Hein, 2020b] to evaluate baselines. PGD is constructed by multiple updates of adversarial perturbations, and AA is the ensemble of strong attacks including the variants of PGD. Typically, AA is considered one of the strongest attacks. The details about each attack of AA are as follows:

- Auto-PGD (APGD) [Croce and Hein, 2020b]: This is parameter-free adversarial attack that adaptively changes the step size by considering the optimization of the perturbations. APGD has three variations depending on loss functions: APGD$_{ce}$, APGD$_{dlr}$, and APGD$_t$[1].
- FAB [Croce and Hein, 2020a]: This attack minimizes the norm of the perturbation necessary to achieve a misclassification. FAB has two variants, FAB and FAB$_t$.
- Square [Andriushchenko et al., 2020]: Compared to others, this attack belongs to the black-box attacks and is also known as score-based attack. This attack iteratively inserts an artificial square to the inputs to search optimal perturbations causing huge changes on predictions.

We set the hyper-parameter settings for each attack based on *standard* version of AA in *robust-bench* framework [Croce et al., 2021]. Note that we exclude Square attack from the AA because the stochastic process in PhaseAT can be robust against Square [Qin et al., 2021], which prevents the fair comparison with other baselines which do not include stochastic process. We thus move the results of Square attack to the Supplementary Section 4.2.

3.2 DATASET INFORMATION

We evaluate each baseline on two benchmark datasets, CIFAR-10 and ImageNet. CIFAR-10 [Krizhevsky et al., 2009] consist of 60,000 images of 32×32×3 size for 10 classes, and ImageNet contains 1.2M images of 224×224×3 size for 1,000 classes. Instead of existing ImageNet, we use the smaller version of ImageNet which used in recent baselines [Sriramanan et al., 2020, 2021], which contains 120K images of 224×224×3 size for 100 classes [2].

[1] Subscript *ce* and *dlr* on APGD indicates the *cross-entropy loss* and *difference of logits ratio*, respectively, and t stands for targeted attacks. The attacks without t subscripts are non-targeted attacks.

[2] Selected classes are listed in `https://github.com/val-iisc/GAMA-GAT`

Table 1: Performance evaluation on CIFAR-10 dataset. The backbone networks are **WideResNet-34-10**. Best and second best results are highlighted in boldface and underline, respectively.

Method	Standard accuracy	PGD_{50}	AA
FBF [Wong et al., 2020]	82.1	54.4	51.3
GAT [Sriramanan et al., 2020]	84.7	56.1	52.1
NuAT [Sriramanan et al., 2021]	85.1	54.6	53.4
PhaseAT (Ours.)	88.8	62.3	59.2

Table 2: Performance evaluation on CIFAR-10 dataset against two different black-box attacks.

Method	Standard accuracy	Transfer-based attack		Score-based attack
		VGG-11	ResNet-18	
FBF [Wong et al., 2020]	84.0	80.5	80.6	53.5
GAT [Sriramanan et al., 2020]	80.5	79.8	80.3	54.1
NuAT [Sriramanan et al., 2021]	81.6	79.5	80.5	56.7
PhaseAT (Ours.)	86.2	83.8	85.0	76.5

3.3 BASELINE SETTING

PhaseAT is compared to both non-iterative (FBF, GAT, and NuAT) and iterative (FBF, GAT, and NuAT) methods (PGD, TRADES, and AWP). The hyper-parameter settings of each baseline are listed in Table 3. Since the evaluation results on ImageNet come from previous works [Sriramanan et al., 2020, 2021], the table only includes the parameters reported in these works (unknown parameters are denoted with $-$.).

4 ADDITIONAL EVALUATION

4.1 DIFFERENT ARCHITECTURES

We conduct additional experiments by scaling the PhaseAT backbone networks to verify the effectiveness of PhaseAT on different architectures. We use WideResNet-34-10 architecture instead of PreActResNet-18 to evaluate each baseline on CIFAR-10.The comparison results are listed in Table 1. Similar to the main experiment, we see that PhaseAT achieves the best results amonst all non-iterative methods, demonstrating that PhaseAT can be well scaled to the larger networks.

4.2 ADVERSARIAL ROBUSTNESS AGAINST BLACK-BOX ATTACKS

As DNN models are often hidden from users in real-world applications, the robustness against black-box attacks is also crucial. Among the different kinds of black-box attacks, we consider transfer-based [Liu et al., 2016, Papernot et al., 2017] and score-based attacks. For transfer-based attacks, we use VGG-11 and ResNet-18 as substitute models and construct the attacks using seven steps of PGD [Madry et al., 2018]. For score-based attacks, we adopt square attack [Andriushchenko et al., 2020] with 5,000 query budgets, which is a gradient-free attack and one of the strongest attacks in black-box attacks.

Table 2 shows the robust accuracy against black-box attacks. Similar to white-box attacks, PhaseAT shows better accuracy against both transfer-based and score-based attacks in comparison to other non-iterative methods. In score-based attacks, the difference in performance between others and PhaseAT is particularly noticeable. This can be explained by the stochastic process of PhaseAT because Qin et al. [Qin et al., 2021] demonstrate that randomized defense (e.g., Gaussian noise in the inputs) can robustly prevent the model from score-based attacks. This is why we exclude square attack from the AA attack for a fair comparison with other baselines that do not include the stochastic process. Note that the stochastic property can not be circumvented in the black-box scenario because it is infeasible to design adaptive attacks (i.e., EOT attacks) as in the white-box scenario. Comprehensive results show that PhaseAT could be a robust defense strategy against both white-box and black-box attacks.

5 PROOFS OF THEOREMS 3.1 AND 3.2

5.1 PRELIMINARIES

Before proving Theorems 3.1 and 3.2 in this section, we start with a detailed explanation of DNNs, and then introduce the mathematical tools required for proof which can be found in standard references (e.g. [Stein and Weiss, 1971, Wolff, 2003, Muscalu and Schlag, 2013, Evans, 2010]).

Deep Neural Networks. A DNN with K-hidden layers and general activation functions is a vector-valued function $\mathcal{T}_\theta(x) : \mathbb{R}^d \to \mathbb{R}^{m_{K+1}}$ where m_k denotes the number of nodes in the k-th layer. For $1 \le k \le K+1$, we set $\boldsymbol{W}^{(k)} \in \mathbb{R}^{m_k \times m_{k-1}}$ and $\boldsymbol{b}^{(k)} \in \mathbb{R}^{m_k}$ as the matrices whose entries consist of the weights and biases called parameters. The parameter vector θ is then defined as

$$\theta = \big(\mathrm{vec}(\boldsymbol{W}^{(1)}), \mathrm{vec}(\boldsymbol{b}^{(1)}), \cdots, \mathrm{vec}(\boldsymbol{W}^{(K+1)}), \mathrm{vec}(\boldsymbol{b}^{(K+1)})\big) \in \mathbb{R}^M,$$

where $M = \sum_{k=1}^{K+1}(m_{k-1}+1)m_k$ is the number of the parameters. Given $\theta \in \mathbb{R}^M$ and an activation function $\sigma : \mathbb{R} \to \mathbb{R}$, the DNN output $\mathcal{T}_\theta^{(K+1)}(x) : \mathbb{R}^d \to \mathbb{R}^{m_{K+1}}$ is expressed in terms of composite functions; setting $\mathcal{T}_\theta^{(0)}(x) = x$, $\mathcal{T}_\theta^{(k)}(x) : \mathbb{R}^d \to \mathbb{R}^{m_k}$ is defined recursively as

$$(\mathcal{T}_\theta^{(k)}(x))_i = \sigma\big((\boldsymbol{W}^k \mathcal{T}_\theta^{k-1} + \boldsymbol{b}^k)_i\big), \ 1 \le i \le m_k, \ 1 \le k \le K.$$

We denote the DNN output $\mathcal{T}_\theta^{(K+1)}(x) = \boldsymbol{W}^{(K+1)}\mathcal{T}_\theta^{(K)} + \boldsymbol{b}^{(K+1)}$ by $\mathcal{T}_\theta(x)$.

The Basic Properties of Fourier Transforms. Let $f \in L^1(\mathbb{R}^d)$. The Fourier transform of f is defined by

$$\widehat{f}(\xi) = \int_{\mathbb{R}^d} e^{-2\pi i x\cdot\xi} f(x)dx.$$

Then clearly

$$\|\widehat{f}\|_{L^\infty} \le \|f\|_{L^1}. \tag{5.1}$$

Additionally if $\widehat{f} \in L^1(\mathbb{R}^d)$, the Fourier inversion holds:

$$f(x) = \int_{\mathbb{R}^d} e^{2\pi i x\cdot\xi}\widehat{f}(\xi)d\xi. \tag{5.2}$$

If $f, g \in L^1(\mathbb{R}^d)$, then $f * g \in L^1(\mathbb{R}^d)$ and

$$\widehat{f * g} = \hat{f}\hat{g}. \tag{5.3}$$

For an n-tuple $\alpha = (\alpha_1, \cdots, \alpha_d)$ of nonnegative integers, we denote

$$D^\alpha = \prod_{j=1}^{d} \frac{\partial^{\alpha_j}}{\partial_{x_j}^{\alpha_j}} \quad \text{and} \quad |\alpha| = \sum_{j=1}^{d} \alpha_j.$$

Then, if $D^\alpha f \in L^1(\mathbb{R}^d)$ whenever $0 \le |\alpha| \le s$,

$$\widehat{D^\alpha f}(\xi) = (2\pi i)^{|\alpha|}\xi^\alpha \widehat{f}(\xi). \tag{5.4}$$

Sobolev Spaces and Gaussian Weights. For $s \in \mathbb{N}$, the Sobolev space $W^{s,\infty}(\mathbb{R}^d)$ is defined as

$$W^{s,\infty}(\mathbb{R}^d) = \{f \in L^\infty(\mathbb{R}^d) : D^\alpha f \in L^\infty(\mathbb{R}^d) \text{ for all } 0 \le |\alpha| \le s\}$$

equipped with the norm

$$\|f\|_{W^{s,\infty}(\mathbb{R}^d)} = \sum_{|\alpha|\le s} \|D^\alpha f\|_{L^\infty(\mathbb{R}^d)}.$$

We also introduce a Gaussian weight $G_\varepsilon(x) = \varepsilon^{-d}e^{-\pi\varepsilon^{-2}|x|^2}$ for any $\varepsilon > 0$ on which the Fourier transform has an explicit form,

$$\widehat{G_\varepsilon}(\xi) = e^{-\pi\varepsilon^2|\xi|^2}. \tag{5.5}$$

The final observation is that G_ε is an approximate identity with respect to the limit $\varepsilon \to 0$ as in the following well-known lemma:

Lemma 5.1. *Let $f \in C(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d)$. Then*

$$\lim_{\varepsilon \to 0} \int_{\mathbb{R}^d} G_\varepsilon(x-y) f(y) dy = f(x) \tag{5.6}$$

for all $x \in \mathbb{R}^d$.

5.2 PROOF OF THEOREM 3.1

In what follows we may consider a compact domain Ω instead of $\mathbb{R}^d$ because the input data $\{x_j\}_{j=0}^{N-1}$ used for training is sampled from a bounded region.

For a discrete input data $\{x_j\}_{j=0}^{N-1}$, we now recall the total loss in adversarial training from Section 3.1:

$$L(\theta) = \frac{1}{N} \sum_{j=0}^{N-1} \ell(\mathcal{T}_\theta \circ \mathcal{A}, g)(x_j). \tag{5.7}$$

From the continuity of $\mathcal{T}_\theta$ and g in the compact domain Ω, we note that $\ell(\mathcal{T}_\theta \circ A, g)$ is continuous and bounded for general loss functions such as mean-squared error loss and cross-entropy loss. Then we can apply Lemma 5.1 to deduce

$$\begin{aligned} L(\theta) &= \lim_{\varepsilon \to 0} \frac{1}{N} \sum_{j=0}^{N-1} \int_{\mathbb{R}^d} G_\varepsilon(x_j - x) \ell(\mathcal{T}_\theta \circ \mathcal{A}, g)(x) dx \\ &= \lim_{\varepsilon \to 0} \frac{1}{N} \sum_{j=0}^{N-1} \big(G_\varepsilon * \ell(\mathcal{T}_\theta \circ \mathcal{A}, g)\big)(x_j). \end{aligned} \tag{5.8}$$

Using the properties $G_\varepsilon \in L^1(\mathbb{R}^d)$ and $\ell(\mathcal{T}_\theta \circ \mathcal{A}, g) \in L^1(\mathbb{R}^d)$, we then derive from Eq. 5.3 and Eq. 5.5 that

$$G_\varepsilon * \ell(\mathcal{T}_\theta \circ \mathcal{A}, g) \in L^1(\mathbb{R}^d)$$

and

$$\widehat{G_\varepsilon * \ell(\mathcal{T}_\theta \circ \mathcal{A}, g)}(\xi) = e^{-\pi\varepsilon^2|\xi|^2}\, \widehat{\ell(\mathcal{T}_\theta \circ \mathcal{A}, g)}(\xi). \tag{5.9}$$

Note here that by Eq. 5.1

$$\|e^{-\pi\varepsilon^2|\xi|^2}\, \widehat{\ell(\mathcal{T}_\theta \circ \mathcal{A}, g)}(\xi)\|_{L^1} \leq \|\widehat{\ell(\mathcal{T}_\theta \circ \mathcal{A}, g)}\|_{L^\infty} \|e^{-\pi\varepsilon^2|\xi|^2}\|_{L^1} \leq C \|\ell(\mathcal{T}_\theta \circ \mathcal{A}, g)\|_{L^1} < \infty.$$

Hence the Fourier inversion Eq. 5.2 together with Eq. 5.9 implies

$$\big(G_\varepsilon * \ell(\mathcal{T}_\theta \circ \mathcal{A}, g)\big)(x_j) = \int_{\mathbb{R}^d} e^{2\pi i x_j \cdot \xi} e^{-\pi\varepsilon^2|\xi|^2}\, \widehat{\ell(\mathcal{T}_\theta \circ \mathcal{A}, g)}(\xi) d\xi. \tag{5.10}$$

Substituting Eq. 5.10 into the right-hand side of Eq. 5.8, we immediately obtain

$$L(\theta) = \lim_{\varepsilon \to 0} \frac{1}{N} \sum_{j=0}^{N-1} \int_{\mathbb{R}^d} e^{2\pi i x_j \cdot \xi}\, e^{-\pi\varepsilon^2|\xi|^2} \widehat{\ell\big(\mathcal{T}_\theta \circ \mathcal{A}, g\big)}(\xi) d\xi, \tag{5.11}$$

as desired. This completes the proof.

5.3 PROOF OF THEOREM 3.2

Representing $\nabla_\theta L(\theta)$ in the frequency domain. To begin with, we represent $\nabla_\theta L(\theta)$ in the frequency domain in the same way as in Section 5.2. By differentiating both sides of Eq. 5.7 with respect to θ and using Lemma 5.1, we first see

$$\begin{aligned} \nabla_\theta L(\theta) &= \frac{1}{N} \sum_{j=0}^{N-1} \nabla_\theta \ell(\mathcal{T}_\theta \circ \mathcal{A}, g)(x_j) \\ &= \lim_{\varepsilon \to 0} \frac{1}{N} \sum_{j=0}^{N-1} \int_{\mathbb{R}^d} G_\varepsilon(x_j - x) \nabla_\theta \ell(\mathcal{T}_\theta \circ \mathcal{A}, g)(x) dx \end{aligned} \tag{5.12}$$

if $\nabla_\theta \ell(\mathcal{T}_\theta \circ \mathcal{A}, g)$ is continuous and bounded. Since $\ell(\mathcal{T}_\theta \circ \mathcal{A}, g)$ is differentiable with respect to the first argument (as mentioned in Section 3.1) and $\mathcal{T}_\theta$ is differentiable with respect to θ for general activation functions such as ReLU, eLU, tanh and sigmoid, the continuity is generally permissible, and thus the boundedness follows also from compact domain. In fact, the ReLU activation function is not differentiable at the origin and neither is $\mathcal{T}_\theta$ on a certain union of hyperplanes; for example, when considering 1-hidden layer neural network with m_1 nodes and 1-dimensional output, the output is

$$\mathcal{T}_\theta(x) = \sum_{i=1}^{m_1} w_i^{(2)} \sigma(\boldsymbol{W}_i^{(1)} \cdot x + \boldsymbol{b}_i^{(1)}), \quad w_i^{(2)}, b_i^{(1)} \in \mathbb{R},\ \boldsymbol{W}_i^{(1)} \in \mathbb{R}^d$$

and the set of non-differentiable points is a union of hyperplanes given by $\{x \in \mathbb{R}^d : \boldsymbol{W}_i^{(1)} \cdot x + \boldsymbol{b}_i^{(1)} = 0,\ 1 \le i \le m_1\}$. But the d-dimensional volume of such thin sets is zero and thus they may be excluded from the integration region in Eq. 5.6 when applying Lemma 5.1 to obtain Eq. 5.12 for the case of ReLU.

Just by replacing $L(\theta)$ with $\nabla_\theta L(\theta)$ in the argument employed for the proof of Eq. 5.11 and repeating the same argument, it follows now that

$$\nabla_\theta L(\theta) = \lim_{\varepsilon \to 0} \frac{1}{N} \sum_{j=0}^{N-1} \int_{\mathbb{R}^d} e^{2\pi i x_j \cdot \xi}\, e^{-\pi \varepsilon^2 |\xi|^2}\, \widehat{\nabla_\theta \ell\big(\mathcal{T}_\theta \circ \mathcal{A}, g\big)}(\xi) d\xi. \tag{5.13}$$

We then pull ∇_θ to the outside of the integration in Eq. 5.13, and recall $L_{\le\eta}(\theta)$ and $L_{\ge\eta}(\theta)$ from Section 3.1, contributed by low and high frequencies in the loss, to see

$$\nabla_\theta L(\theta) \approx \nabla_\theta L_{\le\eta}(\theta) + \nabla_\theta L_{\ge\eta}(\theta). \tag{5.14}$$

This approximation is more and more accurate as ε diminishes smaller in Eq. 5.13, and the size of ε will be later determined inversely proportional to the number of dataset N or dimension d to consider a natural approximation reflecting the discrete experimental setting.

Estimating $\nabla_\theta L_{\ge\eta}(\theta)$ in terms of η. Now we show that for the i-th element of $\nabla_\theta L_{\ge\eta}(\theta)$

$$\Big| \frac{\partial L_{\ge\eta}(\theta)}{\partial \theta_i} \Big| \le C \max(N, d^d)\, \eta^{-2s} \tag{5.15}$$

which implies

$$|\nabla_\theta L_{\ge\eta}(\theta)| = \Big(\sum_\theta \Big| \frac{\partial L_{\ge\eta}(\theta)}{\partial \theta_i} \Big|^2 \Big)^{1/2} \le C \max(N, d^d)\, \eta^{-2s}.$$

By Eq. 5.14 and this bound, we get

$$|\nabla_\theta L(\theta) - \nabla_\theta L_{\le\eta}(\theta)| \approx |\nabla_\theta L_{\ge\eta}(\theta)| \le C \max(N, d^d)\, \eta^{-2s}$$

which completes the proof of Theorem 3.2.

To show Eq. 5.15, we first use the chain rule to calculate

$$\frac{\partial L_{\ge\eta}(\theta)}{\partial \theta_i} = \frac{1}{N} \sum_{j=0}^{N-1} \int_{|\xi| \ge \eta} e^{2\pi i x_j \cdot \xi} e^{-\pi \varepsilon^2 |\xi|^2} \widehat{\nabla_{\mathcal{T}_\theta} \ell(\mathcal{T}_\theta \circ \mathcal{A}, g) \cdot \tfrac{\partial(\mathcal{T}_\theta \circ \mathcal{A})}{\partial \theta_i}}(\xi) d\xi.$$

Since $\eta \le \langle \xi \rangle := \sqrt{1 + |\xi|^2}$ for all $0 < \eta \le |\xi|$, we then see that for $s \in \mathbb{N}$

$$\begin{aligned} \Big| \frac{\partial L_{\ge\eta}(\theta)}{\partial \theta_i} \Big| &\le \frac{1}{N} \sum_{j=0}^{N-1} \eta^{-2s} \int_{|\xi| \ge \eta} e^{-\pi \varepsilon^2 |\xi|^2} \Big| \langle \xi \rangle^{2s} \widehat{\nabla_{\mathcal{T}_\theta} \ell(\mathcal{T}_\theta \circ \mathcal{A}, g) \cdot \tfrac{\partial(\mathcal{T}_\theta \circ \mathcal{A})}{\partial \theta_i}}(\xi) \Big| d\xi \\ &\le \frac{1}{N} \sum_{j=0}^{N-1} \eta^{-2s} \Big\| \langle \xi \rangle^{2s} \widehat{\nabla_{\mathcal{T}_\theta} \ell(\mathcal{T}_\theta \circ \mathcal{A}, g) \cdot \tfrac{\partial(\mathcal{T}_\theta \circ \mathcal{A})}{\partial \theta_i}}(\xi) \Big\|_{L^\infty} \| e^{-\pi \varepsilon^2 |\xi|^2} \|_{L^1}. \end{aligned} \tag{5.16}$$

By a change of variables $\varepsilon \xi \to \xi$, we note

$$\| e^{-\pi \varepsilon^2 |\xi|^2} \|_{L^1} = \varepsilon^{-d} \int_{\mathbb{R}^d} e^{-\pi |\xi|^2} d\xi \le C \varepsilon^{-d}.$$

Hence, if we show that the L^∞-norm in Eq. 5.16 is finite, then

$$\left|\frac{\partial L_{\geq \eta}(\theta)}{\partial \theta_i}\right| \leq C\eta^{-2s}\varepsilon^{-d}.$$

Finally, if we take $\varepsilon = \min\{1/\sqrt[d]{N}, 1/d\}$ for large N, d, we conclude

$$\left|\frac{\partial L_{\geq \eta}(\theta)}{\partial \theta_i}\right| \leq C\max\{N, d^d\}\eta^{-2s} \tag{5.17}$$

as desired.

Now all we have to do is to bound the L^∞-norm in Eq. 5.16. Using the simple inequalities

$$\langle \xi \rangle \leq 1 + |\xi|, \quad (1+|\xi|)^M \leq C\sum_{|\alpha|\leq M}|\xi^\alpha|,$$

and Eq. 5.4, Eq. 5.1 in turn, we first see

$$\begin{aligned}
\left\|\langle\xi\rangle^{2s}\widehat{\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\cdot\tfrac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}}\right\|_{L^\infty} &\leq \left\|(1+|\xi|)^{2s}\widehat{\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\cdot\tfrac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}}\right\|_{L^\infty} \\
&\leq C\sum_{|\alpha|\leq 2s}\left\|\xi^\alpha\widehat{\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\cdot\tfrac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}}\right\|_{L^\infty} \\
&\leq C\sum_{|\alpha|\leq 2s}\left\|\widehat{D^\alpha\big(\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\cdot\tfrac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}\big)}\right\|_{L^\infty} \\
&\leq C\sum_{|\alpha|\leq 2s}\left\|D^\alpha\big(\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\cdot\frac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}\big)\right\|_{L^1}.
\end{aligned}$$

By Leibniz's rule we then bound the L^1-norm in the above as

$$\left\|D^\alpha\big(\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\cdot\frac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}\big)\right\|_{L^1} \leq C\sum_{\substack{|\alpha_1|+|\alpha_2|\\=|\alpha|}}\left\|D^{\alpha_1}\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\cdot D^{\alpha_2}\frac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}\right\|_{L^1}.$$

When $|\alpha| \leq s-1$, $|\alpha_1| \leq s$ and $|\alpha_2|+1 \leq s$, and then the L^1-norm in the right-hand side is generally finite since $\sigma \in W^{s,\infty}(\mathbb{R})$, $g \in W^{s,\infty}(\mathbb{R}^d)$, and the L^1-norm may be taken over the compact domain Ω; for example, $\ell(\mathcal{T}_\theta\circ\mathcal{A},g)(x) = |(\mathcal{T}_\theta\circ\mathcal{A})(x)-g(x)|^2$ and

$$\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)(x) = 2((\mathcal{T}_\theta\circ\mathcal{A})(x)-g(x))$$

for mean-squared error loss. Since $(\mathcal{T}_\theta\circ\mathcal{A})(x)$ is expressed as compositions of σ, the regularity of $\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)(x)$ is exactly determined by that of σ and g. Namely,

$$\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)(x) \in W^{s,\infty}(\mathbb{R}^d), \quad (\mathcal{T}_\theta\circ\mathcal{A})(x) \in W^{s,\infty}(\mathbb{R}^d), \tag{5.18}$$

from which the L^1-norm taken over the compact domain Ω is finite since $|\alpha_1| \leq s$ and $|\alpha_2|+1 \leq s$.

On the other hand, when $s \leq |\alpha| \leq 2s$ we set $|\alpha| = s+j$ with $0 \leq j \leq s$. Firstly, if $0 \leq |\alpha_1| \leq s$ (and so $j \leq |\alpha_2| \leq s+j$ since $|\alpha| = |\alpha_1|+|\alpha_2|$), then we bound

$$\left\|D^{\alpha_1}\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\cdot D^{\alpha_2}\frac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}\right\|_{L^1} \leq \left\|D^{\alpha_1}\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\right\|_{L^\infty}\left\|D^{\alpha_2}\frac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}\right\|_{L^1}.$$

Here, by Eq. 5.18, the L^∞-norm in the right-hand side is finite since $|\alpha_1| \leq s$, while the finiteness of L^1-norm follows from the fact that $D^\beta\sigma \in L^1(\mathbb{R})$ where $|\beta| = |\alpha_2|+1$. This fact is indeed valid for general activation functions such as ReLU, eLU, tanh and sigmoid since the L^1-norm may be taken over the compact domain Ω. Finally, if $s+1 \leq |\alpha_1| \leq s+j$ (and so $0 \leq |\alpha_2| \leq j-1$), then we bound this time

$$\left\|D^{\alpha_1}\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\cdot D^{\alpha_2}\frac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}\right\|_{L^1} \leq \left\|D^{\alpha_1}\nabla_{\mathcal{T}_\theta}\ell(\mathcal{T}_\theta\circ\mathcal{A},g)\right\|_{L^1}\left\|D^{\alpha_2}\frac{\partial(\mathcal{T}_\theta\circ\mathcal{A})}{\partial\theta_i}\right\|_{L^\infty}.$$

Here, the L^∞-norm in the right-hand side is finite by Eq. 5.18 since $|\alpha_2|+1 \leq j \leq s$. The finiteness of L^1-norm also comes from $D^\beta\sigma \in L^1(\mathbb{R})$ and $D^\beta g \in L^1(\mathbb{R}^d)$ with $s+1 \leq |\beta| \leq 2s$. Here, the condition $D^\beta\sigma \in L^1(\mathbb{R})$ is valid generally as above, and the bound Eq. 5.17 is still valid with η^{-s} even if the condition $D^\beta g \in L^1(\mathbb{R}^d)$ is not required. This is the case $j = 0$ in the proof.

References

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *Proc. the European Conference on Computer Vision (ECCV)*, 2020.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *Proc. the International Conference on Machine Learning (ICML)*, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proc. the International Conference on Machine Learning (ICML)*, 2020b.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Lawrence C. Evans. *Partial differential equations. Second edition. Graduate Studies in Mathematics*. American Mathematical Society, 2010.

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

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. 2018.

Camil Muscalu and Wilhelm Schlag. *Classical and multilinear harmonic analysis. Vol. I. Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 2013.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the ACM on Asia conference on computer and communications security*, 2017.

Zeyu Qin, Yanbo Fan, Hongyuan Zha, and Baoyuan Wu. Random noise defense against query-based black-box attacks. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Guided adversarial attack for evaluating and enhancing adversarial defenses. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Towards efficient and effective adversarial training. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Elias M. Stein and Guido Weiss. *Introduction to Fourier analysis on Euclidean spaces*. Princeton University Press, 1971.

Thomas H. Wolff. *Lectures on harmonic analysis*. American Mathematical Society, 2003.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *Proc. The International Conference on Learning Representations (ICLR)*, 2020.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yan Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 2020.

Table 3: Hyper-parameter setting for all baselines.

Method	Hyper-parameters	CIFAR-10 (PreActResNet-18)	ImageNet-100 (ResNet-18)
FBF	perturbation ϵ	0.031	0.031
	perturbation step size	0.039	0.039
	learning rate	0.1	-
	epoch	30	-
	batch size	256	-
GAT	perturbation ϵ	0.031	0.031
	perturbation step size	0.031	0.031
	learning rate	0.1	0.1
	epoch	100	100
	batch size	64	64
NuAT	perturbation ϵ	0.031	0.031
	perturbation step size	0.031	0.031
	learning rate	0.1	0.1
	epoch	100	100
	batch size	64	64
PGD	perturbation ϵ	0.031	0.031
	perturbation step size	0.039	0.039
	number of iterations	7	-
	learning rate	0.1	-
	epoch	30	-
	batch size	256	-
TRADES	perturbation ϵ	0.031	0.031
	perturbation step size	0.007	-
	beta	6.0	-
	learning rate	0.1	-
	epoch	100	-
	batch size	128	-
PhaseDNN	perturbation ϵ	0.031	0.031
	perturbation step size	0.039	0.039
	frequency range	[0, 50000)	[0, 50000)
	number of heads	3	3
	learning rate	0.1	0.1
	epoch	30	50
	batch size	256	128