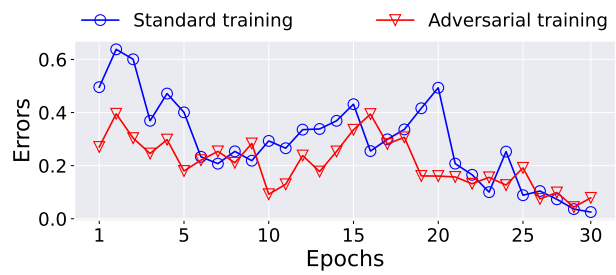
Phase-shifted Adversarial Training

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

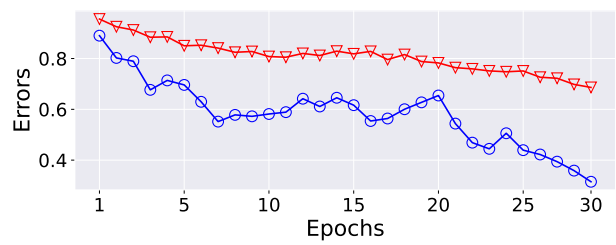
[1]Deargen Inc., Seoul, Republic of Korea
[2]School of Mathematics, Korea Institute for Advanced Study, Seoul, Republic of Korea
[3]Department of Mathematics, Sungkyunkwan University, Suwon, Republic of Korea
[4]Deargen USA Inc., Atlanta, GA

Abstract

Adversarial training (AT) has been considered an imperative component for safely deploying neural network-based applications. However, it typically comes with slow convergence and worse performance on clean samples (i.e., non-adversarial samples). In this work, we analyze the behavior of neural networks during learning with adversarial samples through the lens of response frequency. Interestingly, we observe that AT causes neural networks to converge slowly to high-frequency information, resulting in highly oscillatory predictions near each data point. To learn high-frequency content efficiently, we first prove that a universal phenomenon, the frequency principle (i.e., lower frequencies are learned first), still holds in AT. Building upon this theoretical foundation, we present a novel approach to AT, which we call phase-shifted adversarial training (PhaseAT). In PhaseAT, the high-frequency components, which are a contributing factor to slow convergence, are adaptively shifted into the low-frequency range where faster convergence occurs. For evaluation, we conduct extensive experiments on CIFAR-10 and ImageNet, using an adaptive attack that is carefully designed for reliable evaluation. Comprehensive results show that PhaseAT substantially improves convergence for high-frequency information, thereby leading to improved adversarial robustness.



(a) Low-frequency errors



(b) High-frequency errors

Figure 1: Errors of frequency components (high, low) between the training dataset and neural networks. Here, we use CIFAR-10 dataset for validation.

1 INTRODUCTION

Despite the remarkable success, deep neural networks are known to be susceptible to crafted imperceptible noise called *adversarial attacks* [Szegedy et al., 2013], which can have severe consequences when deployed in critical applications such as self-driving cars, medical diagnosis, and surveillance systems. In response to such negative implications, there has been a recent surge in research aimed at preventing adversarial attacks, such as adversarial training [Madry et al., 2018, Wong et al., 2020, Sriramanan et al., 2020, 2021], data augmentation [Gong et al., 2021, Wang et al., 2021, Rebuffi et al., 2021], and regularization [Qin et al., 2019]. Adversarial training is considered one of the most effective ways to achieve adversarial robustness. The early attempt generates the attack by only a single gradient descent on the given input, which is known as fast gradient sign method (FGSM) [Goodfellow et al., 2015]. However, it was later shown to be ineffective against strong multiple-step attacks [Kurakin

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

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

et al., 2016], e.g., projected gradient descent (PGD). Hence a myriad of defense strategies are introduced to build the robust models against the strong attacks by injecting the regularization to the perturbations [Sriramanan et al., 2020, 2021], randomly initializing the perturbations [Wong et al., 2020, Tramer and Boneh, 2019], and increasing the number of update steps for the perturbation to approximate the strong attack [Madry et al., 2018, Zhang et al., 2019].

Instead of introducing new attack and defense strategies, we analyze adversarial training through the lens of the frequency of the general mapping between inputs and outputs (e.g., neural networks). For this purpose, we calculate the errors of the frequency components between the dataset and neural networks to observe how the networks converge to the labeling function of the training dataset in terms of the frequencies. Compared to standard training (blue dots in Figure 1), we find that adding adversarial examples (red inverted triangles in Figure 1) causes the model to slowly converge in the high-frequency components (Figure 1(b)). [1] This indicates that adversarial robustness comes at significantly increased training time compared to standard training.

To learn high-frequency contents of the dataset efficiently, phase-shift deep neural networks (PhaseDNN) [Cai et al., 2020] is proposed based on a universal phenomenon of frequency principle (F-principle) [Xu et al., 2019, 2020, Rahaman et al., 2019, Luo et al., 2021], that is, *deep neural networks often fit target functions from low to high frequencies during the training*. PhaseDNN learns high-frequency contents by shifting these frequencies to the range of low-frequency to exploit the faster convergence in the low-frequencies of the DNN. However, it is challenging to apply PhaseDNN to adversarial training because PhaseDNN was optimized to solve a single-dimensional data (e.g., electromagnetic wave propagation, seismic waves).

In this work, we propose phase-shifted adversarial training (PhaseAT) to achieve adversarial robustness efficiently and effectively. To this end, we theoretically prove that the F-principle holds not only in standard training but also in adversarial training. We then extend the phaseDNN to adopt high-dimensional data and learn the adversarial data effectively. In summary, our contributions include the following:

- We conduct a pioneering analysis of adversarial training from the perspective of response frequency.
- We establish a mathematical foundation for how adversarial training behaves by considerably extending a universal phenomenon of frequency principle.
- We propose a new phase-shifted adversarial training algorithm based on our proposed theory, outperforming other strong baselines in many different settings.

[1] We use the filtering method [Xu et al., 2020] which explicitly splits the frequency spectrum into the high and low components and calculates the frequencies by applying the Fourier transform of the Gaussian function (please refer to Supplementary material for more detailed information).

2 BACKGROUND OF ADVERSARIAL TRAINING

Adversarial training is a method for learning networks that are robust to adversarial attacks. Given a network $\mathcal{T}$ parameterized by θ, a dataset $\{x_j, y_j\}_{j=0}^{N-1}$ where N is the size of dataset, a loss function ℓ and a threat model Δ, the learning problem can be cast to the following robust optimization problem.

$$\min_{\theta} \sum_{j=0}^{N-1} \max_{\delta \in \Delta} \ell(\mathcal{T}(x_j + \delta), y_j) \tag{2.1}$$

where δ is the adversarial perturbation. As we consider the adversarial robustness against L_∞ constrained adversaries, the threat model Δ takes the perturbation δ such that $\|\delta\|_\infty \leq \epsilon$ for some $\epsilon > 0$. For adversarial training, it is common to use adversarial attack to approximate the inner maximization over Δ, followed by some variation of gradient descent on the model parameters θ. For example, FGSM attack [Goodfellow et al., 2015] has the following approximation.

$$\delta = \epsilon \cdot \text{sign}(\nabla_x \ell(\mathcal{T}(x_j), y_j)) \tag{2.2}$$

3 F-PRINCIPLE IN ADVERSARIAL TRAINING

This section is devoted to theoretically demonstrating the F-principle in adversarial training as well as standard training (i.e., $\delta = 0$). We first represent the total loss $L(\theta)$ in the frequency domain and then quantify the rate of change of $L(\theta)$ contributed by high frequencies. We provide detailed proof in Supplementary material.

3.1 THE TOTAL LOSS IN THE FREQUENCY DOMAIN

Given a dataset $\{x_j, y_j\}_{j=0}^{N-1}$, $\mathcal{T}_\theta$ is the DNN output[2] and $g(x)$ is the target function (also known as labeling function) such that $g(x_j) = y_j$. Then the total loss L is generally defined by

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

Here, for example, $\ell(\mathcal{T}_\theta, g)(x) = \|\mathcal{T}_\theta(x) - g(x)\|^2$ for mean-squared error loss function, and for cross-entropy loss function $\ell(\mathcal{T}_\theta, g)(x) = -g(x) \cdot \log \mathcal{T}_\theta(x)$ where the log function

[2] Here we write $\mathcal{T}_\theta$ instead of $\mathcal{T}$ to explicitly denote the dependency on parameter θ.

acts on each vector component of $\mathcal{T}_\theta$. In adversarial training, we define an *adversarial function* $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^d$ by $\mathcal{A}(x) = x + \delta(x)$ with the adversarial perturbation δ and the corresponding output is given by $\mathcal{T}_\theta \circ \mathcal{A}$. In reality, it may be considered that $\mathcal{T}_\theta$ and g are bounded in a compact domain containing $\{x_j\}_{j=0}^{N-1}$. Then for the two common examples, $\ell(\mathcal{T}_\theta \circ \mathcal{A}, g)$ is absolutely integrable and $\ell(\mathcal{T}_\theta \circ \mathcal{A}, g)$ is differentiable with respect to the first argument. In this regard, these properties are considered to be possessed by a loss function generally.

Theorem 3.1. *Let $\widehat{f}$ denote the Fourier transform of f. Then, in the frequency domain*

$$L(\theta) = \lim_{\varepsilon \to 0} \frac{1}{N} \sum_{j=0}^{N-1} \int_{\mathbb{R}^d} e^{2\pi i x_j \cdot \xi} e^{-\pi \|\varepsilon \xi\|^2} \widehat{\ell(\mathcal{T}_\theta \circ \mathcal{A}, g)}(\xi) d\xi. \tag{3.1}$$

This representation is inspired by a convergence property, $G_\varepsilon * L(\theta) \to L(\theta)$, of convolution with approximate identities $G_\varepsilon(x) := \varepsilon^{-d} e^{-\pi \|\varepsilon^{-1} x\|^2}$. See Lemma D.1 in Supplementary material.

Now we split the integral in Eq. 3.1 into the region $\|\xi\| \leq \eta$ and $\|\xi\| \geq \eta$ for any $\eta > 0$ to separate the total loss into two parts $L_{\leq \eta}(\theta)$ and $L_{\geq \eta}(\theta)$ contributed by low and high frequencies, respectively:

$$L(\theta) = \lim_{\varepsilon \to 0} \big(L_{\leq \eta}(\theta) + L_{\geq \eta}(\theta) \big). \tag{3.2}$$

3.2 QUANTIFYING THE RATE OF CHANGE IN TOTAL LOSS

Let $W^{s,\infty}(\mathbb{R}^d)$ denote the Sobolev spaces[3] which consist of bounded functions having bounded derivatives up to a given order s, and thus the index s reflects the degree of regularity.

Theorem 3.2. *Consider a DNN with multi-layer in adversarial training. If an activation function σ and the target function g satisfy $\sigma \in W^{s,\infty}(\mathbb{R})$ and $g \in W^{2s,\infty}(\mathbb{R}^d)$ for some $s \in \mathbb{N}$, then there exists a constant $C > 0$ independent of s, η such that for any $\eta > 0$*

$$\begin{aligned} &\|\nabla_\theta L(\theta) - \nabla_\theta L_{\leq \eta}(\theta)\| \\ &\qquad \approx \|\nabla_\theta L_{\geq \eta}(\theta)\| \leq C \max\{N, d^d\} \eta^{-2s}. \end{aligned} \tag{3.3}$$

Theorem 3.2 implies that the rate of decrease in the corresponding high-frequency region loss ($L_{\geq \eta}(\theta)$) is much greater than the rate of increase in η. In other words, a model tends to fit target functions from low to high frequency information during the adversarial training. The mathematical insight of this theorem is that the regularity of a network converts into the decay of the total loss in the frequency domain. Specifically, a network with a more regular activation function (bigger s) converges more slowly, according to η^{-2s} in Eq. 3.3. As the frequency increases, this becomes more evident. For example, if σ is ReLU or eLu then $s = 1$ or $s = 2$, respectively. For $\tanh$ and sigmoid activation functions, s may be taken arbitrarily large.

The approximation in Eq. 3.3 becomes more accurate as ε diminishes in Eq. 3.2, and $\varepsilon = \min\{1/\sqrt[d]{N}, 1/d\}$ is inversely proportional to the size of dataset (N) or the dimension of input data (d). Therefore, when N or d are large, ε decreases. This is a common phenomenon in real-world datasets, which are typically high-dimensional and contain a large number of data samples (e.g., images and languages).

Our innovative theory differs from its predecessors [Xu et al., 2020, Luo et al., 2021] in two ways. First, we generalize F-principle by showing that it holds for the cross-entropy loss. Second, we provide a faster decay rate with $2s$ in Eq. 3.3, which serves as one of the motivations for using phase-shifting, particularly in adversarial training. Finally, we provide the mathematical justification for the F-principle in adversarial training settings.

4 PHASE-SHIFTED ADVERSARIAL TRAINING

In this section, we detail the proposed method, coined phase-shifted adversarial training (PhaseAT). We first present the existing PhaseDNN [Cai et al., 2020] and its limitations (Section 4.1). We then redesign the original PhaseDNN to make it more practical and suitable for adversarial training (Section 4.2). Finally, we elaborate PhaseAT by optimizing PhaseDNN through adversarial training (Section 4.3).

4.1 PHASE SHIFT DEEP NEURAL NETWORKS

To learn highly oscillatory functions efficiently, [Cai et al., 2020] propose PhaseDNN which shifts the high-frequency components of the dataset to the range of the low-frequency range for fast convergence because according to F-principle, neural networks learn low-frequency information first before learning high-frequency components. On the phase-shifted dataset, the neural networks learn the target function with low-frequency contents. Unfortunately, this requires that the frequency extraction of the original training data has to be done numerically using convolutions with a frequency selection kernel, which requires a huge memory footprint, i.e., a storage of O(N × N). Alternatively, the neural networks can be learned on the data from all ranges of frequencies while the phase-shift is included in the makeup of the PhaseDNN. This version of PhaseDNN is called *coupled PhaseDNN*, and we adopt this version to avoid the cost of decomposing the frequency components on the original dataset.

Since the phase-shift is performed on the networks rather

[3]For a vector-valued f, we also write $f \in W^{s,\infty}(\mathbb{R}^d)$ to represent all of its component functions are contained in the space.

than the dataset, PhaseDNN consists of an ensemble of networks, each of which is dedicated to a specific frequency. To learn higher frequency components, phase-shift is applied to each network separately. The output of PhaseDNN can be represented as follows:

$$\mathcal{T}(x) = \sum_{m=0}^{M-1} e^{i\omega_m x} \cdot \mathcal{T}_m(x) \tag{4.1}$$

where M is the size of ensemble, $T_m(x)$ represents one of the networks in the ensemble, and ω_m is the specific frequency for the networks T_m. Let the labeling function of the dataset be $g(\cdot)$, the phaseDNN is optimized by following the least square errors:

$$\sum_{j=0}^{N-1} |g(x_j) - \mathcal{T}(x_j)|^2 = \sum_{j=0}^{N-1} \left| g(x_j) - \sum_{m=0}^{M-1} e^{i\omega_m x_j} \cdot \mathcal{T}_m(x_j) \right|^2 \tag{4.2}$$

Note that we always include the lowest frequency in the selected frequency, i.e., $\omega_0 = 0$, because the low frequencies typically dominates in real datasets [Xu et al., 2020].

4.2 MULTI-HEADED PHASEDNN

Applying the previous PhaseDNN to adversarial training has a few challenges. First, PhaseDNN was designed to learn single-dimensional data; the number of all possible frequencies grows exponentially with the dimension of the input, which prevents the use of high-dimensional inputs (e.g., images and languages). Second, PhaseDNN requires multiple networks to perform the phase-shift of the different frequencies, causing a large memory footprint.

To address the first challenge, we project the given inputs to the first principal component and use the projected scalar values to represent the original data. The forward propagation of PhaseDNN (Eq. 4.1) is reformulated as:

$$\mathcal{T}(x_j) = \sum_{m=0}^{M-1} e^{i\omega_m (x_j \cdot p)} \cdot \mathcal{T}_m(x_j) \tag{4.3}$$

where p is the first principal component of the input space calculated on the training dataset. Instead of observing all frequencies of high-dimensional data, we focus on the frequencies of the data along the first principal component p. Additionally, we bound the product result by normalizing the two vectors (i.e., x and p) and multiplying the constant C to fix the range of data points.

Lastly, we introduce a multi-headed PhaseDNN to avoid using the ensemble, which consumes large computational resources. We make each network $\mathcal{T}_m$ share the feature extracting networks and has frequency-dedicated networks for predictions. Thus, Eq 4.3 is reformulated as follows:

$$\mathcal{T}(x_j) = \sum_{m=0}^{M-1} e^{i\omega_m (x_j \cdot p)} \cdot \mathcal{H}_m(\mathcal{F}(x_j)) \tag{4.4}$$

where $\mathcal{F}(\cdot)$, $\mathcal{H}_m(\cdot)$ are the shared networks and the m-th frequency-dedicated classifier, respectively. It is worth noting that the classifiers $\mathcal{H}$ only make up a small portion of the total parameters, allowing PhaseDNN to efficiently learn highly oscillatory functions with a lower memory footprint.

4.3 PHASEDNN WITH ADVERSARIAL TRAINING

PhaseDNN requires shift frequencies ($\{\omega_m\}_{m=0}^{M-1}$) for each head ($\{\mathcal{H}_m\}_{m=0}^{M-1}$) used during training. The following explains how to choose those frequencies. As the goal of training a network is to minimize the differences between a clean ($\{x_j\}_{j=0}^{N-1}$) and an adversarial version ($\{x_j = x_j + \delta\}_{j=0}^{N-1}$) of the same data point, we choose the target frequencies with the largest difference in Fourier coefficients between the two. In practice, estimating the Fourier coefficients of the total dataset in every optimization step requires huge computational resources; therefore, we approximate this by estimating the exponential moving average of the coefficients on mini batches. The discrepancy of the frequency k between clean and adversarial batches is determined as follows:

$$d_k = |\mathcal{F}_k(X) - \mathcal{F}_k(X + \Delta)| \tag{4.5}$$

where X and Δ indicate the batched data and its corresponding perturbations, respectively, $\mathcal{F}_k(\cdot)$ is the Fourier coefficient which is obtained as follows:

$$\mathcal{F}_k(X) = \sum_{j=0}^{B-1} \mathcal{T}(X_j) \cdot e^{-2\pi i k (X_j \cdot p)} \tag{4.6}$$

where B is the batch size. The estimated discrepancy d_k for all frequencies is then used to derive the multinomial distribution to sample the frequencies ω_m for each head of the phaseDNN[4]. The reason for sampling frequencies from a multinomial distribution is that the training dataset is constantly changing during adversarial training. In this case, a fixed set of frequencies (e.g., peaks of frequencies used in [Xu et al., 2020]) does not accurately reflect the critical frequency information. As a result, by stochastically learning the frequency differences, the model could decrease the prediction differences between clean and adversarial data.

Attack generation. With the practically-designed PhaseDNN, we perform adversarial training, which causes the neural networks to highly oscillate near the training data. Motivated by the recent finding that FGSM-based training can sufficiently prevent strong attacks [Wong et al., 2020], we train PhaseDNN to be robust against FGSM attacks.

The FGSM attack is formulated by initializing the perturbation on uniform distribution ranging ($-\epsilon$, ϵ). The perturbation is then updated based on the sign of the gradients

[4]Similar to the previous work, we always include the zero frequency for the one of the head networks.

Algorithm 1 Phase-shifted Adversarial Training

Require: Training epochs T, dataset $\mathcal{D}$, dataset size N, perturbation size ϵ, perturbation step α, networks $\mathcal{T}$

for $t = 1 \ldots T$ **do**
  for each Batch $(X, Y) \sim \mathcal{D}$ **do**
    $\Delta = \text{Uniform}(-\epsilon, \epsilon)$
    // Alternate training across mini-batches
    if j % 2 == 0 **then**
      $\Delta = \Delta + \alpha \cdot \text{sign}(\nabla_{\Delta} \ell_{\text{ce}}(\mathcal{T}(X + \Delta), Y))$
    else
      $\Delta = \Delta + \alpha \cdot \text{sign}(\nabla_{\Delta} \ell_{\text{ce}}(\mathcal{T}_0(X + \Delta), Y))$
    end if
    $\Delta = \max(\min(\Delta, \epsilon), -\epsilon)$
    // Sampling frequencies for $\mathcal{T}$ (Eq. (4.5), (4.6))
    $d = |\mathcal{F}(X) - \mathcal{F}(X + \Delta)|$
    $\{\omega_m\}_{m=1}^{M} \sim \text{Multinomial}(d)$
    // Optimize networks based on loss (Eq. (4.7)
    $\theta = \theta - \nabla_{\theta}[\ell_{\text{adv}}(X, Y))]$
  end for
end for

of the cross-entropy with the ground-truth labels. Since PhaseDNN selects the frequencies in a stochastic manner, the generated attack can encourage PhaseDNN to better fit the diverse components having high-frequencies (detailed in Section 5.2). Moreover, we perform the phase-shift in alternate mini-batches which further diversifies the attacks used for adversarial training similar to the previous studies [Sriramanan et al., 2020, 2021].

Adversarial training. We improve the model's robustness against generated attacks by replacing the clean inputs with perturbed ones during training. Additional improvements can be achieved by regularization to minimize the effects of potential white-box attacks. One possible white-box attack would set all shift frequencies to zero, resulting in gradients similar to normal AT rather than PhaseAT. Our regularization term encourages the model to behave differently than the standard AT. We implement this by minimizing the prediction similarity between the phase-shifted model and the model that does not have the phase-shift. In summary, the total objective function is the sum of the cross-entropy loss and the regularization.

$$\ell_{adv}(x, y) = \ell_{ce}(\mathcal{T}(x + \delta), y) + \left| \frac{\sigma(\mathcal{T}(x + \delta)) \cdot \sigma(\mathcal{T}_0(x + \delta))}{\|\sigma(\mathcal{T}(x + \delta))\| \|\sigma(\mathcal{T}_0(x + \delta))\|} \right| \tag{4.7}$$

where ℓ_{ce} is the cross entropy function, σ indicates softmax function, $\mathcal{T}_0$ indicates the model without the phase-shift term (i.e., $\omega_m = 0, \forall\ m \in [1, M]$). By encouraging the model to have different predictions than normal AT, the model achieves robustness against the normal AT attack as well [5] The proposed defense is detailed in Algorithm-1.

5 EVALUATIONS

5.1 EXPERIMENTAL SETUP

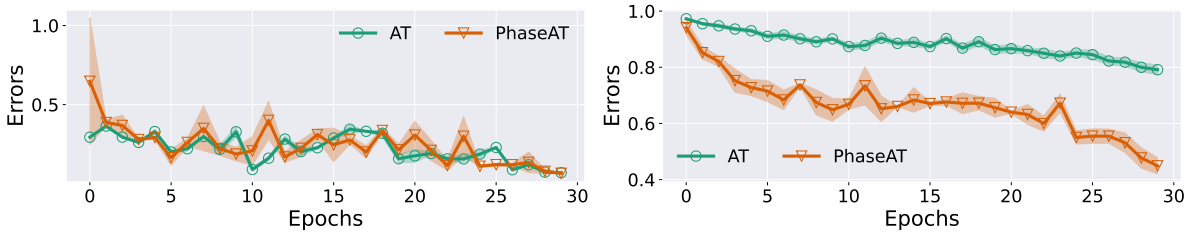
Baselines. We compare our method with both non-iterative and iterative methods. Non-iterative methods updates the perturbation once to generate the attacks, whereas iterative methods generate the perturbations through multiple optimization steps. We choose three recent non-iterative methods for comparison, namely FBF [Wong et al., 2020], GAT[Sriramanan et al., 2020], and NuAT [Sriramanan et al., 2021]. For iterative methods, we select four representative baselines, i.e., PGD [Madry et al., 2018], TRADES [Zhang et al., 2019], GAIRAT [Zhang et al., 2020], and AWP [Wu et al., 2020]. We carefully set the hyper-parameters for the aforementioned methods. We use ResNet-18 [He et al., 2016] and WideResNet-34-10 [Zagoruyko and Komodakis, 2016] architectures [He et al., 2016] for the evaluations and, the detailed settings are listed in Supplementary material.

Datasets and Evaluations. Following recent works, we perform experiments on two datasets: CIFAR-10 [Krizhevsky et al., 2009] and ImageNet [Deng et al., 2009]. To compare with the recent works, we use ImageNet-100 [Sriramanan et al., 2020, 2021] which is the small version of ImageNet including 100 classes. To evaluate adversarial robustness, we consider both white-box and black-box attacks. For white-box attacks, PGD and AutoAttack (AA) are mainly used, each of which contains L_∞ adversarial constraint of $\epsilon = 8/255$. Specifically, the AA attacks involve APGD$_{ce}$, APGD$_{dlr}$, APGD$_t$ [Croce and Hein, 2020b] and FAB$_t$ [Croce and Hein, 2020a][6]. The black-box attacks include transfer-based and query-based attacks. For the transfer-based attack, we generate PGD-7 perturbations using other networks (e.g., VGG-11 and ResNet-18 that is differently initialized) to attack each baseline. For query-based attacks, we use square attacks [Andriushchenko et al., 2020] with 5,000 query budgets. The results of black-box attacks are included in Supplementary material.

Consideration of Adaptive Attack. Compared to previous methods, PhaseAT includes stochastic process to learn diverse frequencies of the datasets, which prevents reliable evaluation of the adversarial robustness. Following the guideline [Tramèr et al., 2018], we carefully design the adaptive attack for our method. First, to circumvent the stochastic process, the adaptive attack involves the expectation over transformation (EOT) [Athalye et al., 2018] to approximate the true gradient of the inference model. Sec-

[5] We provide the ablation study about the regularization term in Supplementary material.

[6] We include the details of each attack in Supplementary material.



(a) Low-frequency errors

(b) High-frequency errors

Figure 2: Errors of frequency components (high, low) between the training dataset and neural networks for normal adversarial training method and PhaseAT.

Table 1: Performance evaluation of the different designs of adaptive attacks on CIFAR-10 dataset.

Attack	Accuracy
PGD_{50}	61.5
PGD_{50} + EOT	60.9
PGD_{50} + Frequency	59.7
PGD_{50} + EOT + Frequency	59.5

ond, we assume that the adversary has full knowledge of the learning algorithm. Hence the adversary could mimic the strategy of frequency selection used in the training/inference phase, i.e., multinomial distribution of the discrepancy. We tabulate how these adaptive attacks affect our method in Table 1.

This shows that reducing the effect of the stochastic properties by EOT decreases the performance of our method. Particularly, mimicking the frequency selection strategy further reduces the performance. However, note that PhaseAT still reveals strong adversarial robustness even though the adaptive attacks are applied (refer to main paper), which requires ten times more costs to generate the attacks compared to other baselines. Such adaptive attacks are applied to our method for a fair comparison.

5.2 MAIN RESULTS AGAINST WHITE-BOX ATTACKS

Before comparing the performance of all methods, we first verify whether our method learns high-frequency components faster than the model without the phase-shift. To that end, we use the filtering method [Xu et al., 2020] used in previous analysis (Figure 1). Figure 2 shows the errors for each frequency. Here, we use the settings of CIFAR-10. For the low-frequency part, the errors between PhaseAT and AT are negligible, whereas the errors of the high-frequency between two methods is noticeable, demonstrating our hypothesis that PhaseAT can efficiently learn high-frequency components.

We then confirm that fast learning of high-frequency components leads to better robustness. Table 2 shows the comparison results. Here, we use the two different networks, ResNet-18 and WideResNet-34-10, to verify the generality of PhaseAT. Non-iterative methods tend to show lower accuracy than iterative methods. However, the non-iterative version of PhaseAT outperforms iterative baselines in terms of both standard and robust accuracy. For example, PhaseAT shows 5.3% and 8.5% performance improvement over AWP in terms of standard and PGD accuracy, respectively, and comparable performance for AA. In the experimental results, PhaseAT outperforms FBF, the major distinction of which is the phase-shift functionality. This suggests that learning high-frequency information properly is particularly beneficial for learning adversarial data.

We turn our focus to a larger dataset, i.e., ImageNet. We compare our method with all non-iterative methods and tabulate the results in Table 3. We again find a similar trend in performance to that of CIFAR-10 except for the strongest baseline is the non-iterative method (i.e., NuAT). While the proposed method shows the comparable performance for clean accuracy, it outperforms others by a large margin in terms of the robust accuracy against the AA attack. This shows that PhaseAT can be scalable to larger datasets.

5.3 CONVERGENCE OF CLEAN AND ROBUST ACCURACY

We verify that faster learning on high-frequency information leads to better convergence for robust accuracy. To that end, we report the standard and AA accuracy during each training epoch. We use the same settings of CIFAR-10 and tabulate the results on Figure 3. Compared to the normal adversarial training (denoted as AT), PhaseAT shows faster and better convergence both on standard and robust accuracy. This indicates that the training model has smoothed predictions near the data, effectively reducing the oscillating property of predictions when adding adversarial perturbations.

Table 2: Performance evaluation on CIFAR-10 dataset against white-box attacks on two different architectures. Best and second best results are highlighted in boldface and underline, respectively.

Method	Clean accuracy	PGD_{50+EOT}	AA_{+EOT}
Normal	93.1	0.0	0.0
FBF	<u>84.0</u>	43.8	41.0
GAT	80.5	53.2	47.4
NuAT	81.6	52.0	48.3
PhaseAT (ours)	**86.2**	**59.5**	**52.1**
PGD-AT	81.3	50.8	47.3
TRADES	79.5	52.2	48.5
GAIRAT	82.7	<u>56.3</u>	34.5
AWP	81.8	54.8	<u>51.1</u>

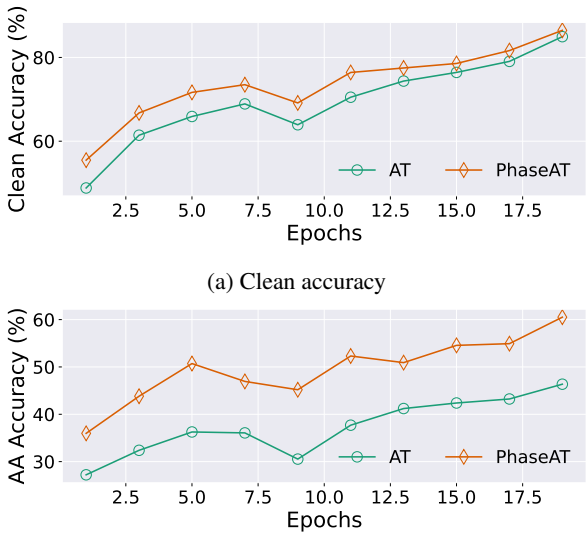
(a) ResNet-18

Method	Clean accuracy	PGD_{50+EOT}	AA_{+EOT}
Normal	94.5	0.0	0.0
FBF	82.1	54.4	51.3
GAT	84.7	56.1	52.1
NuAT	85.1	54.6	53.1
PhaseAT (ours)	**88.8**	**62.3**	**59.2**
PGD-AT	86.7	56.8	54.7
TRADES	<u>87.6</u>	55.9	53.9
GAIRAT	85.3	57.6	42.6
AWP	86.9	<u>60.4</u>	<u>56.5</u>

(b) WideResNet-34-10

Table 3: Performance evaluation on ImageNet-100 dataset against white-box attacks. Best and second best results are highlighted in boldface and underline, respectively. The results for each baseline come from the previous works [Sriramanan et al., 2020, 2021].

Method	Clean accuracy	AA_{+EOT}
Normal	77.1	0.0
FBF	**70.5**	20.4
GAT	68.0	28.9
NuAT	69.0	32.0
PhaseAT (ours)	<u>69.2</u>	**35.6**
PGD-AT	68.6	<u>33.0</u>
TRADES	62.9	31.7
AWP	64.8	29.2



(a) Clean accuracy

(b) Robustness Accuracy

Figure 3: Clean and robust accuracy during the training.

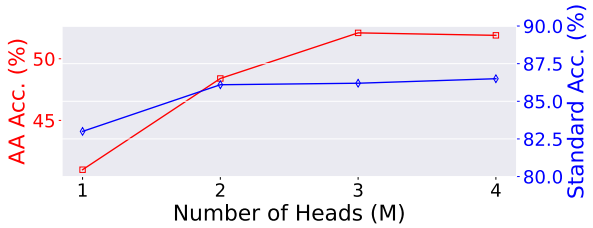
5.4 SENSITIVITY TO THE DIFFERENT SETTINGS OF PHASEAT

We analyze the sensitivity of PhaseAT on different settings. Since the most significant components of PhaseAT are the head and frequency, we mainly control the number of heads (i.e., M) and the range of frequency. Figure 4 shows how the change of each parameter affects the standard and robust accuracy on CIFAR-10. We first see that utilizing more heads leads to the improved accuracy while incurring additional computation costs. In addition, the robust accuracy is more sensitive to the different number of heads, whereas the clean accuracy does not differ significantly when more than two heads are used. When it comes to the frequency range, we observe that utilizing the wider range of the frequency leads to the improved robust accuracy. For the clean accuracy case, on the other hand, the wider range has no significant impact, which is consistent with the previous finding that low-frequency typically dominates in the real dataset [Xu et al., 2020]. In the experiments, we use the three heads and 50k of the frequency range based on the above validation results. For the frequency range case, no further improvement was observed for more than the 50k range.

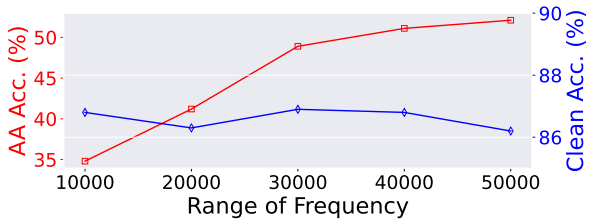
5.5 COMPUTATIONAL COMPLEXITY

We compare the training time required to achieve the best robust accuracy for each method. To fairly measure it, we use the optimized schedule in the implementations (NuAT[7],

[7] https://github.com/locuslab/fast_adversarial.



(a) Sensitivity to head



(b) Sensitivity to frequency

Figure 4: Sensitivity to the different number of heads and the range of frequency for PhaseAT.

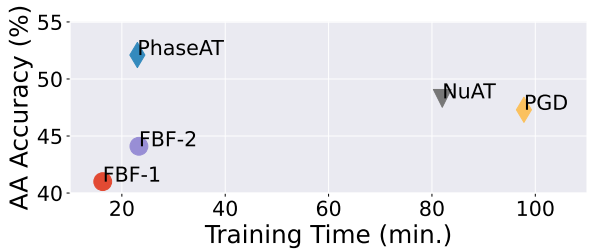


Figure 5: Comparison of the training time of each method to achieve the best robust accuracy.

FBF and PGD[8]). Figure 5 plots the comparison on CIFAR-10. The fastest defense amongst all is FBF-1 and PhaseAT shows the comparable training time. This indicates the additional time required by Fourier transform and forward-propagation to multiple heads is negligibly small. We also compare the different versions of FBF, namely FBF-2, which uses more training epochs than FBF-1 to match the training time with PhaseAT. Despite FBF's increased accuracy (41.0 to 44.1%), PhaseAT outperforms it by a significant margin, demonstrating the efficacy of the proposed approach.

6 RELATED WORK

6.1 ADVERSARIAL TRAINING

One of the most effective defenses against adversarial attacks is adversarial training. Specifically, iteratively updating the attacks during training tends to show better robustness as the adversary typically performs multiple updates to generate stronger attacks [Madry et al., 2018, Zhang et al., 2019, Wu et al., 2020]. However, the adversarial robustness comes at a large computational cost and highly increased training time due to the multiple optimizations. Hence the research towards non-iterative methods has been getting attention [Wong et al., 2020, Sriramanan et al., 2020, 2021]. The common strategy is to make the FGSM-based training robust against iterative attacks. For example, [Wong et al., 2020] discover that training with FGSM-based training can be a stronger defense against iterative attacks. This FGSM can be achieved by initializing the perturbation as uniform distribution with properly adjusted perturbation and step sizes. Instead of the initialization, [Sriramanan et al., 2020] propose a non-iterative adversarial training method by introducing L_2 regularization for smoothing the loss surface initially and further reducing the weight of this term gradually over iterations for better optimization. Similarly, [Sriramanan et al., 2021] enhance the above relaxation term by involving the use of nuclear norm to generate a more reliable gradient direction by smoothing the loss surface.

6.2 F-PRINCIPLE IN STANDARD TRAINING

The F-principle was first discovered empirically in [Rahaman et al., 2019] and [Xu et al., 2019] independently. A theoretical proof was given by [Xu et al., 2020] only for a one-hidden layer $\tanh$ network with a mean-squared error loss function. It was later extended in [Luo et al., 2021] to multi-layer networks with general activation functions and general loss functions excluding cross-entropy loss function.

7 CONCLUSION

In this work, we have analyzed the behavior of adversarial training through the lens of the response frequency and have empirically observed that adversarial training causes neural networks to have slow convergence on high-frequency components. To recover such slow convergence, we first prove that F-principle still holds in adversarial training involving more generalized settings. On this basis, we propose phase-shifted adversarial training, denoted as PhaseAT, in which the model learns high-frequency contents by shifting them to low-frequency ranges. In our method, we redesign the PhaseDNN to adopt high-dimensional inputs and effectively learn adversarial data. For reliable evaluation, we carefully design adaptive attacks and evaluate the proposed method on two popular datasets, namely CIFAR-10 and ImageNet. The results clearly show that PhaseAT shows strong performance compared to other baselines, demonstrating the efficacy of faster high-frequency learning in adversarial training. While PhaseAT enables faster training on adversarial examples, the pre-processing phase to derive the first principal component can be overhead as the number of datasets increases. In future work, we will explore the way to derive the first principle component efficiently.

[8] https://github.com/val-iisc/NuAT

ACKNOWLEDGMENT

This work was supported by the Center for Advanced Computation and a KIAS Individual Grant (MG082901) at Korea Institute for Advanced Study and the POSCO Science Fellowship of POSCO TJ Park Foundation (S. Kim), and by NRF-2022R1A2C1011312 (I. Seo).

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.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proc. the International Conference on Machine Learning (ICML)*, 2018.

Wei Cai, Xiaoguang Li, and Lizuo Liu. A phase shift deep neural network for high frequency approximation and wave problems. *SIAM Journal on Scientific Computing*, 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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Chengyue Gong, Tongzheng Ren, Mao Ye, and Qiang Liu. Maxup: Lightweight adversarial training with data augmentation improves neural network training. In *Proc. the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proc. The International Conference on Learning Representations (ICLR)*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *Proc. The International Conference on Learning Representations (ICLR)*, 2016.

Tao Luo, Zheng Ma, Zhi-Qin John Xu, and Yaoyu Zhang. Theory of the frequency principle for general deep neural networks. *CSIAM Transactions on Applied Mathematics*, 2021.

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

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proc. the International Conference on Machine Learning (ICML)*, 2019.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. 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.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proc. The International Conference on Learning Representations (ICLR)*, 2013.

Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *Proc. The International Conference on Learning Representations (ICLR)*, 2018.

Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. 2020.

Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *International Conference on Neural Information Processing*, 2019.

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

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proc. the International Conference on Machine Learning (ICML)*, 2019.

Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *Proc. The International Conference on Learning Representations (ICLR)*, 2020.