Towards Building More Robust Models with Frequency Bias

Qingwen Bu[1,3]   Dong Huang[2*]   Heming Cui[1,2]

[1]Shanghai Artificial Intelligence Laboratory   [2]The University of Hong Kong
[3]Shanghai Jiao Tong University

qwbu01@sjtu.edu.cn, dhuang@cs.hku.hk, heming@cs.hku.hk

Abstract

The vulnerability of deep neural networks to adversarial samples has been a major impediment to their broad applications, despite their success in various fields. Recently, some works suggested that adversarially-trained models emphasize the importance of low-frequency information to achieve higher robustness. While several attempts have been made to leverage this frequency characteristic, they have all faced the issue that applying low-pass filters directly to input images leads to irreversible loss of discriminative information and poor generalizability to datasets with distinct frequency features. This paper presents a plug-and-play module called the Frequency Preference Control Module that adaptively reconfigures the low- and high-frequency components of intermediate feature representations, providing better utilization of frequency in robust learning. Empirical studies show that our proposed module can be easily incorporated into any adversarial training framework, further improving model robustness across different architectures and datasets. Additionally, experiments were conducted to examine how the frequency bias of robust models impacts the adversarial training process and its final robustness, revealing interesting insights.

1. Introduction

As deep learning methods are making a splash in various fields, their security and robustness are drawing more and more attention from both academia and industry. DNNs for image classification have been proven to be easily fooled by adversarial examples, with imperceptible perturbations added to natural images. The vulnerability is also uncovered in many other safety-critical fields like medical diagnosis [10] and autonomous driving [8]. Adversarial training is regarded as a trustworthy approach for producing robust models, and considerable subsequent effort has been done to increase its efficiency [28, 33] and efficacy [40, 41]. De-

spite continuously advanced robust learning methods, the relationship between some intrinsic characteristics of the model structure, such as frequency characteristics, and the robustness it exhibits seems to be rarely discussed.

The emergence of vision transformers (ViTs) revealed a novel architecture that performs on par or even better than convolutional networks (CNNs) in many vision tasks. As more and more in-depth work unfolds, some of ViT's fundamentals and characteristics are exposed. Several recent works try to improve the performance of ViTs with a frequency lens. [29] proposed naturally trained ViTs are more robust to adversarial attack, especially for high-frequency perturbations. It is also noticed that ViTs reduce high-frequency signals while CNNs amplify them [25]. [31] successfully enhances ViT performance by deliberately retaining high-frequency information. The complementarity of CNNs and ViTs exhibited in the frequency domain is an intriguing starting point for studying and further improving the robustness of the model. But the relations between robustness and the frequency bias of models have not been fully studied and utilized in the context of robust learning.

Research [11] has suggested that deep neural networks, when trained on image classification data sets, tend to exhibit bias towards texture information, which is the high-frequency element present in images. By contrast, models that are adversarially trained primarily emphasize the importance of low-frequency information, leading to improved robustness [42]. To delve further into this phenomenon, several studies have experimented with adjusting the constraints placed on either low or high-frequency signals in the loss regularization term [2], or incorporating perturbations of variable frequencies during adversarial training [23]. There are also several attempts [43, 16] aimed at exploiting the above mentioned frequency properties to build more robust models. However, they have primarily been limited to applying low-pass filtering directly to clean or adversarial inputs. Regrettably, this methodology is associated with an irreversible loss of high-frequency information within the image and thus causes a marked decrease in accuracy across clean samples (e.g., 4-5% clean accuracy

---

*Corresponding author.

drop compared to standard AT methods) [16]. Furthermore, it is imperative to adapt the parameters of the low-pass filter to better suit different datasets that embody distinct frequency characteristics.

In order to solve the aforementioned problems and to adopt a new perspective to study the frequency characteristics of robust models, we propose a plug-and-play module called Frequency Preference Control Module (FPCM) to reconfigure the low-frequency and high-frequency components of the intermediate features learned within the model, which can be simply cooperated with any adversarial training framework and further boost model's robustness. While previous methods require special tuning of their introduced hyperparameters to adapt to datasets with various frequency characteristics, our module can be co-optimized with the model in adversarial training with no need for any fine-grained hyperparameter adjustments, which also allows it to adapt and then utilize frequency features exhibited in different stages (i.e., layers) of a model. Our empirical study reveals that our approach demonstrates adaptability across different model architectures and datasets and continuously drives robust accuracy improvement. Moreover, with FPCM, we can pioneer the study of the frequency characteristics of intermediate-level feature representations and take a closer look at how the frequency bias of a robust model would impact its robustness.

In a nutshell, our contribution can be folded into following aspects:

1. We proposed the Frequency Preference Control Module that can be effortlessly incorporated into any AT approach and further improve the model robustness.

2. Leveraging our proposed module, we show how the frequency of feature representation would impact the final robustness and adversarial training process.

2. Related Work

2.1. Adversarial Attack

Szegedy *et al*. [30] found the vulnerability of DNNs against adversarial examples and proposed L-BGFS based attack. Subsequently, Goodfellow *et al*. [12] argued that the main reason for the vulnerability of neural networks to adversarial perturbations is their linear nature, and used Fast Sign Gradient Method (FGSM) to generate adversarial examples efficiently. Kurakin *et al*. [21] extended FGSM to a more effective iteration-based attack as Basic Iterative Method (BIM). Subsequently, by adding random initialization and extending to more iterative steps, Madry *et al*. [22] propose PGD as a universal first-order adversary. Boundary-based attack DeepFool [24] and optimization-based attacks like C&W [4].

In addition to the thriving white-box attack methods based on the assumption of model transparency, researchers are also especially interested in black-box attacks, where the structure and parameters of the DNN are unknown to the attackers. (i.e., only a few queries or training data are available). There are two types of black-box attacks: query-based and transfer-based. To produce adversarial instances, query-based algorithms estimate the gradient information via queries [3, 17, 1]. While transfer-based attacks [9, 35, 27] employ white-box attacks on a local surrogate model to produce transferable adversarial perturbations.

AutoAttack [6] is a combination of four complementary attack methods (i.e., APGD-CE, APGD-DLR, FAB [5], and Square Attack [1], which has become a very popular adversarial robustness benchmark today.

2.2. Adversarial Training as a Defense

Adversarial training is one of the notable defense methods as an empirical defense, which is also adopted in this paper to build robust models. And its basic idea can be expressed as a min-max optimization problem:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\in\mathcal{X}} \left[\max_{\delta\in[-\epsilon,\epsilon]} \ell(f_\theta(x+\delta); y) \right]$$

where the f_θ is a DNN parameterized by θ, $\mathcal{X}$ stands for the training dataset and ℓ represents the loss function. And perturbations δ are bounded into the ϵ-ball.

To solve the inner maximization problem, projected gradient descent (PGD) [22] is a prevailing and effective method to generate perturbations. Many promising methods then sprang up. Some of them provide more efficient loss function designs [39, 32], some methods design stronger regularization methods [34, 26], and some focus on solving the problem of the excessive computational overhead of adversarial training [28, 33].

2.3. Frequency Analysis of ViTs and CNNs

In engineering applications, the vast majority of Fourier transform applications use the discrete Fourier transform (DFT), where the Fourier transform takes a discrete form in both the time and frequency domains. Applying DFT to a flattened image signal x is equal to left multiplying a DFT matrix, the rows of which are the Fourier basis $f_k = [e^{2\pi j(k-1)\cdot 0}, ..., e^{2\pi j(k-1)\cdot(n-1)}]/\sqrt{n} \in \mathbb{R}^n$, where k represents the kth row of the DFT matrix and j represents the imaginary unit. The Fast Fourier Transform (FFT) is usually used in practical applications to efficiently compute the DFT.

Recent work [31, 25] studied and revealed the low-pass characteristic of basic ViT blocks. ViT tends to reduce high-frequency signals in the feature map and thus almost

$B \times C \times H \times W$

low-freq feature
$B \times WH \times C$

weights
$B \times 1 \times C$

IFFT

Low-pass Filter

Conv1d
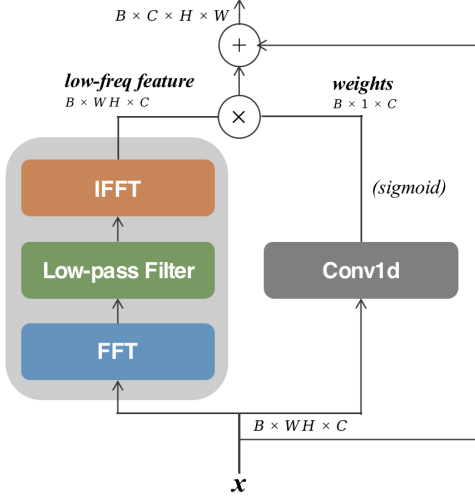
FFT

(sigmoid)

$B \times WH \times C$

x

Figure 1. The structure of FPCM

only the DC component is preserved with the model going deeper. Additionally, MSAs spatially smoothen feature maps with self-attention weights, acting like a natural denoising module for better generalization and robustness. With rigid theoretical analysis, Wang *et al.* [31] try to avoid over-smoothing by giving higher weights to high-frequency signals in deep ViTs. On the contrary, convolutional networks go the opposite way in terms of frequency characteristics and amplify high-frequency signals. As per the aforementioned findings and the underlying connection between Vit's low-frequency characteristics and its robustness, it is promising to supplement the low-frequency features that convolutional networks are missing.

3. Methodology

3.1. Frequency Preference Control Module (FPCM)

We first introduce the proposed FPCM (shown in Fig.1), which reconfigures low- and high-frequency components in features. Given an input x, we transform it into a frequency domain via DFT. We then suppress the high-frequency signal with an exponential low-pass filter, followed by the iDFT transformation to restore spatial domain features.

Formally, denote $x \in \mathbb{R}^{C \times H \times W}$ as the input feature, and $\mathcal{F} \in \mathbb{C}^{HW \times C}$ as its frequency representation. We have:

$$\mathcal{F}(u,v) = \sum_{a=0}^{H-1} \sum_{b=0}^{W-1} x_{a,b} e^{-j2\pi(\frac{v}{H}a + \frac{u}{W}b)}$$

where H and W represent the height and width of the input feature respectively, and j is the imaginary unit. In practice, the transformation from the discrete spatial domain to frequency domain can be done by FFT with an optimized computational complexity of $O(HWlog(HW))$. Subsequently,

the filtered frequency representation $\hat{\mathcal{F}}$ is formulated by:

$$\hat{\mathcal{F}}(u,v) = F(u,v)H(u,v)$$

where $H(u,v)$ denotes a low-pass filter in frequency domain. In our work, we mainly study the Gaussian low-pass filter (GLPF), which is a common choice in the field of image processing, causing no ringing effect:

$$H(u,v) = e^{-(\frac{D(u,v)}{2D_0})^2} = e^{-(\frac{D(u,v)}{2(HW*\beta)})^2}$$

where $D(u,v)$ represents the distance from the pixel (u,v) to the centre of the two-dimensional spectrum, and D_0 is the cutoff frequency. For any feature map with height H and width W, the total sequence length of the corresponding Fourier spectrum would be HW. Correspondingly, We set β as a hyper-parameter controlling the cut-off frequency. Having obtained the filtered frequency, we then apply the Fourier inverse transform to restore the feature map with only the low-frequency components $\hat{x}$, and derive the final re-weighted low-frequency features from the element-wise production of $\hat{x}$ and the inter-channel weights $\alpha \in \mathcal{R}^C$ learned from the original input:

$$\alpha = 0.5 * sigmoid(W_\alpha^T x) + 0.5$$

by default, we push the value interval of α to $[0.5, 1]$ to make the model focus more on the low-frequency features, inspired mainly by previous studies on robust learning frequency characteristics and success attempts [25]. Overall, the FPCM can be formulated as the weighted summation of low-frequency and high-frequency features:

$$FPCM(x) = \alpha\hat{x} + (1-\alpha)(x - \hat{x})$$

we also studied FPCM with fixed weights α, where our method will become completely parameter-free. The design of FPCM follows the following principles:

- Efficiency: the extra parameter and computational overhead it introduces are marginal compared to standard DNNs

- Differentiability: FFT and the subsequent low-pass filtering are essentially linear transformations that do not mask the gradient, which allows joint optimization with DNNs.

- Adaptability: learnable weights capture the inter-channel frequency characteristics and bring the possibility of adapting the frequency differences of the output feature maps between different stages (layers) of a model, various models, and datasets.
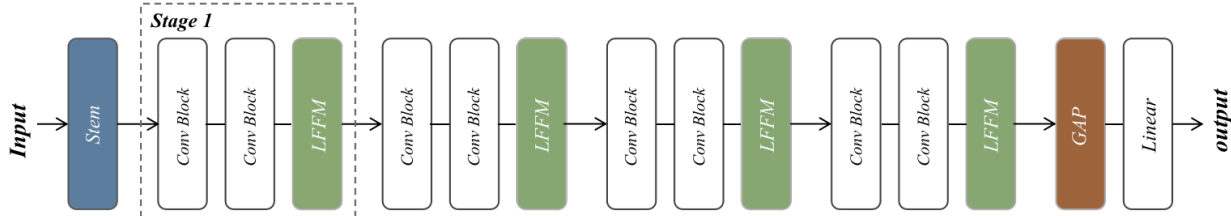
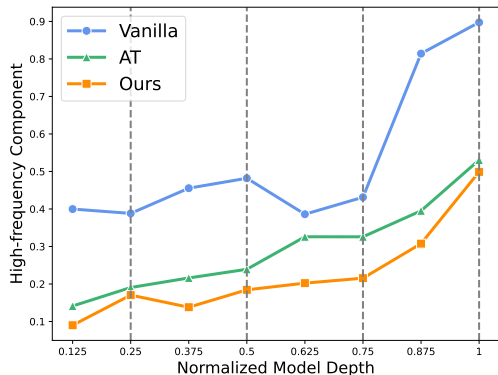Figure 2. Detailed architecture for our model altered from ResNet18.

Figure 3. Frobenius norm of the every layers' output high-frequency signals. Convolutional layers increase high-frequency components in feature maps. The vertical grey dashed line represents the last layer at every stage.

3.2. Harmonizing FPCM with Regular CNNs

In light of prior research demonstrating that convolutional layers typically increase feature map variance, with the variance tending to increase with depth and peaking at the ends of each stage, our investigation focuses on analyzing the feature map frequency characteristics of both a vanilla ResNet18 model and an adversarially trained one. We present the results in Fig.3, which illustrate that high-frequency feature components tend to aggregate as the model goes deeper. The learned high-frequency features of each stage peak at the stage's final layer, as indicated by a greater quantity of high-frequency components at the layer marked by the dashed line when compared to the previous layer. Our findings also demonstrate that, in comparison to the vanilla model, the adversarially trained model prioritizes low-frequency features, with an overall reduction in the number of high-frequency components, consistent with the findings outlined in [16] regarding the smoothing of convolutional kernels by adversarial training. Based on these observations, we hypothesize that incorporating FPCMs closer to the end of each stage can effectively reduce high-frequency signals and further enhance model robustness. To this end, we present an overview of our meta-model in Fig.2, utilizing the ResNet18 architecture as an example. We insert FPCM "painlessly" without destroying

the original model architecture, which also works for other kinds of models like WideResNets.

3.3. Frequency Principle of Deep Learning

Recently, Xu *et al.* [36] studied the F-principle of deep learning both empirically and theoretically, revealing that DNNs first model the low-frequency features in data. The F-Principle is a salient and intuitive notion, as it aligns with the principles governing human visual perception. Specifically, when humans encounter unfamiliar stimuli, they tend to process and retain coarse, high-level information about its shape and structure prior to encoding more detailed features. Empirically, the weight loss landscape is a widely used indicator to characterize the standard generalization gap in a standard training scenario. For robustness learning, it is also indicated in previous work that a flatter loss landscape leads to higher test robustness as long as the training is sufficient. Also in line with the frequency principle, early stopping can improve the generalization ability of DNNs in practice.

Taking these mentioned aspects into consideration, we endeavor to enhance the concordance between the learning process of the original model and FPCM by modifying the cutoff frequency. Intuitively, we adjust the cutoff frequency linearly as follows:

$$D_{0_t} = \frac{S}{2} + \frac{t * \left(\frac{S}{8} - \frac{S}{2}\right)}{T}$$

where t is the current model training epoch, T represents the total number of training epochs, and S is the length of the frequency spectrum. Therefore, in the process, we gradually decrease the cutoff frequency from $S/2$ to $S/8$. The same setting is used for the perturbations generated during training, but the cutoff frequency is fixed to $S/8$ during evaluation. The impetus behind our approach is to expedite the process of convergence while simultaneously enabling the model to attend to information with complete frequency during the initial training phase. However, we aim to employ more low-frequency signals in the terminal stage, which we believe is to be beneficial for robust learning.

Method	Clean	PGD-10	PGD-20	PGD-50	C&W	AutoAttack
PGD-AT	85.32	55.17	54.28	53.98	53.38	51.43
TRADES	84.72	56.75	56.10	55.9	53.87	52.30
MART	84.17	58.98	58.56	58.06	54.58	51.10
FAT	87.97	50.31	49.86	48.79	48.65	47.48
GAIRAT	86.30	59.64	58.91	58.74	45.57	40.30
AWP	85.57	58.92	58.13	57.92	56.03	53.90
LBGAT	**88.22**	56.25	54.66	54.30	54.29	52.23
LAS-AT	86.23	57.64	56.49	56.12	55.73	53.58
Ours-AT	85.97	55.35	54.45	54.24	53.93	51.87
Ours-TRADES	84.63	57.91	57.32	57.26	53.93	52.88
Ours-AWP	85.15	59.76	59.01	58.87	56.73	55.18
Ours-AWP-LAS	88.10	**60.97**	**59.67**	**59.31**	**57.38**	**55.47**

Table 1. Test robustness (%) on the CIFAR-10 database using WRN34-10. Number in bold indicates the best.

Method	Clean	PGD-10	PGD-20	PGD-50	C&W	AutoAttack
PGD-AT	85.13	51.13	50.17	49.82	48.54	47.42
TRADES	84.43	53.71	52.53	52.28	50.81	49.55
FAT	**87.72**	47.46	46.69	46.33	49.66	43.14
GAIRAT	83.40	55.81	54.76	54.21	38.81	31.25
AWP	82.66	55.64	54.95	54.93	50.92	50.21
LAS-AT	82.40	54.31	53.45	53.10	50.76	49.91
Ours-AT	82.90	53.35	52.50	52.29	51.01	49.26
Ours-TRADES	82.12	55.41	54.75	54.71	50.93	49.93
Ours-AWP	82.84	**56.49**	**55.94**	**55.86**	**51.81**	**51.28**

Table 2. Test robustness (%) on the CIFAR-10 database using ResNet18. Number in bold indicates the best.

4. Experiment

In this section, we conduct a set of experiments to establish the efficacy of the proposed approach. Subsequently, we investigate the effects of parameters such as cutoff frequency and weights for low-frequency features on both clean and robust accuracy. Further, we undertake an in-depth analysis of the frequency domain characteristics of adversarial training, and distinguish our method from other frequency-based techniques.

4.1. Evaluation Setup

Training We conduct experiments on CIFAR10 & CIFAR100 [20] mainly with ResNet18 [15] and WRN-34-10 [38], both of which are the mainstream models for robustness evaluation. We train ResNet18 for 120 epochs with the learning rate initialized at 0.01 and reduced to 1e-3 and 1e-4 at the 84th (0.7×120) and 108th (0.9×120) epoch. For WRN-34-10, the learning rate is initially started at 0.1 and decays to one-tenth and at the 42nd and 54th epoch respectively throughout the 60-epoch training process. The optimizer is SGD with momentum set to 0.9 and the weight decay factor of 3.5e-3 and 5e-4 for ResNet18 and WRN-34-

10 respectively. We train and evaluate the baseline counterpart (PGD-AT [22] and TRADES [39]) of our methods with the same training and evaluation settings to make a fair comparison. For the method we built based on AWP [34], we trained according to the settings proposed in their work, extending the number of training epochs to 200 for ResNet and 150 for WRN for performing sufficient training, given its stronger regularization properties. We also compare our method (Ours-AT, Ours-TRADES, and Ours-AWP) with the following baselines: 1) MART [32], 2) FAT [40], 3) GAIRAT [41], 4) LBGAT [7] and 5) LAS-AT [19].

Evaluation We employ several prevailing attack methods to evaluate the robustness of trained models, including PGD, $C\&W_\infty$, and AutoAttack. Unless otherwise specified, all attack methods are performed with perturbation budget $\epsilon = 8/255$ under ℓ_∞. PGD is equipped with random-start with attack step size set to $\alpha = 2/255$. The total step of C&W is 50. We report the averaged result of 5 independent runs where clean and robust accuracy are used as the evaluation metrics. We also leverage the Weighted Robust Accuracy (W-Robust) [14] to help measure the trade-off between

clean and robust accuracy, it is defined as follows:

$$\mathcal{A}_f = \pi_{D_{nat}} P_{D_{nat}}[f(x) = y] + \pi_{D_{adv}} P_{D_{adv}}[f(x) = y]$$

where $\mathcal{A}_f$ are the accuracy of a model f on x drawn from either the clean distribution D_{nat} or adversarial distribution D_{adv}. We adopt $\pi_{D_{nat}} = \pi_{D_{adv}} = 0.5$ by default, treating clean and robust accuracy equally for comprehensive evaluation.

4.2. Robustness Comparison

Our method is a plug-and-play component with no interventions in adversarial training. Therefore, it can be easily combined with other AT methods like TRADES which provides optimized loss design, and LAS which introduces an automatic strategy for generating adversaries.

Comparisons on CIFAR-10 dataset. The results on CIFAR-10 of WRN-34-10 and ResNet18 are listed in Table.1 and Table. 2. The three proposed models all exceed their counterpart baseline models under most of the attack scenarios. Considering the trade-off between clean accuracy and robustness, our method makes notable improvement on robust accuracy with no significant sacrifice of generalization ability on clean samples. For example, with WRN-34-10 as the experimented model, we boost the performance under PGD-50 attack by 0.95% and AutoAttack by 1.28% upon the powerful AWP method. We can further push up the robustness under PGD-50 by 0.66% and under AA by 0.29% with LAS to a generate better strategy for perturbation, which also shows the compatibility of our method. In terms of our ResNet18 model based on PGD-AT, we further improve the robustness under PGD-50 by 2.47%, C&W by 2.47%, and AA by 1.84%. For both ResNet18 and WRN-34-10, our method based on AWP achieves the best performance under all attacks.

In order to provide additional evidence of the efficacy of our proposed methodology, we endeavored to create a model using WRN-70-16 architecture and compared its performance against state-of-the-art robust models without additional training data. The results are presented in Table. 5. Our approach, build upon AWP, was found to yield greater robust accuracy under AA. Moreover, the performance enhancement yielded by our method is on par or outperforms that observed with the potent LAS technique. Notably, we also can build a more robust model based on LAS-AWP.

Comparisons on more complicated datasets. To demonstrate the generalization ability of our methods, we conduct experiments on CIFAR-100 which contains more categories than CIFAR-10 and Imagenette which is of higher resolution (160*160). The Imagenette dataset is a subset of ImageNet, consisting of ten easily classifiable categories. The results are given in Table 3 and Table 4. Our models all exceed their baselines in terms of robustness. Specifically,

Method	Clean	PGD-20	C&W	AA
PGD-AT	**86.6**	62.5	60.4	59.3
Ours-AT	86.5	**63.0**	**61.5**	**60.5**
TRADES	**83.7**	64.8	62.1	61.9
Ours-TRADES	83.5	**65.3**	**62.8**	**62.4**

Table 3. Test robustness (%) on the Imagenette database using ResNet18. Number in bold indicates the best.

Method	Clean	PGD-20	C&W	AA
PGD-AT	57.75	27.18	23.41	22.37
TRADES	56.75	28.05	24.17	22.89
Ours-AT	**58.79**	28.75	**26.60**	**25.19**
Ours-TRADES	56.89	**30.44**	26.33	25.13
PGD-AT	60.21	30.78	29.50	27.39
TRADES	58.43	29.23	27.05	25.78
Ours-AT	**60.22**	**31.95**	**30.14**	**27.81**
Ours-TRADES	57.75	31.88	27.93	26.98

Table 4. Test robustness (%) on the CIFAR-100 database using ResNet18 (top column) and WRN-34-10 (bottom column). Number in bold indicates the best.

Method	Clean	AutoAttack
Gowal *et al.* [13]	85.29	57.20
LAS-AWP [19]	85.66	57.61
Ours-AWP	**85.94**	**57.62**

Table 5. Clean and robust accuracy (%) on the CIFAR-10 with WRN-70-16

on the imagenette database, we improve performance under AA by 1.2% compared to PGD-AT and 0.5% in comparison to TRADES. On CIFAR-100 with ResNet18 as the targeted model, the robustness under PGD-20 and AA increases by 2.39% and 2.24% respectively. All the above mentioned numbers reveal that our method can be easily combined with AT frameworks and further improve their performance steadily across different models and datasets.

4.3. Ablation Studies

We utilize two hyper-parameters that merit further investigation, where α governs the trade-off between low- and high-frequency components and β regulates the cutoff frequency of the low-pass filter.

4.3.1 Cutoff frequency

Results of various settings for cutoff frequency are listed in Table. 7. With the weights $\alpha \in \mathcal{R}^C$ staying learnable, we first try to set fixed β for the whole training and evaluation process. A smaller beta equates to less high-frequency information being retained. It is consistent with the intuition that the clean accuracy drops with β set to be smaller, while the robust accuracy goes the opposite way. This suggests

Metrics	Fixed weights							Learnable	
	0	0.1	0.25	0.5	0.75	0.9	1	Conv1d	MLP
Clean	-	80.01	81.25	81.86	82.34	81.97	82.27	81.87	**82.83**
PGD-20	-	53.19	53.81	53.91	53.76	54.23	54.16	**54.75**	54.10
W-Robust	-	66.60	67.53	67.94	68.05	68.10	68.22	68.31	**68.47**
Param	-							1,360	0.35M

Table 6. Clean and robust accuracy on CIFAR10 dataset of our ResNet18 model with various weights for low-frequency features.

	Linear	Learnable	Fixed		
			0.5	0.25	0.125
Clean	82.12	81.89	**82.56**	81.15	80.32
PGD-20	54.75	54.85	52.34	55.12	**55.34**
AA	49.96	49.03	49.08	49.66	**50.13**
W-Robust	**66.04**	65.46	65.82	65.41	65.23

Table 7. Clean and robust accuracy of our method with the cutoff frequency factor β varying.

Stages	Max	Min	Mean	Var
stage1	0.9648	0.5393	0.6879	0.0064
stage2	0.9777	0.5354	0.7024	0.0103
stage3	0.8047	0.7403	0.7697	0.0001

Table 8. Some statistics about the learned weights α at different stages of a WRN-34-10 model.

that convolutional networks rely on high-frequency texture information for better classification performance on clean samples. In contrast, robust models tend to rely more on low-frequency signals to achieve greater robustness, which could partially explain why the clean accuracy of robust models is generally lower than that of vanilla models. We also tried with learnable parameters $\hat{\beta} \in \mathcal{R}^C$, initialized as an all-ones vector, to realize adaptive cutoff frequency for different channels. The possible explanation for the absence of progress in contrast to fixed values might be associated with its redundancy with learnable frequency weights α. Upon examining the inner numerical values acquired, an intriguing phenomenon is revealed whereby the estimated coefficient $\hat{\beta}$ exhibits a tendency to diminish in deeper stages, which may indicate that less high frequency is needed in deeper layers for robust learning. Lastly, our proposed linear adjustment of the cutoff frequency during training yields the highest W-Robust.

4.3.2 Re-weighing of low frequency components

The weights balancing between low- and high-frequency feature components play an important role in robust learning. Therefore, we conduct experiments with α ranging from 0 to 1, as shown in Table.6. Due to the fact that a majority of information is in DC components, the model cannot properly converge with α set to 0. Altering α to a value

greater than 0.5, i.e., prioritizing low-frequency parts, leads to enhanced accuracy concerning both robustness and clarity, thereby elevating W-Robust by approximately 0.37%-1.79%. Incorporation with learnable weights elevates the performance further by 0.39% on W-Robsut. While we use only one convolutional layer with minimal parameters to develop relationships among the different channels in our initial setup, we further study the FPCM regarding more parameters and non-linear operations (indicated as *MLP*), but the improvement is marginal (i.e., +0.03% W-Robust).

We proceeded to examine the learned weights and computed relevant statistics as illustrated in Table.8. As per our previous observation, we noticed that deeper stages generally had higher weights assigned to low-frequency components compared to earlier stages. For instance, the average α in stage3 was observed to be 0.7697 while it was merely 0.6879 in stage1. It should be highlighted that the weight value in stage3 did not drop below 0.74 emphasizing the significance of low-frequency signals aiding robust learning in deeper layers. From an alternative perspective, we observe that robust models aim to counteract the issue highlighted in Fig. 3, where high-frequency signals gradually accumulate in CNNs. By analyzing the statistics of various channels and samples, we have observed that the sample-wise variance of the learned weights is typically small (i.e., usually lower than 1e-4). This implies that the FPCM approach has been tailored to alter the intrinsic frequency characteristics of the robust model, instead of being data-specific.

4.4. Discussions

4.4.1 Frequency-bias of robust learning

Previous work has proposed that convolutional layers have high-pass characteristics that cause vanilla ResNet models to be more sensitive to high-frequency signals and thus more vulnerable to high-frequency noise. Our experiment in Fig.3 does verify this claim, finding that the high-frequency signal becomes richer as the network gets deeper. In this section, we delve deeper into the frequency characteristics of adversarially trained models.

We first study the robustness under noise containing different proportions of high-frequency components, as shown in Fig. 4. We still adopt the low-pass filter formulation introduced in Sec.3.1 to conduct this experiment. Therefore,
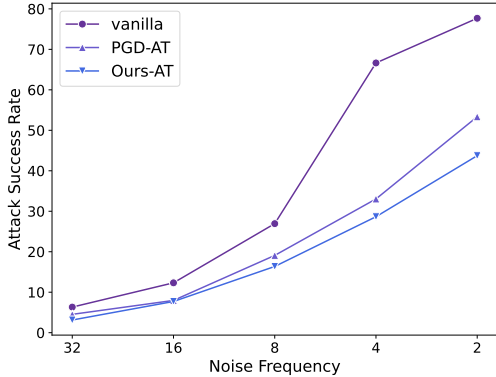
Figure 4. The attack success rate (%) of noise with respect to its frequency. The numbers on the x-axis indicates the cut-off frequency factor $1/\beta$.

a smaller β indicates less high-frequency signal contained in the perturbation. We measure the accuracy drop with frequency-based random noise and the perturbed data can be formulated by: $x_{noise} = x_0 + \mathcal{F}^{-1}(\mathcal{F}(\delta) \odot H)$, where x_0 is the clean data, $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ are Fourier transform and inverse Fourier transform, δ is random noise, and H denotes the filter. For the vanilla model, the slope of the line graph increases gradually with more high-frequency components, indicating its vulnerability to high-frequency noise. While similar patterns can be seen in PGD-AT, PGD-AT significantly reduces the model vulnerability against high-frequency noises, which indicates AT models naturally focus more on the low-frequency features in data. Furthermore, by explicitly letting the model focus on the low-frequency components, we further improve the robustness of the adversarially trained model to high-frequency noise, resulting in an overall more robust model. In contrast, the robustness to very low-frequency signals does not improve considerably from vanilla to PGD-AT and Ours-AT.

We then take a look at the learning curve of our method with fixed weights α for low-frequency features. Results are plotted in Fig.5. With α set to 0.1, the robust accuracy is lower than other settings during the whole training process. The plot of the loss landscape may partially explain this phenomenon as it is not sufficiently trained on the training data with impaired low-frequency signals. As the α continues to rise, the loss curve moves further down, indicating low-frequency signals can accelerate the robust learning process. We believe that all of the above phenomena illustrate the dominance of low-frequency features in robust learning.

4.4.2 Differences & advantages over other frequency-based methods

Zhang *et al*. [43] proposed to suppress high-frequency components in the input image to perform adversarial de-
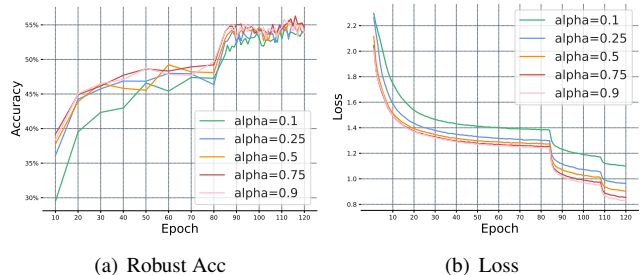

(a) Robust Acc	(b) Loss

Figure 5. Robust accuracy and loss landscape during the training process. For the first 80 epochs, we only keep the accuracy for every 10 epoch for simplicity. Here we use $\beta = 0.125$.

Method	Clean	PGD-20	C&W	AA	W-Robust
FR	80.59	59.49	54.33	52.06	66.33
FR-SWA	81.09	**60.12**	56.14	54.35	67.72
Ours	**85.15**	59.01	**56.73**	**55.18**	**70.17**

Table 9. Clean and robust accuracy of WRN-34-10 on CIFAR-10 dataset. W-Robust is the average of *Clean* and *AA*.

fense. Following the idea of applying low-pass filtering (LPF) to the natural and adversarial inputs, Huang *et al*. [16] additionally introduce a frequency regularization (FR) term, which aims at bridging the frequency domain distance (measured by L_1) of clean and adversarial data, by applying DFT on the output logits. They further tried to smooth the convolutional kernels by leveraging Stochastic Weight Averaging (SWA) [18]. Bernard *et al*. [2] also employs regularization on the frequency of output logits with original and low-pass filtered images as the input. Orientating real-world super resolution models, Yue *et al*. [37] adopts frequency masks to block out high-frequency components in input that possibly contain harmful perturbations in a stochastic manner. Since FR [16] is the state-of-the-art method derived from the frequency perspective, we make further comparison with it in clean and robust accuracy, and the results are listed in Table. 9. It can be seen that our method not only has obvious advantages under AA (i.e., 0.83% accuracy improvement), but retains greater generalization on clean data (+4.06%). The significant decrease in the accuracy of FR on clean samples may be due to its direct filtering of the input images, which leads to irretrievable signal components that are instrumental for the model to correctly classify clean samples.

In a nutshell, we are the first to focus on the frequency of intermediate features rather than that of the input clean data or adversary. Our method enhances the robustness of models by explicitly modifying the frequency bias of robust models, as opposed to relying on stronger regularization techniques.

5. Conclusion

We proposed the Frequency Preference Control Module (FPCM) to adaptively reconfigure low- and high-frequency components in the intermediate feature representations. Based on our finding that high-frequency signals tend to aggregate as the model deepens, we designed a heuristic scheme to insert FPCM after each stage of the regular model to promote harmonious operation among them. Empirical studies demonstrate that our method can be easily incorporated into any adversarial training framework and further boost its performance with minimal computation overhead. Furthermore, by leveraging FPCM, we conducted experiments that shed light on the frequency characteristics of robust models, revealing insights regarding the impact of low-frequency bias on their robustness.

6. Acknowledgement

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

[2] Rémi Bernhard, Pierre-Alain Moëllic, Martial Mermillod, Yannick Bourrier, Romain Cohendet, Miguel Solinas, and Marina Reyboz. Impact of spatial frequency based constraints on adversarial robustness. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[3] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[5] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020.

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[7] Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15721–15730, 2021.

[8] Yao Deng, Xi Zheng, Tianyi Zhang, Chen Chen, Guannan Lou, and Miryung Kim. An analysis of adversarial attacks and defenses on autonomous driving models. In *2020 IEEE international conference on pervasive computing and communications (PerCom)*, pages 1–10. IEEE, 2020.

[9] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

[10] Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

[11] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[13] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[14] Nezihe Merve Gürel, Xiangyu Qi, Luka Rimanic, Ce Zhang, and Bo Li. Knowledge enhanced machine learning pipeline against diverse adversarial attacks. In *International Conference on Machine Learning*, pages 3976–3987. PMLR, 2021.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Binxiao Huang, Chaofan Tao, Rui Lin, and Ngai Wong. Frequency regularization for improving adversarial robustness. *arXiv preprint arXiv:2212.12732*, 2022.

[17] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR, 2018.

[18] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[19] Xiaojun Jia, Yong Zhang, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao. Las-at: adversarial training with learnable attack strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13398–13408, 2022.

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

[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[23] Shishira R Maiya, Max Ehrlich, Vatsal Agarwal, Ser-Nam Lim, Tom Goldstein, and Abhinav Shrivastava. A frequency perspective of adversarial robustness. *arXiv preprint arXiv:2111.00861*, 2021.

[24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[25] Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022.

[26] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. *Advances in Neural Information Processing Systems*, 32, 2019.

[27] Mathieu Salzmann et al. Learning transferable adversarial perturbations. *Advances in Neural Information Processing Systems*, 34:13950–13962, 2021.

[28] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

[29] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*, 2021.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[31] Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. *arXiv preprint arXiv:2203.05962*, 2022.

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

[33] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[34] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

[35] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

[36] Zhi-Qin John Xu, Yaoyu Zhang, and Tao Luo. Overview frequency principle/spectral bias in deep learning. *arXiv preprint arXiv:2201.07395*, 2022.

[37] Jiutao Yue, Haofeng Li, Pengxu Wei, Guanbin Li, and Liang Lin. Robust real-world image super-resolution against adversarial attacks. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5148–5157, 2021.

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

[39] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

[40] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pages 11278–11287. PMLR, 2020.

[41] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2010.01736*, 2020.

[42] Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *International conference on machine learning*, pages 7502–7511. PMLR, 2019.

[43] Zhendong Zhang, Cheolkon Jung, and Xiaolong Liang. Adversarial defense by suppressing high-frequency components. *arXiv preprint arXiv:1908.06566*, 2019.