Rethinking and Improving Robustness of Convolutional Neural Networks: a Shapley Value-based Approach in Frequency Domain

Anonymous Author(s)
Affiliation
Address
email

Abstract

The existence of adversarial examples poses concerns for the robustness of convolutional neural networks (CNN), for which a popular hypothesis is about the frequency bias phenomenon: CNNs rely more on high-frequency components (HFC) for classification than humans. Most works manually select and roughly divide the image frequency spectrum for their analysis. In this work, we introduce Shapley value, a metric of cooperative game theory, into the frequency domain and propose to quantify the positive (negative) impact of every frequency component of data on CNNs. Based on the Shapley value, we show that the importance proportion between low frequency components (LFC) and HFC varies at an instance-level. We further investigate the distinctive behavior of standard trained (ST) and adversarial trained (AT) models on clean and adversarial examples at the dataset-level. The observations motivate us to perform an in-depth analysis and lead to multiple novel hypotheses about i). the difference between ST model and AT model; ii), the attack bias at different frequency component level; iii). the generalization limitation of AT including the fairness problem. Finally, we propose a shapley-value guided data augmentation for better robustness. Experimental results have shown its effectiveness.

1 Introduction

Though convolutional neural networks (CNNs) show their great generalization power in various vision tasks, the existence of adversarial examples [28, 14, 20] show that CNNs are prone to be affected by adversarial noises that are imperceptible to humans. One popular hypothesis about the brittleness of CNNs is that CNNs can exploit high frequency components (HFC) that are not perceivable to humans [32, 33, 1, 17]. Moreover, a line of works propose to utilize HFC filtering to improve attack or defense methods [11, 18, 41]. However, towards the above *frequency bias* phenomenon, most works [32, 33, 17] intuitively divide the image frequency spectrum into two parts as low frequency component (LFC) and HFC then evaluate the contribution of each part. Moreover, current works [15, 24] reveal that some successful adversarial attacks also depend on LFC, which reveals the limitation of previous works: i). overlooking the contribution of each individual frequency component in LFC and HFC; ii). only do statistical dataset-level analysis.

Instead of studying the impact of HFC and LFC at the dataset-level, we resort to quantify the contribution of each frequency component with help of Shapley value [23]. Shapley value is a metric from cooperative game theory which has been widely used in explainable machine learning [7, 2, 19, 27]. Since these works measure pixel-level importance for prediction in the spatial domain, our work is the first to apply Shapley value to the frequency domain. Shapley value satisfies several desirable properties for quantifying the contribution of each frequency component, and we justify that applying

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

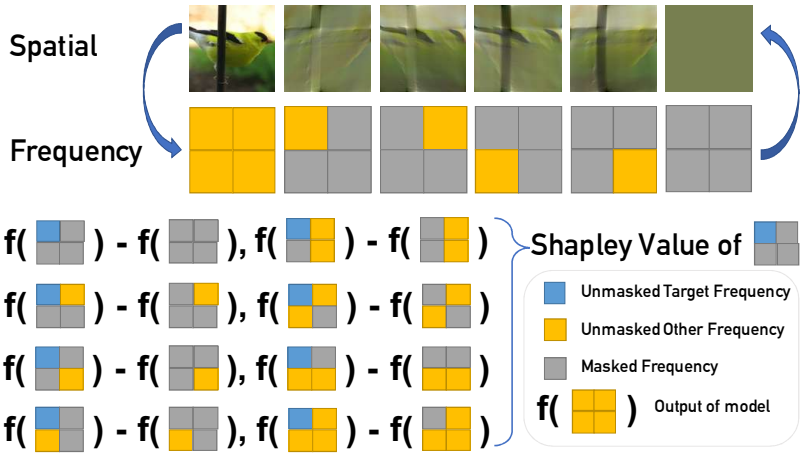


Figure 1: Illustration of the Shapley value calculated in frequency domain when the frequency components are split into 2×2 patches. The Shapley value of a frequency component is the average marginal effect the frequency component has on the output of the model.

Shapley value in the frequency domain rather than spatial domain is a more natural approach on CNNs (c.f. Remark 1).

Based on the Shapley value, we quantify the contribution of each frequency component by measuring the difference between the model's output regarding the absence of the frequency component. An illustration of the quantifying method is presented in Fig. 1. Through our quantification via Shapley value, we are capable of inspecting the contribution of different frequency components of each sample. For simplicity, we refer to the frequency components with positive contribution as positive frequency components (PFC) and frequency components with negative contribution as negative frequency components (NFC).

At the instance level, we have observed intriguing disparity between different data samples, *e.g.* model may rely more on HFC to inference on some instances, as shown in Fig. 2(a), or rely more on LFC to inference on the other instances, as shown in Fig. 2(b). At the dataset level, we have observed clear patterns that provide insights into the cause of the adversarial robustness, the attack bias on different frequency components, and the fairness problem in adversarial training (AT). We utilize our quantification result for an in-depth analysis to the defense mechanism of AT. Based on our analyses, we further propose a simple and principled data augmentation method that utilizes the NFCs we found to achieve better robustness. Our major contributions are as follows:

- Instead of inspecting CNNs based on the manually divided frequency spectrum, we introduce the Shapley value metric into the frequency domain and directly quantify the contribution of each individual frequency component on CNNs, with the advantage of desirable theoretical properties.
- At dataset level, we find that ST models merely exploit HFC but are more vulnerable on HFC under adversarial attacks while AT models improve robustness via filtering out HFC, bringing a new hypothesis about the trade-off between accuracy and robustness [25, 39, 30].
- We observe the *attack bias* towards frequency component-level contribution: positive parts in HFC is more easily attacked while negative ones in LFC are more brittle.
- We further take a frequency-domain perspective to investigate the fairness problem in AT with explanations for the class-wise disparity [34, 29].
- Motivated by the discriminative class-wise patterns of NFC, we propose to augment data with NFC of its class, named by Class-wise Shapley value-guided Augmentation (CSA), to boost robustness.

2 Notations and Preliminaries

In this section, we set up the notations used in this paper, followed by basic ideas of Discrete Fourier Transform, adversarial attack and defense, and the Shapley value.

Notation: $\langle X, y \rangle$ denotes a data sample where $X \in \mathbb{R}^{d_1 \times d_2}$ is a image and $y \in \{1, \ldots, C\}$ is its label where C is the number of classes. For simplicity, we omit the channel of images.

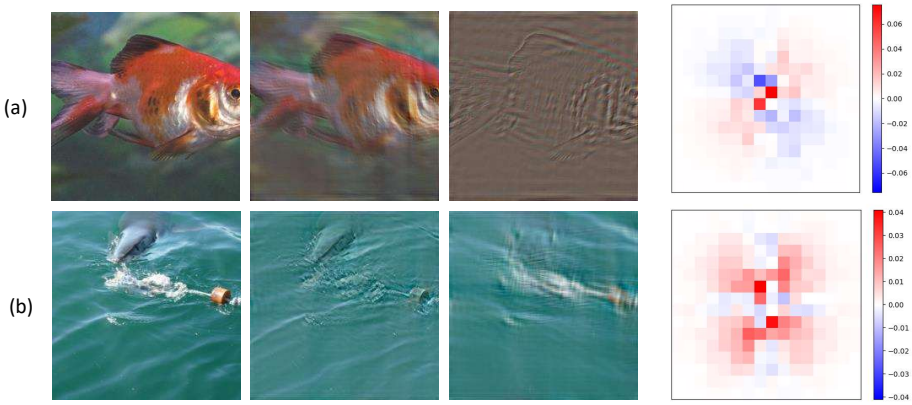


Figure 2: Result of Shapley value based quantification on ImageNet. Left to right: origin image, image reconstructed with positive frequency components (PFC), images reconstructed with negative frequency components (NFC), the heat map of Shapley value in frequency domain. Note that the heatmap is shifted,the closer to the center the lower the frequency

$f(\cdot,\theta):\mathbb{R}^{d_1\times d_2}\rightarrow\mathbb{R}^C$ denotes a neural network that takes image as input and output a prediction where θ represent its parameters. We use $\ell(\cdot,\cdot)$ to denote loss function and $\|\cdot\|_p$ to denote ℓ_p norm. $\mathcal{F}(\cdot):\mathbb{R}^{d_1\times d_2}\rightarrow\mathbb{C}^{d_1\times d_2}$ denotes the Discrete Fourier Transfom (DFT). $\mathcal{F}^{-1}(\cdot):\mathbb{C}^{d_1\times d_2}\rightarrow\mathbb{R}^{d_1\times d_2}$ denotes the inverse Discrete Fourier Transform.

Discrete Fourier Transform (DFT). DFT transforms finite signals into complex-valued functions of frequency. In this paper the 2D-Discrete Fourier Transform is computed as:

$$\mathcal{F}(X)(u,v)=\sum_{m=0}^{d_1-1}\sum_{n=0}^{d_2-1}X(m,n)e^{-i2\pi(\frac{mu}{d_1}+\frac{nv}{d_2})} \tag{1}$$

The inversion of DFT is computed as:

$$X(m,n)=\frac{1}{d_1d_2}\sum_{u=0}^{d_1-1}\sum_{v=0}^{d_2-1}\mathcal{F}(X)(u,v)e^{i2\pi(\frac{mu}{d_1}+\frac{nv}{d_2})} \tag{2}$$

Adversarial Attack. Adversarial Attack [28, 14] aims to find a small noise δ bounded in ℓ_p space to cause failure in neural networks.

$$\delta=\arg\max_{\delta}\ell(f(X+\delta),y),s.t.\|\delta\|_p\leq\epsilon \tag{3}$$

Defense. Adversarial training [20] becomes the most effective defense method [3], which is typically conducted by minimizing loss directly on adversarial training data.

$$\theta=\arg\min_{\theta}\mathbb{E}[\max_{\|\delta\|_p\leq\epsilon}\ell(f(X+\delta),y)] \tag{4}$$

Definition of the Shapley Value: Let $\mathcal{N}=\{1,\cdots,n\}$ denotes the finite set of players. Each non-empty subset $\mathcal{S}\subseteq\mathcal{N}$ is called a coalition. A cooperative game is defined by the pair $(\mathcal{N},V)$, where $V:2^{\mathcal{N}}\rightarrow\mathbb{R}$ is a mapping that assigns a real number to each coalition and satisfies $V(\emptyset)=0$. Let $\pi\in\Pi(\mathcal{N})$ denotes a permutation of player set $\mathcal{N}$ and $\pi(i)$ is the position of player $i\in\mathcal{N}$, where $\Pi(\mathcal{N})$ is the set of permutations. Then the predecessor set of player $i\in\mathcal{N}$ in permutation π is $P_i^{\pi}=\{j\in\mathcal{N}|\pi(j)<\pi(i)\}$. The Shapley value ϕ_i^V of player i is defined as:

$$\phi_i^{\mathcal{N},V}=\frac{1}{|\Pi(\mathcal{N})|}\sum_{\pi\in\Pi(\mathcal{N})}[V(P_i^{\pi}\cup\{i\})-V(P_i^{\pi})] \tag{5}$$

The Shapley value [23] of the player is the average marginal contribution of the player to the value of predecessor set in every possible permutation.

Properties of Shapley Value: It satisfies many desirable properties including *Null player*, *Efficiency*, *Symmetry*, *Linearity*. See Appendix D for more details about the properties of Shapley value.

3 Shapley Value-based Quantification

3.1 Contribution quantification for frequency component

In our assumption, frequency components in the data sample cooperate with each other and compose the feature that the model process to infer.To apply the Shapley value to the frequency components of input, we construct a cooperative game in this scenario. Considering frequency components as players, we define the set of players for data sample $\langle X, y \rangle$ as:

$$\mathcal{N}_X = \{\mathcal{F}(X)(0,1), \cdots, \mathcal{F}(X)(d_1-1, d_2-1)\} \tag{6}$$

Note that $\mathcal{F}(X)(0,0)$ determines the mean value of the image, we exclude this frequency component from the player set and keep the mean value of the image unchanged to fairly calculate the marginal contribution of the frequency components.

For a model $f(\cdot)$, we define the output of the model on $S \subseteq \mathcal{N}$ as:

$$f(S) = f\left(\mathcal{F}^{-1}(\sigma_S \odot \mathcal{F}(X))\right)_y \tag{7}$$

where $f(\cdot)_y$ denotes the y-th output of the model and $\odot$ denotes Hadamard product. $\sigma_S \in [0,1]^{d_1 \times d_2}$ is the mask generated according to S satisfying:

$$\sigma_S(u,v) = \mathbb{I}[\mathcal{F}(X)(u,v) \in S] \tag{8}$$

Specifically, $\sigma_S(0,0) = 1$. The corresponding characteristic function of the game is defined as:

$$V(S) = f(S) - f(\emptyset), S \in \mathcal{N} \tag{9}$$

By definition, we calculate the Shapley value $\phi_{u,v}^{\mathcal{N}_X, f}$ of frequency component $\mathcal{F}(X)(u,v)$ according to Eq. 5. The Shapley value of a frequency component evaluates the average marginal contribution of the frequency component to the neural network's output of the ground-truth class. The Shapley value for frequency components have the desirable properties of the Shapley value including *Null player*, *Efficiency*, *Symmetry* and *Linearity*.

3.2 Justification for frequency based Shapley value on CNN

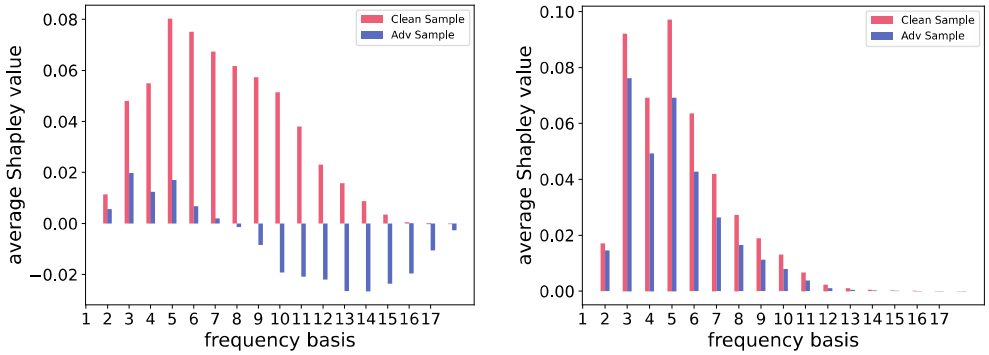
When it comes to measuring the contribution of input features in images, the common approach is decomposing features into pixels in the spatial domain and considering each pixel as a player [7, 2, 19, 27]. Missing players that are not in a coalition S are replaced with a reference value. The reference value are often chosen intuitively as zero or the mean value of the image. The choice of reference value may affect the explanatory results on CNN.

Remark. 1 (Decomposition of convolution) *The output of a convolution operation $f_{conv}(\cdot)$ on X can be decomposed into a linear function of output on frequency components on X. Formally,*

$$f_{conv}(X) = \sum_{u=0}^{d_1-1} \sum_{v=0}^{d_2-1} \left[\mathcal{F}(X)(u,v) f_{conv} \left[Re\left(\frac{1}{d_1 d_2} \sum_{u=0}^{d_1-1} \sum_{v=0}^{d_2-1} e^{i2\pi(\frac{mu}{d_1} + \frac{nv}{d_2})} \right) \right] \right] \tag{10}$$

As proposed in Remark 1, the output of a convolution operation is a linear function of outputs with respect to each frequency component. Hence the reference value for frequency components is naturally zero.

Furthermore, the Shapley value intuitively aligns more with what is important in an additive setting, as mentioned in previous works [16]. For non-additive models, Shapley value may fail to explain the importance of players. For example, function $f(x) = x_1 x_2 x_3$ with reference value set to be zero. The Shapley value of each variable is $\frac{1}{3} f(x)$ while the value of each variable may vary. For frequency components, the CNN is approximately a linear function. See Appendix D for detailed proof.



(a) ST model (b) AT model

Figure 3: Average shapley value of frequency componets in clean samples and adversarial samples over frequency basis on CIFAR10

4 Rethinking Robustness in Frequency Domain

In this section, we conduct extensive experiments on CIFAR10 and ImageNet to give a fundamental analysis of the contributions of different frequency components. In Sec. 4.1, we introduce details and results of instance-level sampling for Shapley value calculation. In Sec. 4.2 and Sec. 4.3, we inspect the contribution of frequency components of clean (adversarial) data with standard (adversarial) training paradigms respectively, followed by an in-depth analysis of the way adversarial attacks work in Sec. 4.4. We demonstrate the discussion of fairness in Sec. 4.5

4.1 The Shapley Value of Frequency Components

To get the Shapley value of each frequency component, we employ Monte Carlo Permutation Sampling to approximate the Shapley value [23]. We sample the Shapley value on CIFAR10 and ImageNet. To reduce computation complexity, we split frequency components of data samples in CIFAR10 into 16×16 patches that contains 2×2 frequency components and split frequency components into 32×32 patches that contains 7×7 frequency components in ImageNet. As for the model, we use ResNet-18 for CIFAR10 and use pre-trained ResNet-50 that is provided in robustbench [9] for ImageNet. See Appendix A for more details.

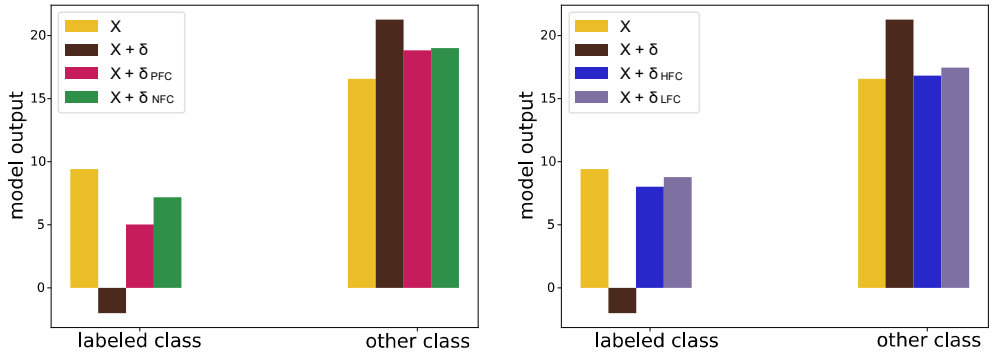
Quantification result We demonstrate our quantification result on ImageNet in Fig 2. Each row contains the results of a data sample. We show the original image, the heatmap of Shapley value, the reconstructed image from positive frequency components (PFC), and the reconstructed image from negative frequency components (NFC) from left to right.

The Shapley value of each frequency component varies between different data samples. As shown in Fig. 2, the NFCs in the first image are mainly of high frequency or medium frequency while the LFCs are mostly positive. As a result, the NFCs compose the contour of the goldfish and other textures while PFCs recover the main body of the fish just like how humans perceive it. However, the "shark" image in the second row shows the opposite direction to "goldfish": PFC mainly contains the contour of the shark, i.e., the HFC information, which means the *frequency bias* varies at the instance-level. See Appendix A for more details and more results on ImageNet and CIFAR10.

4.2 Standard trained Models Merely Exploit HFC but Are More Vulnerable on HFC

Observations in Sec. 4.1 are based on clean examples. We further inspect the difference between clean and adversarial examples in the frequency domain based on our Shapley value metric. Adversarial examples are generated with PGD-20 with a Linf bound of $8\backslash255$ and step size of $2\backslash255$. We employ ResNet-18 as the backbone.

As shown in Fig. 3(a), i). for clean examples, LFC have significant positive Shapley value on average for standard trained (ST) model while the average Shapley value of HFC are nearly zero, which means ST model merely exploits HFC; ii). for adversarial examples, the average Shapley value of HFC are significantly negative, which demonstrates that ST model is more vulnerable on HFC.



(a) Experiment result of δ_{PFC} and δ_{NFC} (b) Experiment result of δ_{HFC} and δ_{LFC}

Figure 4: The average output of labeled class and other classes of a ST model on CIFAR10. δ is the adversarial noise generated against the model. δ_{PFC}, δ_{NFC}, δ_{HFC} and δ_{LFC} are the masked adversarial noise that are constrained on PFC, NFC, HFC and LFC of the data respectively. Each figure demonstrates the model output of clean sample, adversarial sample, and samples attacked by restrained adversarial noise.

4.3 Adversarial trained Models Improve Robustness as a HFC Filter

Having observed the discriminative behavior of clean (adversarial) examples on ST, we step into analyzing the behavior of Adversarial trained (AT) model. AT model is trained with PGD-10 with a ℓ_∞ bound of $8\backslash255$ and step size of $2\backslash255$. The details of the training are provided in Appendix A.

As shown in Fig. 3(b), LFC have high positive Shapley value on both clean and adversarial examples and the average Shapley value of HFC is nearly zero, which may be the key to improved robustness of AT. We hypothesize that AT tends to filter out NFC and since NFC mainly reside in HFC, the final outcome is that AT focuses more on LHC, overlooking information in HFC. However, as Fig. 3(a) shows, the HFC of clean images plays a remarkable positive impact on model prediction. Thus we assume that the tension between accuracy and robustness of AT possibly accounts for the overlooking of AT on HFC, especially positive ingredients.

4.4 Attack bias in Frequency Domain

Followed by findings in Sec. 4.2, we further inspect the attack bias on ST at a more fine-grained level. For an adversarial noise δ generated by Eq. 3 on ST model f and data sample X, we transform it into frequency domain as $\mathcal{F}(\delta)$. To determine the impact of adversarial noise, we apply masks based on different metrics on the adversarial noise δ in the frequency domain and recover it back to the pixel domain to evaluate its effect combined with the original example.

4.4.1 Adversarial Noises Play Different Roles on PFC and NFC

We first build masks based on the division between PFC and NFC of the data sample and the masked noise is defined as follows:

$$\begin{aligned} \delta_{PFC} &= \mathcal{F}^{-1}(\sigma_{PFC} \odot \mathcal{F}(\delta)), \qquad \sigma_{PFC}(u,v) = \mathbb{I}(\phi^{X,f}_{u,v} > 0) \\ \delta_{NFC} &= \mathcal{F}^{-1}(\sigma_{NFC} \odot \mathcal{F}(\delta)), \qquad \sigma_{NFC}(u,v) = \mathbb{I}(\phi^{X,f}_{u,v} < 0) \end{aligned} \tag{11}$$

We future split class space into target class and other classes and we calculate the output w.r.t. labeled class as $f(X)_y$ and the output regarding other classes is defined as $\log \sum_{c \neq y} \exp\left(f(X)_c\right)$. As shown in Fig. 4(a), the unmasked noises δ achieve the most effective affect. Moreover, adversarial noises δ_{PFC} lead to a lower output regarding the labeled class than δ_{NFC} while δ_{NFC} lead to a slightly higher output regarding other classes than δ_{PFC}.

The above result indicates that adversarial noises masked by image PFC mainly disturb the features of the labeled class exploited by the model while noises masked by image NFC mainly inject features of other classes to confuse the model.

4.4.2 Roles of Adversarial Noises on HFC and LFC Are Similar to Noise on PFC and NFC

Similar to Eq. 11, we define adversarial noises δ masked by image HFC and LFC as

$$\begin{aligned} \delta_{LFC} &= \mathcal{F}^{-1}(\sigma_{LFC} \odot \mathcal{F}(\delta)), \qquad \sigma_{LFC}(u,v) = \mathbb{I}(\|u\| + \|v\| < \epsilon_{low}) \\ \delta_{HFC} &= \mathcal{F}^{-1}(\sigma_{HFC} \odot \mathcal{F}(\delta)), \qquad \sigma_{HFC}(u,v) = \mathbb{I}(\|u\| + \|v\| > \epsilon_{high}) \end{aligned} \tag{12}$$

where ϵ_{low} and ϵ_{high} is the intuitively determined threshold for low frequency and high frequency. In our experiment, $\epsilon_{low} = 0.3(d_1 + d_2)$, $\epsilon_{high} = 0.7(d_1 + d_2)$.

As Fig. 4(b) shows, applying δ_{HFC} leads to a lower output of the labeled class while applying δ_{LFC} leads to a higher output of other classes. The difference is similar to the difference between δ_{PFC} and δ_{NFC} in Sec. 4.4.1 but less significant.

4.4.3 A Closer Look On the Relationship between HFC, LFC and PFC, NFC

Recall that in Fig. 3(a), with ST models being attacked, most NFC reside in HFC of data while PFC gather around LFC. However, results in Sec. 4.4.1 4.4.2 imply that the attack exhibits some bias towards the frequency domain: positive parts (PFC) of HFC are more easily to be attacked and so does negative parts (NFC) of LFC, which seems to contradict with Fig. 3(a). To further explain the intriguing conflict, we further calculate the ratio of PFC and NFC over the frequency components of the same frequency. As shown in Fig. 5, both PFC and NFC take up a fairly certain ratio over the whole frequency spectrum, which shows that the contribution of PFC (NFC) related to different frequency basis varies a lot. Therefore, the existence of PFC in high frequency and NFC in low frequency may lead to the result in Sec. 4.4.2. We argue that because attacks need meet the loss maximization goal within the given budget, it may select frequency components that are more vulnerable to disturb. So PFC in high frequency and NFC in low frequency could be the "sweet spot" for attacks, both of which occupy a relatively smaller portion than the other in Fig. 5.

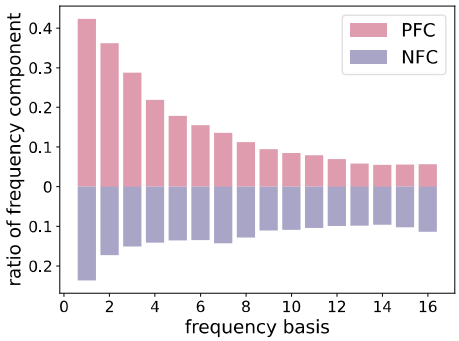


Figure 5: Ratio of the component PFC and NFC over frequency. Results are averaged over clean examples on ST model on CIFAR10.

4.5 Fairness of AT in Frequency Domain

The disparity of accuracy and robustness of AT model is a troublesome phenomenon even on the balanced dataset such as CIFAR-10 [34, 29] which results in fairness concerns. As shown in Fig. 6, while there exists no obvious class-wise disparity in ST model, both test accuracy and robustness of AT model vary a lot between classes. For CIFAR10, we find that the class 0, 1, 8, 9 that belong to the same coarse class "Transportation" exhibit relatively high accuracy and robustness while other classes related to "Animals" are less robust except the class 7 "horse".

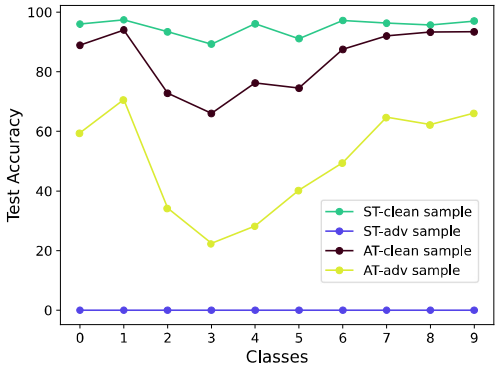
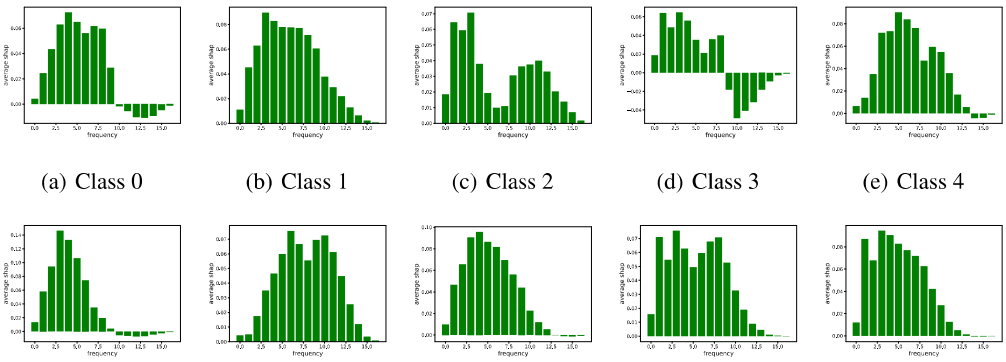


Figure 6: Test Accuracy of ST model and AT model on clean and adversarial examples in different classes from CIFAR10

Recall that in Sec. 4.2, we argue that for ST model, adversarial attacks succeed probably by enlarging the negative impact of HFC while in Sec. 4.3, we assume that AT defends against such attacks by filtering out negative HFC. The above findings motivate us to investigate the class-wise distribution of frequency components. Since adversarial attack would change the distribution of data frequency domain, we choose clean data and ST model as our subject. In specific, we calculate the average Shapley value of clean data samples from each class on ST model and display it over a frequency basis. The results are shown in 7. Though the class-wise accuracy of ST are similar,



(a) Class 0 (b) Class 1 (c) Class 2 (d) Class 3 (e) Class 4

(f) Class 5 (g) Class 6 (h) Class 7 (i) Class 8 (j) Class 9

Figure 7: Average Shapley value of different frequencies of clean samples on ST model in different classes from CIFAR10

we find that ST's prediction on relatively robust classes heavily relies on the LFC (*e.g.* class 1, 7, 8 and 9), while non-robust classes either have high positive average Shapley value on HFC (*e.g.* class 2) or even negative one on HFC (*e.g.* class 3,4 and 5), which means attacks can easily intervene them by disturbing HFC. As AT tends to filter out HFC for robustness, we give our assumptions about the fairness problem on AT: *classes that rely on HFC either positively or negatively would have relatively low accuracy and robustness under AT.*

Note that class 0 is an exception as its robustness is relatively well and we assume that since class 0, 1, 8, 9 belong to the same coarse class "Transportation" and the three classes are relatively robust, it is harder for attacks to utilize negative HFC of class 0 to flip models' decision to any of 1, 8 and 9.

5 Improving robustness via Class-wise Shapley value-guided Augmentation

Motivation. As discussed in Sec. 4.2 and Sec. 4.3, we argue that AT model is more robust than ST model because AT has less NFC due to its filtering effect. Moreover, recall the definition in Sec. 3, frequency components with negative Shapley value make model output lean to other classes rather than the labeled class, which implies that the distribution of image features composed by NFC is mistakenly aligned with that of other classes. Intuitively, by proactively correcting the misalignment of features of the model, we could improve the adversarial robustness for future attacks. What's more, our observations in Sec. 4.5 reveal that there exists obvious disparity at the class level of distribution of NFC. Therefore, we propose to actively augment data with features composed by NFC of the same class, named Class-wise Shapley value-guided Augmentation (CSA). To be specific, based on a CNN f, we obtain the CSA p_X^f of data X by masking its PFC as follows, with the indicator function $\mathbb{I}[\cdot]$:

$$p_X^f(u,v) = \mathbb{I}[\phi_{u,v}^{X,f} < 0] \cdot \mathcal{F}(X)(u,v) \tag{13}$$

Implementation The last question remaining is the time complexity of CSA. Since it takes a large number of samplings for obtaining a stable Shapley value (c.f. Eq. 5) for each sample, it is necessary to downgrade the time complexity. Fortunately, we find that for the same class, the feature distribution composed by NFC aligns well between samples, and depending on only a small ratio of whole data (3% in our experiment) brings out obvious improvement. Detailed analyses at a statistical level and experimental level can be found in Appendix B.

For each class c, we randomly take data samples $\langle X_1, c\rangle, \langle X_2, c\rangle, \cdots, \langle X_n, c\rangle$ and generate a set of CSAs with a pretrained model $\hat{f}$ as: $\mathcal{P}_c^{\hat{f}} = \{p_{X_1}^{\hat{f}}, p_{X_2}^{\hat{f}}, \cdots, p_{X_n}^{\hat{f}}\}$ and train our model as follows:

$$\theta = \arg\min_{\theta} \mathbb{E}\left[\max_{\delta < \|\epsilon\|_p} \ell\left(f[\mathcal{F}^{-1}(\mathcal{F}(X) + \alpha * p^{\hat{f}}) + \delta], y\right)\right], p^{\hat{f}} \in \mathcal{P}_y^{\hat{f}} \tag{14}$$

where the α is a coefficient of the CSA $p^{\hat{f}}$.

Table 1: Testing accuracy of ResNet-18 trained on CIFAR10.

Methods		Natural Accuracy			PGD-20			AutoAttack		
		Best	Last	Diff	Best	Last	Diff	Best	Last	Diff
PGD-AT [20]	baseline	**83.83%**	**84.49%**	**-0.66**	49.87%	46.38%	3.49	46.52%	44.06%	2.46
	CSA	81.57 %	83.11 %	-1.54	**51.13%**	**49.42%**	**1.71**	**47.17%**	**46.56%**	**0.61**
TRADES [39]	baseline	**81.23%**	**81.71%**	**-0.48**	51.18%	51.08%	**0.10**	47.94%	47.74%	0.20
	CSA	80.84%	81.62%	-0.78	**52.57%**	**52.06%**	0.51	**48.94%**	**49.16%**	**-0.22**

Experiment Results We conduct experiments on CIFAR10 to verify the effectiveness of our CSA. We train a ResNet-18 on CIFAR10 under PGD-AT [20] and TRADES [39]. Training details can be found in Appendix C. Attacks are crafted within ℓ_∞ bound with $\epsilon = 8/255$. We select PGD-20 attack [20] and AutoAttack (AA.) [10] as the attack baselines. As Table. 1 shows, under various attacks, our CSA consistently achieves better robustness over defense baselines with little cost of natural accuracy. Note that the CSA set of each class contains the NFC of the first 100 images without shuffling, *i.e.*, 3% of the training set, which shows that obtaining CSA only takes little overhead and the feasibility of our method in realistic settings.

6 Related Work

Adversarial Attack Since the seminal works [28, 14] have been proposed that reveal the vulnerability of DNNs, there is an increasing interest in studying stronger adversarial attacks, such as FGSM [14], C&W attack [4], PGD attack [20], and AutoAttack [10]. The robustness of CNNs in this work has been thoroughly evaluated on these above attack baselines.

Adversarial Defense The development between attack and defense is just like an arms race: the existence of adversarial examples promotes the development of defense methods, among which adversarial training (AT) has been considered the most effective one [3]. However, AT has its own limitations: i) the trade-off between accuracy and robustness [25, 30, 38, 35], ii) *robust overfitting* [22, 8], which brings a large robustness generalization gap. Many works have been proposed to improve AT by sample-wise importance re-weighting [40] or adding more unlabeled data [21, 5]. In this work, we pave a new way of class-specific Shapley value-based data augmentation towards better robustness, which is orthogonal to the above strategies.

Understanding Robustness in Frequency Domain One way to inspect the intriguing nature of adversarial attack and defense is through the frequency perspective. Many works explore the sensitivity of CNNs and frequency properties of the model [31, 36, 1] and study whether convolution itself has an intrinsic bias in frequency domain [6]. A hypothesis is that CNNs exploit HFC which leads to the lack of robustness [32]. Driven by this hypothesis, many pre-processing defend methods have been proposed [11, 18, 41]. Meanwhile, some works also come up with adversarial attacks focusing on LFC [15, 24]. Efforts to quantify and inspect the contribution of each frequency component have been made [33]. We employ Shapley value, a tool that has preferable properties, to quantify the contribution of each frequency component on the fine-grained level.

Explainability with Shapley Value The Shapley Value [23] is proposed to distribute contributions among players in a game. The Shapley value has been widely applied for the explainability of deep learning. Some works propose approximations of Shapley value [19, 7, 2]. Some works focus on the condition to apply Shapley value [12, 13]. And some works focus on the interaction between input features [27, 26, 37]. In our work, we apply the Shapley value in the frequency domain to quantify the contribution of frequency components. The quantification have many desirable theoretical properties.

7 Conclusion & Limitations

In this work, we have made efforts to quantify the contribution of each frequency component on CNNs. We employ Shapley value, a method with many theoretically desirable properties, to quantify the contribution of a single frequency component on the instance-level. Through our quantification, we explore the difference between standard trained (ST) model and adversarial trained(AT) model and examine the attack effect of adversarial noises on different frequency components. Our analysis of the fairness in AT provides insights of the cause of robustness difference between classes. Based on our findings, we further propose a simple yet effective class-wise data augmentation method CSA. We augment data with the NFC that we found in this class. Experimental results have shown the effectiveness of our method. Possible limitations are our current results are limited mainly to CNNs.

References

[1] A. A. Abello, R. Hirata, and Z. Wang. Dissecting the high-frequency bias in convolutional neural networks. In *CVPR*, 2021.

[2] M. Ancona, C. Öztireli, and M. H. Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 272–281. PMLR, 2019.

[3] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*. PMLR, 2018.

[4] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE Computer Society, 2017.

[5] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang. Unlabeled data improves adversarial robustness. In *NeurIPS*, volume 32, 2019.

[6] J. O. Caro, Y. Ju, R. Pyle, S. Dey, W. Brendel, F. Anselmi, and A. Patel. Local convolutions cause an implicit bias towards high frequency adversarial examples, 2020.

[7] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. In *ICLR (Poster)*. OpenReview.net, 2019.

[8] T. Chen, Z. Zhang, S. Liu, S. Chang, and Z. Wang. Robust overfitting may be mitigated by properly learned smoothening. In *ICLR*, 2020.

[9] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. Robustbench: a standardized adversarial robustness benchmark. In *NeurIPS Datasets and Benchmarks*, 2021.

[10] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[11] N. Das, M. Shanbhogue, S. Chen, F. Hohman, S. Li, L. Chen, M. E. Kounavis, and D. H. Chau. SHIELD: fast, practical defense and vaccination for deep learning using JPEG compression. In *KDD*, pages 196–204. ACM, 2018.

[12] C. Frye, D. de Mijolla, T. Begley, L. Cowton, M. Stanley, and I. Feige. Shapley explainability on the data manifold. In *ICLR*. OpenReview.net, 2021.

[13] C. Frye, C. Rowat, and I. Feige. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. In *NeurIPS*, 2020.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR (Poster)*, 2015.

[15] C. Guo, J. S. Frank, and K. Q. Weinberger. Low frequency adversarial perturbation. In *UAI*, volume 115 of *Proceedings of Machine Learning Research*, pages 1127–1137. AUAI Press, 2019.

[16] I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. A. Friedler. Problems with shapley-value-based explanations as feature importance measures. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 5491–5500. PMLR, 2020.

[17] Z. Lin, Y. Gao, and J. Sang. Investigating and explaining the frequency bias in image classification. In *IJCAI*, 2022.

[18] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, and W. Wen. Feature distillation: Dnn-oriented JPEG compression against adversarial examples. In *CVPR*, pages 860–868. Computer Vision Foundation / IEEE, 2019.

[19] S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. In *NIPS*, pages 4765–4774, 2017.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR (Poster)*. OpenReview.net, 2018.

[21] A. Najafi, S.-i. Maeda, M. Koyama, and T. Miyato. Robustness to adversarial perturbations in learning from incomplete data. In *NeurIPS*, volume 32, 2019.

[22] L. Rice, E. Wong, and Z. Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020.

[23] L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games*, page 307–317, 1953.

[24] Y. Sharma, G. W. Ding, and M. A. Brubaker. On the effectiveness of low frequency perturbations. In *IJCAI*, pages 3389–3396. ijcai.org, 2019.

[25] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, 2018.

[26] M. Sundararajan, K. Dhamdhere, and A. Agarwal. The shapley taylor interaction index. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 9259–9268. PMLR, 2020.

[27] M. Sundararajan and A. Najmi. The many shapley values for model explanation. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 9269–9278. PMLR, 2020.

[28] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR (Poster)*, 2014.

[29] Q. Tian, K. Kuang, K. Jiang, F. Wu, and Y. Wang. Analysis and applications of class-wise robustness in adversarial training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1561–1570, 2021.

[30] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[31] Y. Tsuzuku and I. Sato. On the structural sensitivity of deep convolutional networks to the directions of fourier basis functions. In *CVPR*, pages 51–60. Computer Vision Foundation / IEEE, 2019.

[32] H. Wang, X. Wu, Z. Huang, and E. P. Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *CVPR*, pages 8681–8691. Computer Vision Foundation / IEEE, 2020.

[33] Z. Wang, Y. Yang, A. Shrivastava, V. Rawal, and Z. Ding. Towards frequency-based explanation for robust CNN. *CoRR*, abs/2005.03141, 2020.

[34] H. Xu, X. Liu, Y. Li, A. Jain, and J. Tang. To be robust or to be fair: Towards fairness in adversarial training. In *International Conference on Machine Learning*, pages 11492–11501. PMLR, 2021.

[35] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. R. Salakhutdinov, and K. Chaudhuri. A closer look at accuracy vs. robustness. In *NeurIPS*, 2020.

[36] D. Yin, R. G. Lopes, J. Shlens, E. D. Cubuk, and J. Gilmer. A fourier perspective on model robustness in computer vision. In *NeurIPS*, 2019.

[37] H. Zhang, Y. Xie, L. Zheng, D. Zhang, and Q. Zhang. Interpreting multivariate shapley interactions in dnns. In *AAAI*, pages 10877–10886. AAAI Press, 2021.

[38] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[39] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. El Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019.

[40] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021.

[41] Z. Zhang, C. Jung, and X. Liang. Adversarial defense by suppressing high-frequency components. *CoRR*, abs/1908.06566, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]