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 generalization limitation of AT including the fairness problem; iii) the attack bias at different frequency component level. Finally, we propose a Shapley-value guided data augmentation technique for improving the robustness. Experimental results on image classification benchmarks show its effectiveness.

1 Introduction

Though convolutional neural networks (CNNs) have shown their great generalization power in various vision tasks, the existence of adversarial examples [32, 16, 22] show that CNNs are prone to be affected by adversarial noises that are imperceptible to humans. As inspecting the impact of frequency components, *i.e.* signals of a certain frequency in the frequency domain, one popular hypothesis about the brittleness of CNNs is that CNNs can exploit high frequency components (HFCs) that are not perceivable to humans [36, 37, 1, 19]. Based on this hypothesis, a line of works propose to utilize HFC filtering to improve the attack or defense methods [13, 20, 45]. However, towards the above *frequency bias* phenomenon, most works [36, 37, 19] intuitively divide the image frequency spectrum into two parts as low frequency components (LFCs) and HFCs then evaluate the contribution of each part based on the performance of models on the dataset, overlooking the possible disparity in sample-level and frequent component-level. Moreover, current works [17, 28] reveal that some successful adversarial attacks also depend on LFCs, which reveals the limitation of previous works: i) overlooking the contribution of each individual frequency component in LFCs and HFCs; ii) only do qualitative dataset-level analysis.

Previous work [37] quantify and inspect the contribution of each frequency component by the change of model prediction brought by masking frequency components *i.e.* replace the frequency component with zero value. Following the spirit, in this paper we resort to a more sophisticated tool, Shapley

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

value [27] to quantify the contribution of each frequency component. Specifically, Shapley value is a metric from cooperative game theory which has been widely used in explainable machine learning [9, 2, 21, 31]. 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 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 average expected difference between the model's output brought by the absence of the frequency component regarding all possible circumstances where other frequency components are either masked or not. An illustration of the quantifying method is presented in Fig. 1. Through our quantification, one is able to inspect the contribution to the classification task of different frequency components of each sample (instead of dataset level). 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).

We inspect the impact brought by different frequency components. At the instance level, we present intriguing disparity between different data samples, *e.g.* model may rely more on HFCs to inference on some instances, as shown in the first row of Fig. 2, or rely more on LFCs to inference on the other instances, as shown in the second row of Fig. 2. At the dataset level, we inspect and analyze: i) the cause of the adversarial robustness achieved by adversarial training; ii) the fairness problem in adversarial training (AT); iii) the attack bias on different frequency components. We further propose a simple yet principled data augmentation method named Class-wise Shapley value-guided Augmentation (CSA) to correct the misalignment of features between different classes. By augmenting the data sample with features composed by NFCs of other data samples from the same class, the method boosts the adversarial robustness efficiently. Our major contributions are as follows:

- Instead of inspecting CNNs based on the manually divided frequency spectrum as done in existing literature [36], we introduce the Shapley value metric to the frequency domain and directly quantify the contribution of each individual frequency component on CNNs, with the advantage of desirable theoretical properties from the Shapely value itself (Section 3). To our knowledge, this is the first work to apply the Shapley value to the frequency domain on DNNs (specifically CNNs in this paper).

- As inspecting the impact of frequency components on models, we address the possible cause of adversarial robustness achieved by adversarial training(Sec. 4.2) and the fairness problem in the adversarial problem in adversarial training(Sec. 4.3). We quantitatively prove that HFCs play an important role in adversarial training.

- We inspect the impact of different frequency components on adversarial attacks and demonstrate that adversarial attacks on PFCs and HFCs are relatively more effective than attacks on NFCs and LFCs. We further demonstrate that contribution of frequency components is not necessarily determined by its frequency and provide a possible explanation for successful adversarial attacks focusing on LFCs [17, 28](Sec. 4.4).

- Motivated by our explanatory results, we assume that NFCs compose features that are not correctly exploited by the model. Therefore we propose to augment data with NFCs found on data samples of its class, entitled by Class-wise Shapley value-guided Augmentation (CSA), which successfully boosts robustness in our empirical studies (Section 5).

2 Notations and Preliminaries

We present our model in the context of image classification, while we think the concept can still be applied to other backbones e.g. GNNs.

Notation: $\langle \mathbf{X}, \mathbf{y} \rangle$ denotes a data sample where $\mathbf{X} \in \mathbb{R}^{d_1 \times d_2}$ is an image and $\mathbf{y} \in \{1, \dots, C\}$ is its label where C is the number of classes. For simplicity, we omit the channel of images. $f(\cdot, \theta) : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^C$ 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. $\mathcal{F}(\cdot) : \mathbb{R}^{d_1 \times d_2} \to \mathbb{C}^{d_1 \times d_2}$ denotes the Discrete Fourier Transfom (DFT). $\mathcal{F}^{-1}(\cdot) : \mathbb{C}^{d_1 \times d_2} \to \mathbb{R}^{d_1 \times d_2}$ denotes the inverse Discrete Fourier Transform.

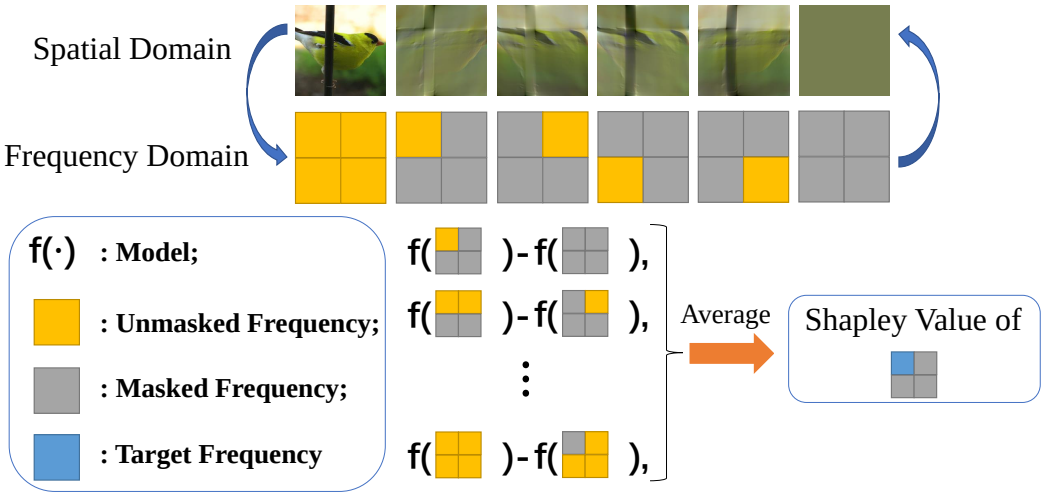


Figure 1: Illustration of the Shapley value calculated in frequency domain when the frequency components are devided into 2×2 patches. The Shapley value of a frequency component is the average marginal effect that the frequency component contributes to the output of the model.

Discrete Fourier Transform (DFT). DFT transforms finite signals into complex-valued functions of frequency. The 2D-Discrete (for image data considered in this paper) Fourier Transform is given by:

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

The inversion of DFT is computed as:

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

Adversarial Attack. It aims to find a small noise δ bounded in ℓ_p space to cause failure [32, 16].

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

Defense. Adversarial training [22] becomes the most effective defense method [3], which is typically conducted by minimizing loss directly on adversarial training data.

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

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 $V : 2^{\mathcal{N}} \to \mathbb{R}$ 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 [27] 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.

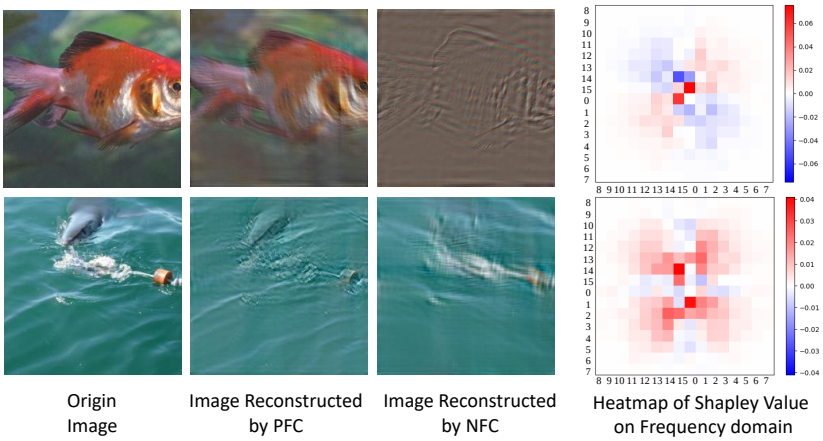


Figure 2: Shapley value based quantification of two example images in ImageNet. Left to right: origin image, image reconstructed with positive frequency components (PFC), images reconstructed with negative frequency components (NFC), the heat map of Shapley value in frequency domain. Note that the heatmap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

3 Shapley Value-based Quantification

3.1 Contribution quantification for frequency component

We assume that 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, we firstly construct a cooperative game in this scenario. Considering frequency components as players, we define the set of players for data sample $\langle \mathbf{X}, \mathbf{y} \rangle$ as:

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

Note that $\mathcal{F}(\mathbf{X})(0,0)$ 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:

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

where $f(\cdot)_{\mathbf{y}}$ denotes the $\mathbf{y}$-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:

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

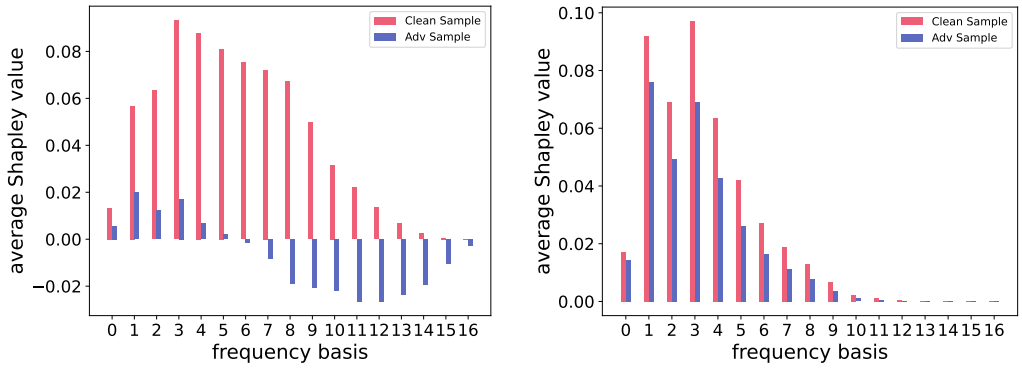
where $\mathbb{I}(\cdot)$ denotes an indicator that output 1 when the condition is satisfied and 0 otherwise. Specifically, $\sigma_S(0,0) = 1$. The corresponding characteristic function of the game is defined as:

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

By definition, we calculate the Shapley value $\phi_{u,v}^{\mathcal{N}_{\mathbf{X}},f}$ of frequency component $\mathcal{F}(\mathbf{X})(u,v)$ 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, which takes desirable properties including *Null player*, *Efficiency*, *Symmetry* and *Linearity*.

3.2 Justification for frequency based Shapley value on CNNs

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 [9, 2, 21, 31]. 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 CNNs.



(a) ST model (b) AT model

Figure 3: Average shapley value of frequency componets in clean samples and adversarial samples over frequency basis of ResNet18 on CIFAR10.

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

$$f_{conv}(\mathbf{X}) = \sum_{u=0}^{d_1-1} \sum_{v=0}^{d_2-1} \left[\mathcal{F}(\mathbf{X})(u,v) f_{conv} \left[\mathcal{F}^{-1}(\mathbf{I}_{d_1 \times d_2}^{u,v}) \right] \right] \tag{10}$$

where $\mathbf{I}_{d_1 \times d_2}^{u,v} \in \mathbb{R}^{d_1 \times d_2}$ contains zeros except the element on the u-th row and v-th column is one. According to Remark 1, the output of a convolution operation is a linear function *w.r.t.* 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 [18]. 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 more details.

4 Rethinking Robustness in Frequency Domain

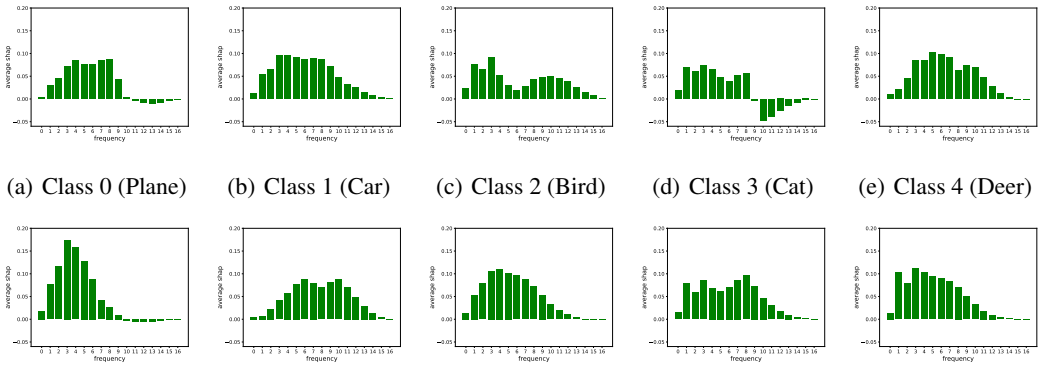
In this section, we conduct extensive experiments to give a fundamental analysis of the contributions of different frequency components. In Sec. 4.1, we introduce details and results of instance-level sampling for Shapley value calculation. In Sec. 4.2, we inspect the contribution of frequency components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

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 [27]. We sample the Shapley value on CIFAR10, CIFAR100, Tiny ImageNet and ImageNet.

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 reconstructed image from positive frequency components (PFCs), the reconstructed image from negative frequency components (NFCs), and the heatmap of Shapley value 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 PFCs are mostly low frequency. 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": PFCs 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.



(a) Class 0 (Plane) (b) Class 1 (Car) (c) Class 2 (Bird) (d) Class 3 (Cat) (e) Class 4 (Deer)

(f) Class 5 (Dog) (g) Class 6 (Frog) (h) Class 7 (Horse) (i) Class 8 (Ship) (j) Class 9 (Truck)

Figure 4: Average Shapley value of different frequencies of clean samples on ST model in different classes from CIFAR10.

4.2 The Cause of the Adversarial Robustness of AT model

Observations in Sec. 4.1 are based on clean examples. We further inspect the difference between ST and AT model in the frequency domain based on our Shapley value metric. For CIFAR10, adversarial examples are generated with PGD-20 with a Linf bound of $8\backslash255$ and step size of $2\backslash255$. We present the experiment results of ResNet-18 on CIFAR10. Refer to Appendix B for more experiment results of vgg16 on CIFAR10 and ResNet-18 on CIFAR100 and refer to Appendix A for more details of the model training.

4.2.1 Standard Trained Models Mainly Exploit LFC but Are More Vulnerable on HFC

As shown in Fig. 3(a), first, for clean examples, LFCs have significant positive Shapley value on average for standard trained (ST) model while the average Shapley value of HFCs are nearly zero, which means ST model barely exploits HFC; Second, for adversarial examples, the mean Shapley value of HFC are significantly negative, which shows that ST model is more vulnerable on HFC.

4.2.2 Adversarial Trained Models Improve Robustness as an HFC Filter

As shown in Fig. 3(b), LFCs have high positive Shapley value on both clean and adversarial examples and the average Shapley value of HFCs are nearly zero on both clean and adversarial examples, which may be the key to improved robustness of AT. We hypothesize that AT tends to filter out the negative impact of HFC and the outcome is that AT focuses more on LFC, overlooking information in HFC. As Fig. 3(a) shows, the HFC of clean images plays a remarkable positive impact. Thus we conjecture that the trade-off between the clean accuracy and the robust accuracy of AT is possibly the result of whether the model utilizes the information in HFCs.

4.3 Fairness of AT in Frequency Domain

The class-wise disparity of robustness of AT model is a troublesome phenomenon even on the balanced dataset such as CIFAR-10 [38, 33] which results in fairness concerns. For CIFAR10, 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".

To investigate the difference between different classes, we demonstrate the average Shapley value of clean data samples from each class for ST model in Fig. 4. Though the class-wise accuracy of ST are similar, we generally find the trend that the absolute Shapley value of HFC is relatively larger on non-robust classes (*e.g.* class 2,3,4 and 5) than robust classes (*e.g.* class 1, 7, 8 and 9).

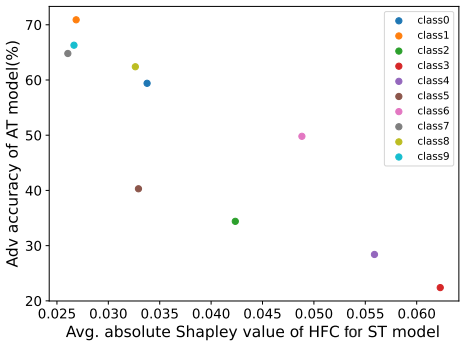


Figure 5: The negative relationship between the adversarial accuracy of AT model and the average absolute Shapley value of HFC for ST model.

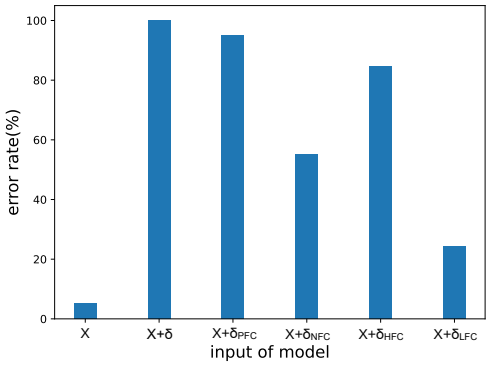


Figure 6: The error rate of ST model on CIFAR10 with different input. X represent origin data. δ is the adversarial noise generated against the model. δ_{PFC}, δ_{NFC}, δ_{HFC} and δ_{LFC} are the masked adversarial noise that are constrained on PFC, NFC, HFC and LFC.

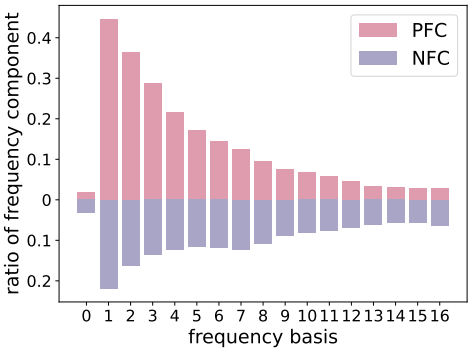


Figure 7: Ratio of the component PFC and NFC with significantly large Shapley value (absolute value greater than 20% of the maximum absolute Shapley value of the frequency). Results are averaged over clean examples on ST model (ResNet18) on CIFAR10.

We further test the adversarial accuracy of AT model and calculate the average absolute Shapley value of HFC (we intuitively take components with frequency higher than 70% of the largest frequency) for ST model. As shown in Fig. 5, there is a negative relationship between the adversarial accuracy of AT model and the average absolute Shapley value of HFC for ST model (the Pearson correlation coefficient is -0.8765). As AT tends to filter out HFCs for robustness, we conjecture that: *classes, where HFCs have a larger impact on the ST model, would have relatively lower accuracy and robustness under AT.*

4.4 Attack Bias in Frequency Domain

In this section, we inspect the performance of adversarial attacks over different frequency components. Specifically, for an adversarial noise δ generated by Eq. 3 on ST model f and data sample $\langle \mathbf{X}, \mathbf{y} \rangle$, we transform the noise δ into frequency domain as $\mathcal{F}(\delta)$, which is further applied by masks based on different metrics to determine the impact of adversarial noise on different frequency components.

4.4.1 Adversarial Attacks on PFCs and HFCs Are More Effective

We first build masks based on the division between PFCs and NFCs of the data sample and the masked noise is defined as follows, where $\odot$ is Hadamard product:

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

Similar to Eq. 11, we define adversarial noises δ on HFCs and LFC as

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

where $r(u,v)$ is the frequency of the frequency component, r_{max} is the maximum frequency in this data sample, ϵ_{low} and ϵ_{high} is the intuitively determined threshold for low frequency and high frequency. In our experiment, $\epsilon_{low} = 0.4$, $\epsilon_{high} = 0.6$.

We test the error rate of ST model as the input data sample is attacked with various noises, as shown in Fig. 6. The error rate is nearly zero for origin data sample X and nearly one hundred percent for data samples attacked with the unmasked noises $X + \delta$. For adversarial noise on PFCs and NFCs, $X + \delta_{PFC}$ reaches a similar error rate of ST model as $X + \delta$ which is significantly larger than the error rate with $X + \delta_{NFC}$. It indicates that the adversarial attack on PFCs is more effective. Similar for HFCs and LFCs, the adversarial attack on HFCs is more effective than NFCs.

4.4.2 A Closer Look on the Relationship between HFC, LFC and PFC, NFC

As demonstrated in Fig. 3(a), with ST models being attacked, most NFCs reside in the HFCs of data while PFCs reside in the LFCs of data. However, results in Sec. 4.4.1 imply that the attack exhibits

some bias in the frequency domain: the attack effect of adversarial noises on PFCs is better than the adversarial noises on NFCs, and so for the adversarial noises on NFCs and LFCs, which seems to contradict Fig. 3(a) and demands a closer look. To explain such an intriguing conflict, we further calculate the ratio of PFCs and NFCs for each frequency over the frequency spectrum. As shown in Fig. 7, both PFCs and NFCs that have a significant impact take up a fair ratio (at least 0.1) of each frequency over the whole frequency spectrum. Therefore, the existence of PFC in high frequency and NFC in low frequency may lead to the result in Sec. 4.4.1.

The existence of PFCs and NFCs over the whole frequency spectrum also provides a possible explanation for the conflict between the hypothesis that exploiting HFCs causes the brittleness of CNNs and the success of the adversarial attacks focusing on LFCs [17, 28] which is addressed in [4, 23]. Since the ratio of PFCs and NFCs are even higher at low frequency, it is possible for adversarial attacks that focus on LFCs to reduce the impact of PFCs and boost the impact of NFCs.

5 Improving Robustness via Class-wise Shapley Value-guided Augmentation

Motivation. As defined in Sec. 3, frequency components with negative Shapley value make the model output lean to other classes rather than the ground-truth 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. Moreover, our observations in Sec. 4.3 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 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 an 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 $p^{\hat{f}}$.

Experiment Results We conduct experiments on CIFAR10 to verify the effectiveness of our CSA. We train models on CIFAR10, CIFAR100 and TinyImageNet under PGD-AT [22] and TRADES [43]. Training details can be found in Appendix C. Attacks are crafted within ℓ_∞ bound with $\epsilon = 8/255$. We select PGD-20 attack [22] and AutoAttack (AA) [12] 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. See Appendix B for more experiment results. Note that the CSA set of each class contains the NFC of the first 150 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 [32, 16] have been proposed that reveal the vulnerability of DNNs, there is an increasing interest in studying stronger adversarial attacks, such as FGSM [16], C&W attack [5], PGD attack [22], and AutoAttack [12]. 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

Table 1: Testing accuracy of models trained on CIFAR10 with CSAs generated with ResNet-18. 'Final' denotes the corresponding performance at the last training epoch, while 'best' refers to the best snapshot model during training.

(a) ResNet18

Methods		Natural Accuracy w/o attack			attacked by PGD-20			attacked by AutoAttack		
		Best	Final	Diff	Best	Final	Diff	Best	Final	Diff
PGD-AT [22]	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 [43]	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**

(b) WRN-28-10

Methods		Natural Accuracy w/o attack			attacked by PGD-20			attacked by AutoAttack		
		Best	Final	Diff	Best	Final	Diff	Best	Final	Diff
PGD-AT [22]	baseline	**85.90%**	85.67%	0.23	50.18%	47.51%	**2.67**	47.86%	45.69%	2.17
	CSA	85.48 %	**85.87 %**	**-0.39**	**53.00%**	**50.83%**	3.50	**49.61%**	**48.75%**	**0.86**
TRADES [43]	baseline	84.13%	84.73%	**-0.60**	53.17%	48.72%	4.45	50.21%	46.65%	3.56
	CSA	**84.41%**	**84.92%**	-0.51	**54.32%**	**50.53%**	**3.79**	**51.40%**	**48.65%**	**2.75**

limitations: i) the trade-off between accuracy and robustness [29, 34, 42, 39], ii) *robust overfitting* [26, 10], which brings a large robustness generalization gap. Many works have been proposed to improve AT by sample-wise importance re-weighting [44] or adding more unlabeled data [25, 6]. 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 lens of frequency spectrum. Many works explore the sensitivity of CNNs and frequency properties of the model [35, 40, 1] and study whether convolution itself has an intrinsic bias in frequency domain [7]. A hypothesis is that CNNs exploit HFCs which leads to the lack of robustness [36]. Driven by this hypothesis, many pre-processing defend methods have been proposed [13, 20, 45]. Meanwhile, some works also come up with adversarial attacks focusing on LFC [17, 28] which leads to assumptions that constituent frequencies of adversarial examples are dependent on the dataset [4, 23]. Efforts to quantify and inspect the contribution of each frequency component have been made [37, 19]. 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 Shapley Value [27] is proposed to distribute contributions among players in a game. It has been widely applied for the explainability of deep learning. Some works propose its approximations [21, 9, 2]. While another line of works focus on the condition to apply Shapley value [14, 15]. Also, the interaction between input features [31, 30, 41] is recently studied. In our work, we apply 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 established 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. Our analysis of the fairness in AT provides insights of the cause of robustness difference between classes. We further examine the attack effect of adversarial noises on different frequency components. Based on our findings, we further propose a simple yet effective class-wise data augmentation method CSA which augments data with the NFC that we found in this class. Experimental results have shown the effectiveness of our method. Note that our current results are limited mainly to CNNs while we believe it is also rewarding to develop new techniques to other backbone e.g. Transformer and GNNs based on our framework, which we leave for immediate future work. We hope that this work could inspire future explanations and defense algorithms of adversarial attacks. Due to the nature of this work, there may not be any potential negative social impact that is easily predictable.

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] R. Bernhard, P. Moëllic, M. Mermillod, Y. Bourrier, R. Cohendet, M. Solinas, and M. Reyboz. Impact of spatial frequency based constraints on adversarial robustness. In *IJCNN*, pages 1–8. IEEE, 2021.

[5] 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.

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

[7] 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.

[8] J. Castro, D. Gómez, and J. Tejada. Polynomial calculation of the shapley value based on sampling. *Comput. Oper. Res.*, 36(5):1726–1730, 2009.

[9] 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.

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

[11] 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.

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

[13] 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.

[14] 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.

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

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

[17] 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.

[18] 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.

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

[20] 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.

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

[22] 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.

[23] S. R. Maiya, M. Ehrlich, V. Agarwal, S. Lim, T. Goldstein, and A. Shrivastava. A frequency perspective of adversarial robustness. *CoRR*, abs/2111.00861, 2021.

[24] S. Maleki, L. Tran-Thanh, G. Hines, T. Rahwan, and A. Rogers. Bounding the estimation error of sampling-based shapley value approximation with/without stratifying. *CoRR*, abs/1306.4265, 2013.

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

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

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

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

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

[38] 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.

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
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]

In this Supplementary Material, we first present details of the Shapley value sampling (Appendix A). Then we give more experimental results on CIFAR-100 and stability analysis of Shapley value (Appendix B). We further show the training details and ablation study for the sample complexity of our CSA (Appendix C). Finally, we add properties of the Shapley value and proof of decomposition of CNNs in frequency domain (Appendix D).

A Details of the Shapley Value Sampling

In this section, we introduce the details of the Shapley value sampling.

A.1 Details of the Model for the Shapley Value Sampling

We sample the Shapley value for models trained on CIFAR10, CIFAR100 and ImageNet. For CIFAR10 and CIFAR100, we employ ResNet-18 and train them ourselves. The ST model on CIFAR10 and CIFAR100 is trained for 100 epochs with SGD optimizer where learning rate is set to be 0.1 and decayed by 0.1 at 75-th epoch and 90-epoch. The momentum is set to be 0.9 and weight decay is set to be $2e-4$. The AT model on CIFAR10 and CIFAR100 is trained with adversarial examples generated with PGD-10 [22] with a L-inf bound where epsilon is set to be $8/255$ and stepsize is set to be $2/255$. The other settings of the training of AT model are the same of the ST model. For ImageNet, we employ standard ResNet-50 provided in robustbench [11].

A.2 Details of the Sampling of the Shapley Value

We employ Monte-Carlo sampling method for the Shapley value [8] to sample the Shapley value. For CIFAR10 and CIFAR100, we divide the image into 16×16 patches in the frequency domain where each patch contains 2×2 frequency components and sample 2000 times for each data sample. For ImageNet, we divide the image into 32×32 patches in the frequency domain where each patch contains 7×7 frequency components and sample 1000 times for each data sample.

A.3 Analysis about the Error Bound of our Shapley Value Sampling

As discussed in previous work [24], once we know the variance of the marginal contributions we have an bound for the estimation error of the Shapley Value. For our estimated Shapley value $\hat{\phi}$, assume the probability of the estimation error being greater than ϵ is less than δ that is

$$Pr(\|\hat{\phi} - \phi\| \geq \epsilon) \leq \delta \tag{15}$$

With Chebyshev's inequality, for marginal contributions with variance as σ^2, the required sample times should be $m \geq \lceil \frac{\sigma^2}{\delta \epsilon^2} \rceil$

We calculate the average variance of the marginal contribution of each patch of the images in the first 200 images of each class of the train set of CIFAR10. The average variance is 1.814×10^{-3}. With $\delta = 0.05$ and $\epsilon = 0.005$ the required sample times is around 1450. Therefore, we set the sample times on CIFAR10 and CIFAR100 as 2000. Limited by the calculation complexity, we set the sampling times for ImageNet at 1000.

A.4 Shapley Value Quantification Results

We demonstrate more Shapley Value Quantification Results on ImageNet and TinyImageNet in Tab. 8 and Tab. 9. Each row contains results of a data sample. We display the origin image, image reconstructed with PFCs, image reconstructed with NFCs and the heatmap of Shapley value from left to right. Note that the for the heatmap of Shapley value, red represents positive and blue represents negative. The closer to the center the lower the frequency.

We report the time consumed for sampling Shapley value for ResNet-18 on CIFAR10 with Intel(R) Core(TM) i7-7820X CPU @ 3.60GHz and one GeForce RTX 2080 Ti in Tab. 2.

B Extra Experiment Results

As demonstrated in Tab. 3, for l_∞ attack, suppressing high frequency components even slightly degrade the testing accuracy on CIFAR10.

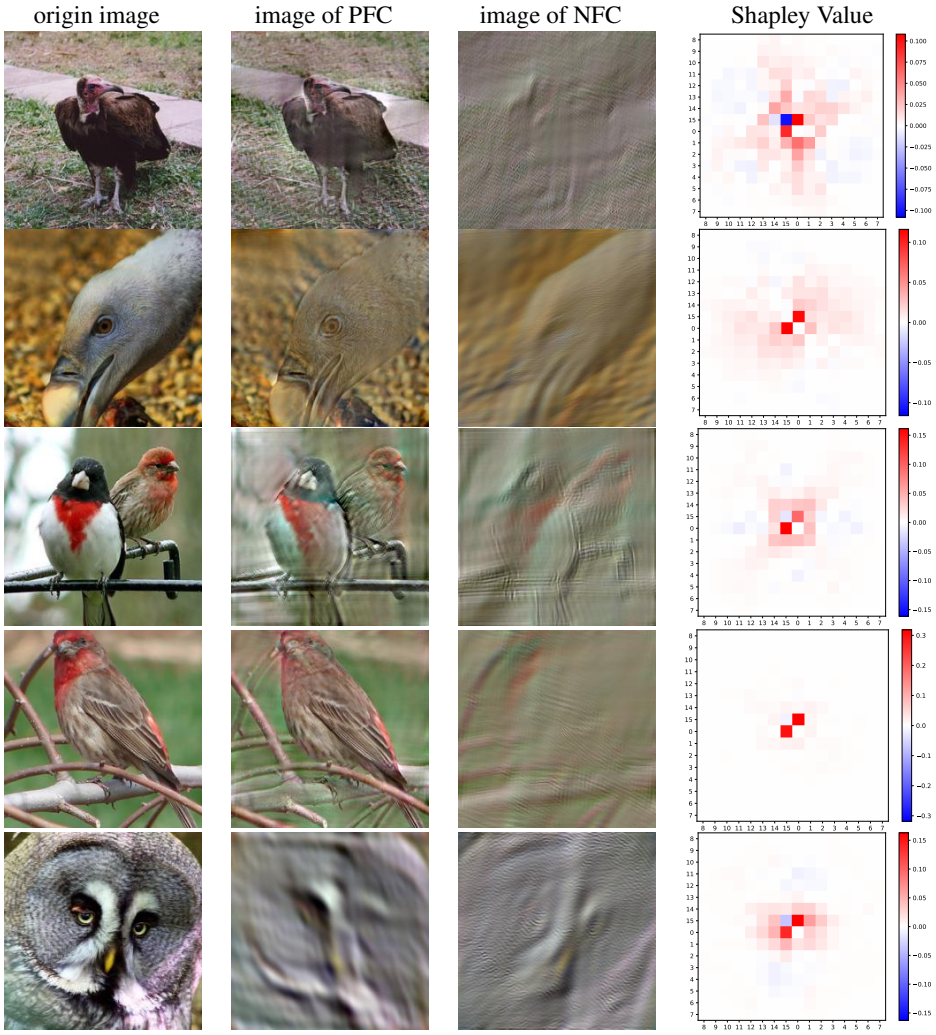


Figure 8: Shapley value of frequency components of data samples from ImageNet

sampling times	of patches	time consumed(s)
1000	16×16	79.58 ± 5.21
1000	32×32	357.37 ± 7.49
2000	16×16	151.50 ± 5.34
2000	32×32	630.67 ± 13.64

Table 2: Time consumed for sampling Shapley value for ResNet-18 on CIFAR10 with Intel(R) Core(TM) i7-7820X CPU @ 3.60GHz and GeForce RTX 2080 Ti. Each result is averaged over 10 test.

Methods		Natural Accuracy w/o attack			attacked by PGD-20			attacked by AutoAttack		
		Best	Final	Diff	Best	Final	Diff	Best	Final	Diff
TRADES [43]	baseline	**81.23%**	**81.71%**	-0.48	51.18%	51.08%	**0.10**	47.94%	47.74%	0.20
	sup HFC	78.14%	80.26%	-2.12	50.66%	50.42%	0.24	45.47%	45.14%	0.33
	CSA	80.84%	81.62%	**-0.78**	**52.57%**	**52.06%**	0.51	**48.94%**	**49.16%**	**-0.22**

Table 3: Testing accuracy of models trained on CIFAR10 with CSA and high frequency suppressing methods. 'baseline' denotes the original TRADES method and 'sup HFC' denotes combining the suppressing high frequency method with TRADES.

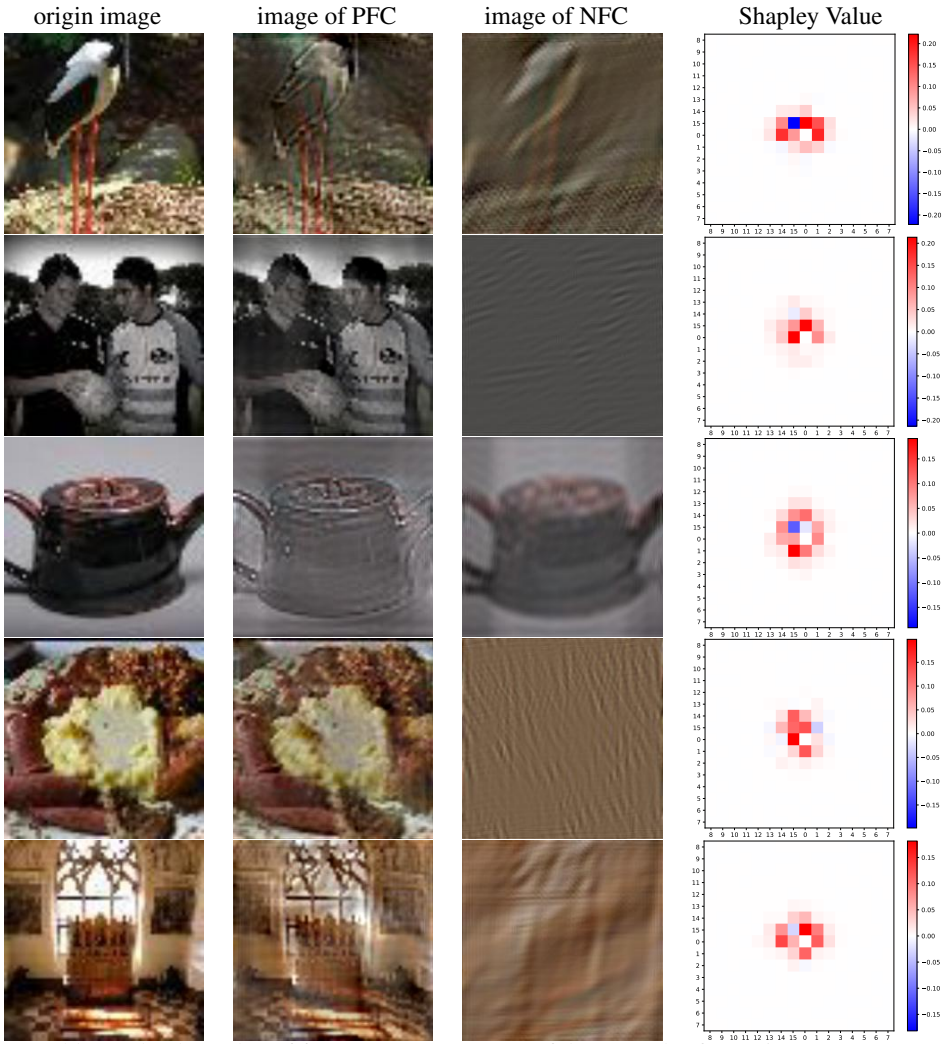


Figure 9: Shapley value of frequency components of data samples from TinyImageNet

B.1 Experiment Results of vgg16 on CIFAR10

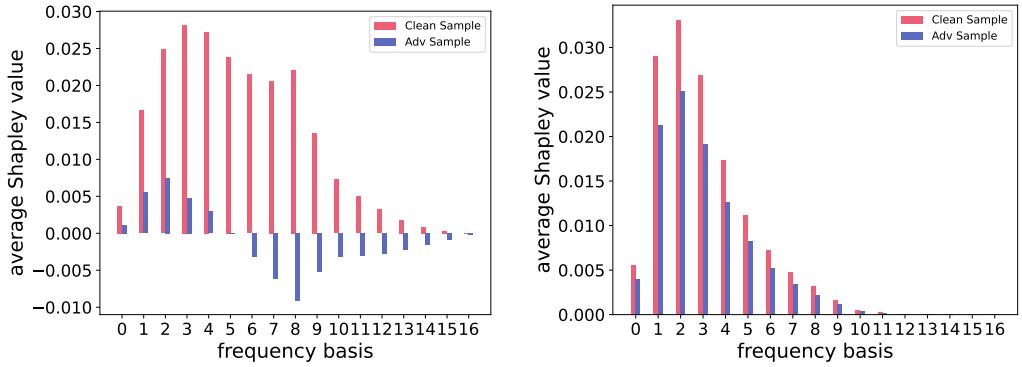
In this section we present the experiment results of vgg16 on CIFAR10, the results are similar to the results of ResNet-18. The dimension of fully connected layers of vgg16 is changed from 4096 to 512 in order to suit the scale of the image in CIFAR10. Fig. 10 shows the similar difference between adversarial trained vgg16 and standard trained vgg16. Fig. 11 demonstrate a similar negative relationship between the adversarial accuracy of AT model and the average absolute Shapley value of HFC for ST model with vgg16. Fig. 12 demonstrate the same attack bias for vgg16 as ResNet18.

B.2 Experiment Results on CIFAR100

In this section, we present the experiment results on CIFAR100.

Shapley value in frequency domain For frequency domain analysis, we conduct the same experiments on CIFAR100 as on CIFAR10. As shown in Fig. 13, we present the average Shapley value of frequency components of the first 10 images of each class over frequency basis. The results are similar to the results on CIFAR10 except that the absolute average Shapley value of adversarial examples on ST model are relatively smaller.

Class-wise Shapley value-guided augmentation We train ResNet-18 augmented with our proposed CSA where the training settings are the same as the settings on CIFAR10. For each class, we extract



(a) ST model (b) AT model

Figure 10: Average Shapley value of frequency componets in clean samples and adversarial samples of vgg16 over frequency basis on CIFAR10

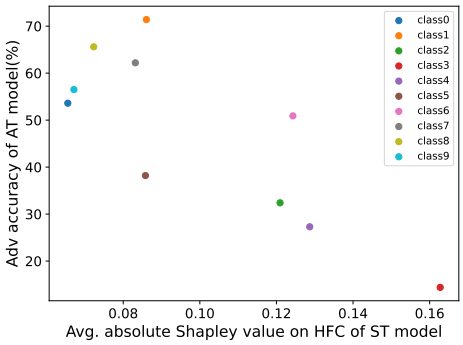


Figure 11: The negative relationship between the adversarial accuracy of AT model and the average absolute Shapley value of HFC for ST model(Experiment conducted on vgg16)

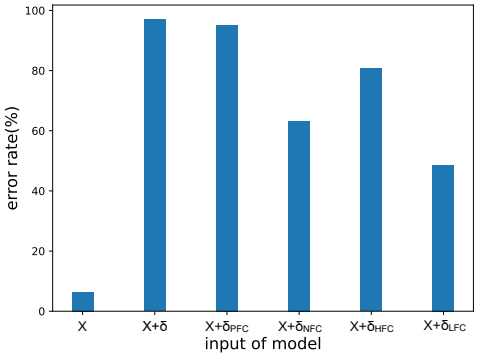
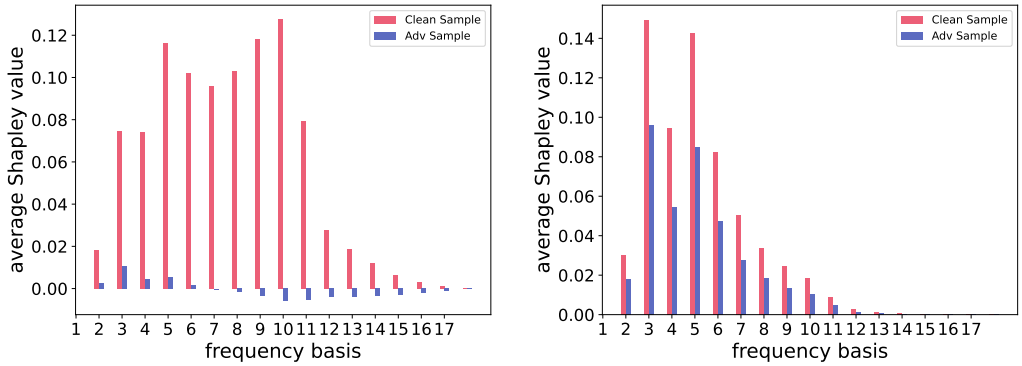


Figure 12: The error rate of vgg16 on CIFAR10 with different input. X represent origin data. δ is the adversarial noise generated against the model.



(a) ST model (b) AT model

Figure 13: Average Shapley value of frequency componets in clean samples and adversarial samples of ResNet-18 over frequency basis on CIFAR100

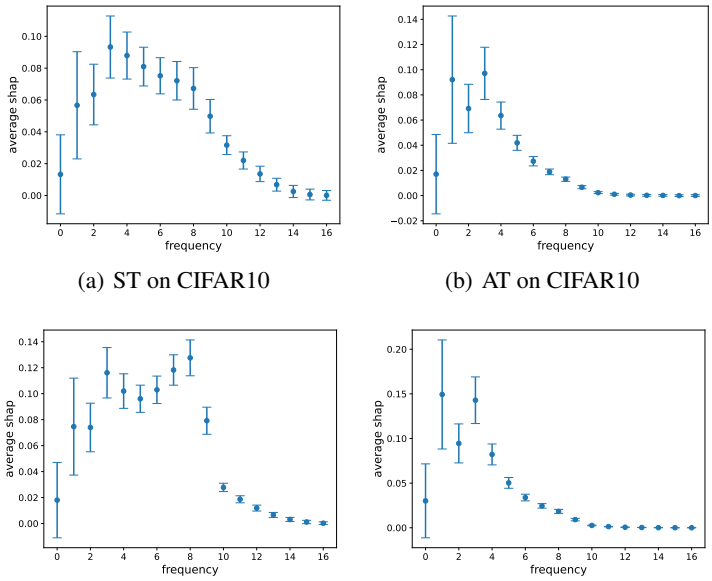
CSA from the first 30 images. The result is presented in Table 4. The data augmentation with CSA could effectively improve the robustness of the model.

Table 4: Testing accuracy of ResNet-18 trained on CIFAR100.

Methods		Natural Accuracy w/o attack			attacked by PGD-20			attacked by AutoAttack		
		Best	Final	Diff	Best	Final	Diff	Best	Final	Diff
PGD-AT [22]	baseline	**56.55%**	**57.13%**	-0.58	24.38%	21.70%	2.68	21.66%	20.10%	1.56
	CSA	56.31 %	56.72 %	**-0.41**	**26.14%**	**23.63%**	**2.51**	**23.02%**	**21.64%**	**1.38**
TRADES [43]	baseline	53.09%	53.09%	**0.00**	27.01%	27.01%	**0.00**	22.76%	22.76%	**0.00**
	CSA	**54.88%**	**54.46%**	0.42	**27.81%**	**27.77%**	0.04	**23.61%**	**23.73%**	-0.12

B.3 Stability analysis of Shapley Value

We show the variance of Average Shapley Value between different data samples to measure the statistical stability on Fig. 14. We plot the error bars of the average Shapley value of frequency components of clean samples on ST model and AT model on CIFAR10 and CIFAR100. The variance is relatively big on LFCs and small on HFCs. For HFCs, the average Shapley value and the variance on AT model is smaller than that on ST model and near zero which shows that AT model are less impacted by HFCs.



(a) ST on CIFAR10 (b) AT on CIFAR10

(c) ST on CIFAR100 (d) AT on CIFAR100

Figure 14: Average Shapley value of clean samples with error bars on CIFAR10 and CIFAR100. The length of the error bar is 2σ where σ is standard error.

C Details of Our Proposed Data Augmentation

C.1 Training Details of CSA

As introduced in Eq. 13, we extract the CSA of images in the train set of CIFAR10 and CIFAR100 for ST model. We train model in the same manner as the AT model trained in Appendix A. Models are trained for 100 epochs with SGD optimizer where learning rate is set to be 0.1 and decayed by 0.1 at 75-th epoch and 90-epoch. The momentum is set to be 0.9 and weight decay is set to be $2e-4$. For other data augmentations on train data, we employ random horizontal flip with the probability set to 0.5. The adversarial examples are generated with PGD-10 [22] with a L-inf bound where epsilon is set to be $8/255$ and stepsize is set to be $2/255$. For TRADES[43], the trade-off regularization parameter β is set to 6. The α in Eq. 14 linearly increase from 0 to 0.5 through the training process.

C.2 Ablation study for the size of our CSA

The number of CSA we used for data augmentation may affect the effectiveness of the data augmentation. We generate CSA for the first 150 images of each class of CIFAR10 that is 3% of whole train set. We train ResNet-18 as described in Appendix C.1 with different number of CSA and test the model with AA [12] and PGD-20 [22] at the last epoch. The result is shown in Fig. 15.

As the number of CSA for each class on CIFAR10 increasing, the test accuracy under AA and PGD-20 attack gradually increase. At the same time the data augmentation with CSA is already effective with very few CSA *i.e.* 30 CSA per class which means the CSA we generate could represent features that are not correctly learned by the model.

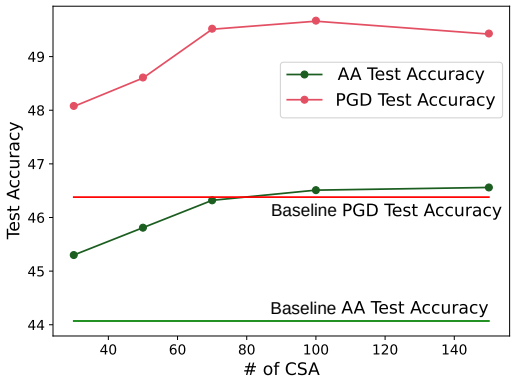


Figure 15: PGD and AA Test Accuracy of ResNet-18 trained with different number of CSA for each class on CIFAR10 at the last epoch

D Justification for the Shapley Value on CNNs in Frequency domain

D.1 Properties of the Shapley value

As proved in the seminal work [27], the Shapley value is the only solution that satisfies the following desirable properties:

- **Null player**: A player i is a null player if $\forall S \subseteq \mathcal{N}, V(S \cup \{i\}) = V(S)$. The Shapley value of every null player is zero.
- **Efficiency**: Shapley value satisfies that $\sum_{i \in \mathcal{N}} \phi_i^{\mathcal{N},V} = V(\mathcal{N})$
- **Symmetry**: Two players $i, j \in \mathcal{N}$ are symmetric if $\forall S \subseteq \mathcal{N} \backslash \{i, j\}, V(S \cup \{i\}) = V(S \cup \{j\})$. The Shapley value satisfies that for all symmetric players i, j, $\phi_i^{\mathcal{N},V} = \phi_j^{\mathcal{N},V}$
- **Linearity**: For two games $(\mathcal{N}, U), (\mathcal{N}, V)$, if a third game is defined as $(\mathcal{N}, W)$ where $\forall S \subseteq \mathcal{N}, W(S) = U(S) + V(S)$. It holds that $\phi_i^{\mathcal{N},W} = \phi_i^{\mathcal{N},U} + \phi_i^{\mathcal{N},V}$

D.2 The Decomposition of Convolution in Frequency Domain

As shown in Remark 1, the output of a convolution operation can be decomposed into sum of output of frequency components of X. For a convolution kernel $C \in \mathbb{R}^{d_3 \times d_4}$, the element at i-th row and

j-th column of the output of convolution function with the kernel f_C is

$$f_C(X)(i,j) = \sum_{m=0}^{d_3-1}\sum_{n=0}^{d_4-1} X(i+m, j+n)C(m,n) \tag{16}$$

$$= \sum_{m=0}^{d_3-1}\sum_{n=0}^{d_4-1}\left[\frac{1}{d_1 d_2}\sum_{u=0}^{d_1-1}\sum_{v=0}^{d_2-1}\mathcal{F}(X)(u,v)e^{i2\pi(\frac{(i+m)u}{d_1}+\frac{(j+n)v}{d_2})}\right]C(m,n) \tag{17}$$

$$= \sum_{u=0}^{d_1-1}\sum_{v=0}^{d_2-1}\mathcal{F}(X)(u,v)\sum_{m=0}^{d_3-1}\sum_{n=0}^{d_4-1}\frac{1}{d_1 d_2}e^{i2\pi(\frac{(i+m)u}{d_1}+\frac{(j+n)v}{d_2})}C(m,n) \tag{18}$$

We represent the $d_1 \times d_2$ matrix where all the elements equal to zero except the element on u-th row and v-th column is one with $I^{u,v}_{d_1\times d_2}$. We have

$$\mathcal{F}^{-1}[I^{u,v}_{d_1\times d_2}](i+m, j+n) = \frac{1}{d_1 d_2}e^{i2\pi(\frac{(i+m)u}{d_1}+\frac{(j+n)v}{d_2})} \tag{19}$$

Then we get

$$f_C(X)(i,j) = \sum_{u=0}^{d_1-1}\sum_{v=0}^{d_2-1}\mathcal{F}(X)(u,v)\sum_{m=0}^{d_3-1}\sum_{n=0}^{d_4-1}\mathcal{F}^{-1}[I^{u,v}_{d_1\times d_2}](i+m, j+n)C(m,n) \tag{20}$$

$$= \sum_{u=0}^{d_1-1}\sum_{v=0}^{d_2-1}\mathcal{F}(X)(u,v)f_C\left[\mathcal{F}^{-1}[I^{u,v}_{d_1\times d_2}]\right](i,j) \tag{21}$$

Therefore, we get

$$f_C(X) = \sum_{u=0}^{d_1-1}\sum_{v=0}^{d_2-1}\mathcal{F}(X)(u,v)f_C\left[\mathcal{F}^{-1}[I^{u,v}_{d_1\times d_2}]\right] \tag{22}$$