Improving Native CNN Robustness with Filter Frequency Regularization

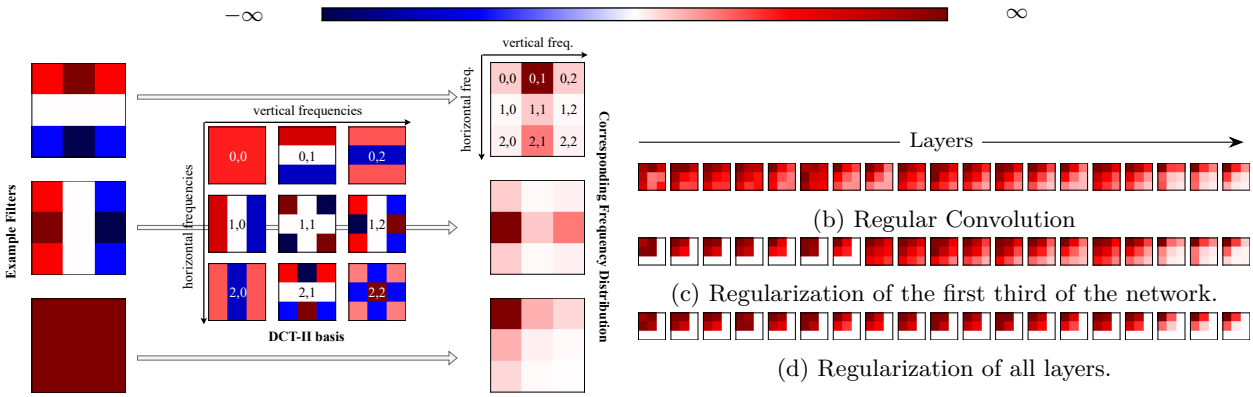
Anonymous authors
Paper under double-blind review

Abstract

Neural networks tend to overfit the training distribution and perform poorly on out-of-distribution data. A conceptually simple solution lies in adversarial training, which introduces worst-case perturbations into the training data and thus improves model generalization to some extent. However, it is only one ingredient towards generally more robust models and requires knowledge about the potential attacks or inference time data corruptions during model training. This paper focuses on the native robustness of models that can learn robust behavior directly from conventional training data without out-of-distribution examples. To this end, we investigate the frequencies present in learned convolution filters. Clean-trained models often prioritize high-frequency information, whereas adversarial training enforces models to shift the focus to low-frequency details during training. By mimicking this behavior through frequency regularization in learned convolution weights, we achieve improved native robustness to adversarial attacks, common corruptions, and other out-of-distribution tests. Additionally, this method leads to more favorable shifts in decision-making towards low-frequency information, such as shapes, which inherently aligns more closely with human vision.

1 Introduction

Modern convolutional neural networks (CNNs) (He et al., 2016; Liu et al., 2022; Tan & Le, 2019) show a steady increase in performance in terms of test accuracy on a wide range of learning tasks. Yet, most models suffer from a low generalization ability, even when faced with small domain shifts. To improve the low generalization ability, previous work focused on aspects such as aliasing in the downsampling operation (Grabinski et al., 2022b;a), the padding operations (Gavrikov & Keuper, 2023), the training schedule (Lopes et al., 2019; Saikia et al., 2021), or analyzing the image feature spectrum (Geirhos et al., 2019; Wang et al., 2020). In addition, introducing perturbed images into the training data, known as adversarial training (AT) (Madry et al., 2018), can alleviate low generalization to some extent. However, AT is not the cure-all to improve network robustness. Tramèr & Boneh (2019); Rice et al. (2020); Yu et al. (2022) show that AT tends to overfit on training attacks. Intuitively, the adversarial attack used during training becomes an in-domain sample of the model, while its robustness to new out-of-domain samples (*e.g.* a different adversarial attack) is hard to anticipate. Saikia et al. (2021) show that AT can even increase the mean corruption error on ImageNet-C (Hendrycks & Dietterich, 2019). Therefore, we argue that AT can only be one ingredient towards building more robust models, while the main focus should rather be to encourage a behavior that we call *native robustness*. We expect from natively robust models that they can learn robust behavior directly from the conventional training data. Thereby, robust behavior includes, on the one hand, a certain degree of adversarial robustness without being confronted with adversarial attacks during training, *i.e.* the model should not easily be fooled using attacks with very small perturbation budgets. Similarly, they should be robust against other perturbations such as common corruptions (Hendrycks & Dietterich, 2019) as long as corruption severities are low. On the other hand, robust behavior implies a better alignment with human perception, *i.e.* models should decide for a specific class more by the shape of an object than by its texture (Geirhos et al., 2019). Note that the expected degree of *specific* robustness can not be compared to the one obtained by techniques that specifically optimize for them, such as adversarial training. For instance,



(a) Computation of frequency distributions

(b) Regular Convolution

(c) Regularization of the first third of the network.

(d) Regularization of all layers.

Figure 1: Our proposed regularization decreases the reliance on high-frequency information of a ResNet-20 trained on CIFAR-10 (b-d) as visible in mean DCT-II coefficients magnitudes (= frequency distribution; computation shown in (a)).

adversarial samples remain out-of-domain samples for such natively robust models. Yet, additionally training these natively robust models with AT should be complementary and have a further beneficial effect.

In this paper, we propose a new perspective on improving native robustness by investigating the frequencies in the learned network filters directly.

Specifically, we propose to project CNN convolution filter weights into the frequency domain by applying a discrete cosine transformation (DCT-II). Although the resulting formulation is in principle equivalent to the commonly adopted CNN formulation, it provides direct access to the learned filter frequencies. Thereby, we aim to investigate the following research questions: (i) Which filter frequencies are predominantly learned in the layers of CNNs? (ii) Can we regularize the frequencies during the training process such as to increase the native robustness of the learned model?

We investigate these questions in the context of image classification - yet our approach bears the potential to be expanded to other tasks such as object detection and segmentation. First, we analyze the learned filter frequencies of modern CNNs and observe that they tend to have a low-frequency bias in deep layers, while filters of earlier layers of the network are either uniformly distributed in frequency space or even biased towards higher frequencies. In the latter cases, the convolution thus relies on high-frequency information. Contrary, adversarial training appears to shift the focus to low filter frequencies in early layers. To leverage this behavior, we introduce a regularization scheme, which increases the bias to low-frequencies in these early layers (see Fig. 1 for a visualization). We evaluate the proposed spectral decomposition and regularization on different CNNs under distribution shifts in test data. Results on CIFAR-10, CIFAR-100 (Krizhevsky, 2009), SVHN (Netzer et al., 2011), MNIST (LeCun et al., 2010), Tiny-ImageNet (Le & Yang, 2015) and ImageNet (Deng et al., 2009) show increased native robustness[1]. In summary, we make the following contributions:

- We observe that adversarial training results in a shift towards a low-frequency bias in the filter weights of early layers that is learned in the early phases of training (Sec. 3).
- Based on this observation, we propose a high-frequency penalization term in the weight space of convolution layers (Sec. 4) to mitigate the reliance on high-frequency information.
- Networks trained with this regularization become gradually, yet consistently, more robust against a wide array of out-of-distribution generalization tasks without reliance on AT or additional data - *i.e.* networks increase their native robustness (Sec. 5). Additional AT is complementary and further improves the measurable adversarial robustness to a variety of attacks.

[1]We will provide the code upon acceptance.

2 Related work

Robustness. While modern neural networks yield accuracies close to or even beyond human performance, they seem to struggle with generalization to out-of-distribution data. In the context of adversarial attacks, it has been shown that minor, for the human eye barely perceivable perturbations can cause models to make wrong predictions with high confidence. Formally, let f be a model parameterized by θ, $\boldsymbol{x}$ an input sample with the corresponding class label $\boldsymbol{y}$, and $\mathcal{L}$ the loss function. Then adversarial attacks will attempt to maximize the loss $\mathcal{L}$ by finding an additive perturbation to an input sample $\boldsymbol{x}'$. To constrain their intensity, perturbations are sought in an $\mathcal{B}_\epsilon(\boldsymbol{x})$ ball centered at $\boldsymbol{x}$ with a radius of ϵ.

$$\begin{aligned} \max_{\boldsymbol{x}' \in \mathcal{B}_\epsilon(\boldsymbol{x})} \mathcal{L}\left(f\left(\boldsymbol{x}'; \theta\right), \boldsymbol{y}\right), \\ \mathcal{B}_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}' : \|\boldsymbol{x} - \boldsymbol{x}'\|_p \leq \epsilon\}. \end{aligned} \tag{1}$$

with $\|\cdot\|_p$ depicting the L^p-norm. The most successful adversarial attacks are white-box attacks, where the attacker has full access to the attacked model. Often, these methods rely on gradient information, such as *projected gradient descent* (PGD) (Kurakin et al., 2017) where the attacker follows the gradient that maximizes the loss and then projects the perturbations back to $\mathcal{B}_\epsilon(\boldsymbol{x})$. Since PGD is computationally expensive, a faster, yet less successful attack that approximates the perturbations by the gradient sign and only performs one step has been proposed: *fast gradient sign method* (FGSM) (Goodfellow et al., 2015).

Regarding defenses in general, the most successful approach to tackle out-of-distribution shifts is *adversarial training* (Madry et al., 2018) where worst-case perturbations are reintroduced to the training data. Often, these methods are accompanied by additional external data (Carmon et al., 2022). Evaluating the defenses on a single attack can be misleading, due to the possibility of attack overfitting (Rice et al., 2020). Towards more reliable benchmarks, *AutoAttack* (AA) (Croce & Hein, 2020a) proposes an ensemble of various white- and black-box attacks such as *APGD* (Croce & Hein, 2020a), *FAB* (Croce & Hein, 2020b), and *Square* (Andriushchenko et al., 2020) and establishes the public *RobustBench* leaderboard. Benchmarks are constrained to $p = 2, \epsilon = 0.5$ on CIFAR-10, $p = \infty, \epsilon = 8/255$ on CIFAR-10/100, and $p = \infty, \epsilon = 4/255$ on ImageNet, respectively. However, these large thresholds are disputed as they generate easily detectable perturbations (Lorenz et al., 2022).

Unfortunately, the possibility of adversarial attacks is only a symptom of larger generalization issues. For example, neural networks fail to generalize under various corruptions such as weather conditions, changes in lighting, noise, and blurring (Dodge & Karam, 2017; Hendrycks & Dietterich, 2019). For fast and comparable benchmarks, common corruption datasets CIFAR-10-C, CIFAR-100-C, and ImageNet-C have been proposed (Hendrycks & Dietterich, 2019), which include 15 (+4 extra) types of corruptions at increasing severity level (from 1 to 5).

Additionally, Geirhos et al. (2019) observed that CNNs are biased towards detecting textures of an image instead of the shape (cue-conflict), which is in contrast to human vision behavior that focuses on shape information, *i.e.* shape bias. In order to overcome this texture bias, they introduced Stylized-ImageNet as new training data with the goal to increase the shape bias of CNNs. For fast evaluation of out-of-distribution (OOD) generalization Geirhos et al. (2021) proposed a benchmark including 17 OOD datasets, from which 12 contain image perturbations and the other 5 are single manipulations of ImageNet (Deng et al., 2009): cue-conflicted texture vs. shape data, sketches (Wang et al., 2019), stylized images, edges, and silhouettes. They evaluated and compared more than 50 different networks to human performance in order to narrow down the gap between human and machine vision.

Frequencies and robustness. Recent work demonstrated the importance of learned frequencies for network robustness. Wang et al. (2020) demonstrated that CNNs initially rely on low-frequency information for prediction, but shift towards high-frequency information as training progresses. On the other hand, AT models predominantly classify based on low-frequency information. As texture information typically resides in higher frequency bands, this is a suitable explanation for the observations by Geirhos et al. (2019). As such, there is also a correlation between AT and a reduced texture bias (Geirhos et al., 2021; Gavrikov et al., 2023). Duan et al. (2021) exploit these findings by proposing an adversarial attack that drops DCT

coefficients corresponding to high frequencies from inputs to fool neural networks. Yet despite the common assumption, adversarial attacks are not always targeting high-frequencies and the behavior depends on the dataset (Maiya et al., 2021; Abello et al., 2021; Bernhard et al., 2021; Ortiz-Jiménez et al., 2020).

Multiple works explore the desensitization of neural networks to HF from various angles in order to avoid AT: Lopes et al. (2019) randomly add noise to image patches, Saikia et al. (2021) regularize the feature maps produced by convolution layers in a dedicated two-stream architecture, and Grabinski et al. (2022a) link robustness to aliasing and introduce a downsampling approach within the frequency domain that removes high-frequency information due to aliasing. In this paper, we follow this line of work and propose an HF-regularization directly in convolution filters to improve OOD generalization.

Basis decomposition. The decomposition of convolution filters is typically studied in the context of compression, see Yaroslavsky (2014) for an overview. The majority of decomposition approaches convert the convolution layer weights to the frequency domain *e.g.* by utilizing the DCT-II-basis (Chen, 2004; Chen et al., 2016; Lo & Hang, 2019; Cheinski & Wawrzynski, 2020; Chen et al., 2022; Ulicny et al., 2022) with the goal to prune and compress the number of frequency components. But works also exist that transform the input images directly for better performance and generalization (Xu et al., 2020; Hossain et al., 2019). In detail, the *discrete cosine transform* (DCT) (Ahmed et al., 1974) maps an input signal into a frequency domain represented by cosine basis functions. In particular, the common DCT-II variant is used in JPEG compression, where it successfully compresses natural images (Wallace, 1992). These works mainly explore the fact, that data of multiple domains is not uniformly distributed in the frequency domain and is typically biased towards low frequencies (Singh & Theunissen, 2004; Ruderman, 1994). Gavrikov & Keuper (2022a;b) showed that the basis of convolution filter kernels obtained via SVD is often highly similar and independent of the architecture, learned task, or dataset. Additionally, the obtained bases have a striking similarity to the DCT-II basis.

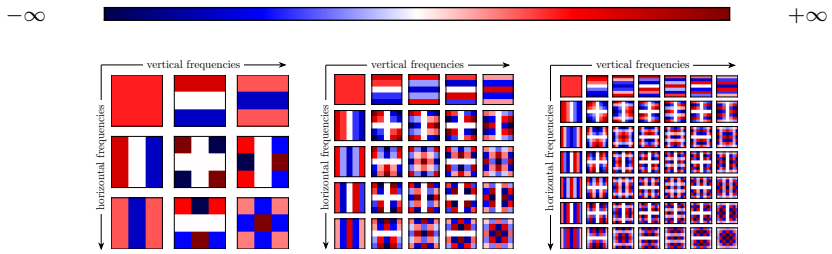
Our realization of the DCT-II basis is similar to Ulicny et al. (2022) and other previous work, however, instead of compression, we explore an orthogonal direction and study the role of individual frequencies in training and apply regularization in the frequency space to improve generalization. DCT merely serves as a tool in our study.

3 Frequency analysis

In this initial analysis, we transform learned convolution filters to the frequency domain. We implement this by changing the basis of convolution weights to DCT-II, revealing the coefficients and therefore frequency information. Formally we define this as follows. Let $\mathbf{V}$ denote the $k \times k$-DCT-II basis. Then every basis vector $\mathbf{V}_{i,j}$ with horizontal frequency j and vertical frequency i is defined as:

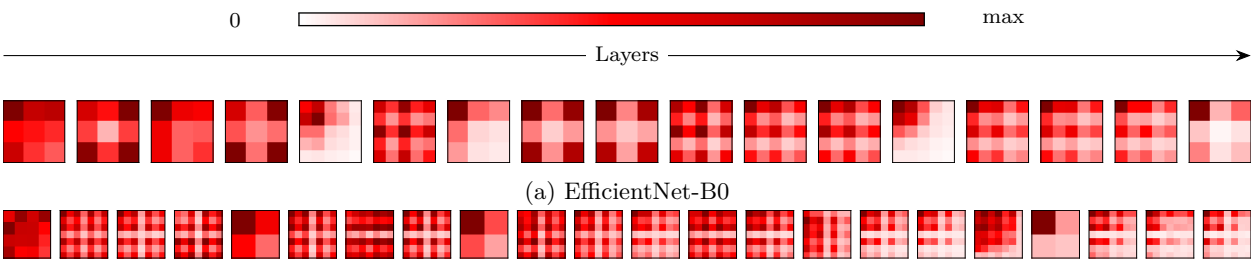
$$\mathbf{V}_{i,j,m,n} = \cos\left[\frac{\pi i}{k}\left(m + \frac{1}{2}\right)\right] \cos\left[\frac{\pi j}{k}\left(n + \frac{1}{2}\right)\right]. \tag{2}$$

Every basis vector is additionally normalized to its L^1 length: $\mathbf{V}_{i,j} = \mathbf{V}_{i,j} / \|\mathbf{V}_{i,j}\|_1$. Exemplary, we show the DCT-II basis vectors for different kernel sizes k in Fig. 2. In principle, DCT-II could be replaced by any



(a) 3x3 (b) 5x5 (c) 7x7

Figure 2: The full DCT-II basis for different resolutions.



(a) EfficientNet-B0

(b) ConvNeXt-Tiny

Figure 3: Frequency distribution of each layer for trained (a) EfficientNet-B0 (containing 3×3 and 5×5 kernels) and (b) ConvNeXt-Tiny (containing 2×2 [downsampling layers], 5×5 [stem] and 7×7 kernels).

other frequency base such as a discrete Fourier or sine transform. Following the basis change, we visualize the average magnitude of coefficients in every convolution layer by heat maps (as shown in Fig. 1). Having the frequency information at hand, we can directly analyze its distribution in common CNNs.

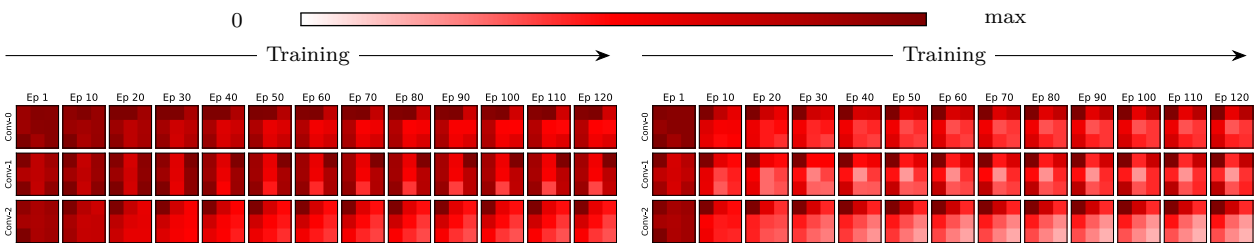
3.1 Analyzing learned convolution weights

We start by analyzing two modern networks trained on ImageNet without any robustness optimization techniques: EfficientNet-B0 (Tan & Le, 2019) and ConvNeXt-Tiny (Liu et al., 2022) (Fig. 3). Our visualizations show that these CNNs do not always learn a uniform frequency spectrum utilization throughout the network. Earlier layers show a more uniform distribution of magnitude or are biased towards higher frequencies. However, deeper convolution layers instead reveal a salient bias towards low frequencies. Some layers even appear to discard a majority of high-frequency information.

In addition, we are interested how adversarial training affects the frequency utilization in convolution filters. As shown from various angles (Wang et al., 2020; Geirhos et al., 2019; Saikia et al., 2021) robust models shift their bias to low-frequencies, as this reduces the possibility of overfitting on high-frequencies and therefore provides better generalization abilities. Thus, we expect that these results transfer to the frequency utilization in weight space to some extent. Indeed, Wang et al. (2020) stated that the very first convolution layer of AT CNNs learns smoother filters which equals to filters that are less reliant on high-frequency information than the equivalents in normally trained models. However, their frequency analysis was limited to the first initial layer, while we aim to provide a holistic analysis over the entire network. This is also backed by our previous observations showing that frequency utilization varies by depth. Further, their results do not appear to be representative of modern models that are trained under L^∞-norm. Such models predominantly learn thresholding filters (Madry et al., 2018) independent of architecture and dataset (Gavrikov & Keuper, 2022b) that do not resemble "common" first layers as shown by Yosinski et al. (2014). As such, they are hardly smooth.

Exemplarily, we proceed by comparing an adversarially trained EfficientNet-B0 with its regularly-trained counterpart (more comparisons are included in the appendix). We observe that adversarial training leads to a characteristically different distribution of learned frequencies during training (Fig. 4). Especially in the first layers, the network learns predominantly from low frequencies, which enables the network to preserve the global image content, rather than overfitting on high-frequency details such as texture. Interestingly, the adversarially-trained model learns this behavior in the early training stages, and faster than under normal training conditions (Rahaman et al., 2019). Deeper layers on the other hand show no salient differences.

Based on these findings, we propose a transformation approach of convolution weights into the frequency domain to interact with frequency information. Secondly, based on the latter finding we propose a high-frequency regularization, to further enforce the low-frequency bias in the first network layers and thus increase the native robustness.



(a) Regular training.

(b) Adversarial training.

Figure 4: Evolution of the frequency distributions in the first three convolution layers of an EfficientNet-B0 in comparison between (a) regular and (b) adversarial training with CIFAR-10. Evolution plots for all layers and for other architectures can be found in the appendix.

4 Modifications to the convolution layers

Let us first formalize the computation flow in a conventional 2D convolution layer $f_{conv2d}(\boldsymbol{x}; \mathbf{W})$, f_{conv2d} transforming an input signal $\boldsymbol{x}$ with d_{in} input-channels into a signal with d_{out} output-channels using a convolution kernel with a size of $k_0 \times k_1$. Further, let $\mathbf{W} \in \mathbb{R}^{d_{out} \times d_{in} \times k_0 \times k_1}$ denote the learned weights (*i.e.* the set of all kernels $\mathbf{W}_{i,j}$ in the respective layer, without bias). Without loss of generality, we assume $k_0 = k_1 = k$ in this paper. The output of $f_{conv2d}(\boldsymbol{x}; \mathbf{W})$ is then defined as:

$$\boldsymbol{y}_s = \sum_{d=1}^{d_{\text{in}}} \mathbf{W}_{s,d} * \boldsymbol{x}_d, \text{ for } s \in \{1, \ldots, d_{\text{out}}\}. \tag{3}$$

In the following, we propose a simple representation in the frequency space by replacing the convolution weight $\mathbf{W}$ with a combination of learned coefficients on the DCT-II basis. In this work, we limit ourselves to kernels with $k \geq 3$. We realize this by two common implementations seen in related literature (*e.g.* Ulicny et al. (2022)). Schematic visualizations of both approaches can be found in the appendix.

Weight decomposition (WD). Our first approach decomposes the weight in a convolution layer into learnable coefficients $\mathbf{C} \in \mathbb{R}^{d_{out} \times d_{in} \times k \times k}$ and the basis $\mathbf{V}$ defined in Eq. 2:

$$\mathbf{W} = \mathbf{C} \cdot \mathbf{V}. \tag{4}$$

Then, the convolution can be rewritten as:

$$\boldsymbol{y}_s = \sum_d (\mathbf{C}_{s,d} \cdot \mathbf{V}) * \boldsymbol{x}_d = \sum_{d,m,n} (\mathbf{C}_{s,d,m,n} \cdot \mathbf{V}_{m,n}) * \boldsymbol{x}_d. \tag{5}$$

This increases the parameters to be kept in memory by a factor of 2 and adds one additional tensor multiplication per layer. However, these additional parameters are constant and do not need to be learned.

Signal decomposition (SD). Alternatively, our second approach does not replace the convolution weight $\mathbf{W}$ directly but performs a depthwise convolution of all combinations of inputs and the fixed basis vectors which is then aggregated by a learnable pointwise (1×1) convolution.

$$\boldsymbol{y}_s = \sum_{d,m,n} \mathbf{C}_{s,d,m,n} \cdot (\mathbf{V}_{m,n} * \boldsymbol{x}_d). \tag{6}$$

This increases the parameter number by a factor of $d_{in}k^2$ to be kept in memory. Again, the number of learnable parameters is not increased. Also, note that the associativity property of convolution reveals the equivalence of both formulations in the forward pass:

$$\boldsymbol{y}_s = \sum_{d,m,n} \mathbf{C}_{s,d,m,n} \cdot (\mathbf{V}_{m,n} * \boldsymbol{x}_d) = \sum_{d,m,n} (\mathbf{C}_{s,d,m,n} \cdot \mathbf{V}_{m,n}) * \boldsymbol{x}_d. \tag{7}$$

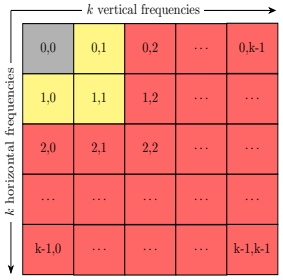


Figure 5: Regularization area of the coefficients in an individual filter kernel. Colors match Eq. 8.

$$\mathcal{R}(\mathbf{C}) = \underbrace{\left\|\mathbf{C}_{:,:,\lfloor k/2 \rfloor:,\lfloor k/2 \rfloor:}\right\|_2}_{\bullet} \cdot \frac{k}{2} + \rho_{diff} \cdot \max(\underbrace{\left\|\mathbf{C}_{:,:,:2,:2}\right\|_2 - \overbrace{\left\|\mathbf{C}_{:,:,0,0}\right\|_2}^{\bullet}}_{\bullet}, 0). \tag{8}$$

However, due to different learning dynamics, the modifications may converge to different solutions. In both approaches, the initial coefficient weights are sampled from a uniform distribution with an adjusted scale as per He et al. (2015). For the *weight decomposition* approach, we use $d_{in}k^2$ as fan information. The basis vectors are initialized as defined in Sec. 3 without any further adjustments.

4.1 Frequency coefficient regularization

As we have seen in Sec. 3.1 neural networks are biased towards low-frequency information, while early layers also introduce more magnitude on high frequencies. However, adversarial training increases the low-frequency bias already in the early training stages resulting in an overall low-frequency dominance after convergence in the first layers. To make use of this finding and increase the robustness of CNNs directly without adversarial training, we propose to regularize the DCT-II coefficients and explore the frequency shift and performance.

The proposed regularization (Eq. 8 and Fig. 5) regularizes the highest frequencies and additionally forces the first coefficient to have a higher magnitude than the subsequent frequency. The occurrence of the latter constraint is determined by the binary hyperparameter ρ_{diff}, with $\rho_{diff} = 1$ throughout the paper, if not stated otherwise. The multiplicative term $\frac{k}{2}$ increases penalization of higher frequencies. Let $coefs(\theta, h)$ denote a function that returns the set of convolution coefficient weights of the learnable parameters θ in the first $1/h$ section of the network depth. In order to enforce the dominance of low frequencies in early layers, we set $h = 3$ as our default value. We train the network with the following modifications to the objective:

$$\min_{\theta} \mathcal{L}\left(f\left(\boldsymbol{x};\theta\right), \boldsymbol{y}\right) + \lambda \sum_{\mathbf{C} \in coefs(\theta,h)} \mathcal{R}(\mathbf{C}). \tag{9}$$

Where $\boldsymbol{x}, \boldsymbol{y} \sim \mathcal{D}$ denotes the training dataset and $\mathcal{L}$ is the original objective. An exemplary visualization of the learned coefficients under regularization is given in Fig. 1 for $h \in \{1, 3\}$.

5 Experiments

In the following, we compare different architectures, with regular convolutions, and both decomposition variants (WD/SD) at varying frequency regularization (+ Reg.) (Eq. 8). For each combination, we report results on clean accuracy, as well as robustness to various aspects.

Models and datasets. We evaluate low-resolution datasets such as CIFAR-10/100 (Krizhevsky, 2009), MNIST (LeCun et al., 2010), SVHN (Netzer et al., 2011), and Tiny-ImageNet (Le & Yang, 2015) on ResNet-20 (as introduced for CIFAR in He et al. (2016)), ResNet-9 - a regular and larger ResNet with optimization for CIFAR and a reduced number of layers (see appendix for further details), and an EfficientNet-B0 (Tan & Le, 2019) where we remove striding from the stem convolution. For ImageNet (Deng et al., 2009), we evaluate EfficientNet-B0 (Tan & Le, 2019) and ConvNeXt-Tiny (Liu et al., 2022). We test $h \in \{1, 3\}$ and $\lambda \in \{0.01, 0.05, 0.1\}$ and report results for the best performance over the mean of 5 runs except for ImageNet (1 run). Details regarding the training can be found in the appendix.

Table 1: **Performance evaluation** of various networks (ResNet-20/9 and EfficientNet-B0) on multiple datasets (CIFAR-10/100, SVHN, TinyImageNet, MNIST) before and after our applied regularization. We report clean accuracy on the respective test sets as well as adversarial robustness against FGSM, PGD-40, and AutoAttack for $L^\infty, \epsilon = 1/255$ ($\epsilon = 16/255$ for MNIST). We report averages over 5 runs.

		Variant	Clean (↑) Val Acc.	Adversarial Acc. (↑) FGSM	PGD-40	AA
CIFAR-10	ResNet-20	CNN	91.29	50.49	30.92	10.78
		WD	91.04	48.40	30.37	10.72
		SD	**91.36**	50.83	32.98	11.97
		WD + Reg.	89.86	50.85	41.81	26.79
		SD + Reg.	90.54	**53.12**	**44.42**	**29.14**
	ResNet-9	CNN	**94.29**	59.58	53.04	37.49
		WD	93.73	55.51	49.84	35.23
		SD	93.97	55.73	50.29	36.0
		WD + Reg.	93.18	59.25	56.08	43.62
		SD + Reg.	93.09	**59.87**	**56.89**	**44.80**
	Eff.Net-B0	CNN	90.38	53.55	54.05	45.51
		WD	**90.51**	49.87	49.97	40.76
		SD	90.44	51.04	51.77	43.39
		WD + Reg.	88.97	**57.91**	59.60	53.30
		SD + Reg.	89.18	57.83	**59.68**	**53.50**

		Variant	Clean (↑) Val Acc.	Adversarial Acc. (↑) FGSM	PGD-40	AA
CIFAR-100	ResNet-20	CNN	**60.41**	14.36	5.45	1.17
		WD	58.90	12.84	4.68	1.01
		SD	60.34	13.87	5.18	1.13
		WD + Reg.	56.65	16.85	**12.73**	**5.59**
		SD + Reg.	58.19	**17.20**	12.24	5.11
SVHN	ResNet-20	CNN	96.31	83.84	79.94	69.81
		WD	**96.35**	83.52	80.01	71.25
		SD	96.34	84.07	80.64	71.74
		WD + Reg.	96.28	84.11	81.21	**73.27**
		SD + Reg.	96.34	**84.17**	**81.23**	73.03
TinyImNet	ResNet-9	CNN	**53.20**	17.79	16.76	9.57
		WD	52.08	17.11	16.19	9.40
		SD	52.12	16.85	15.88	9.15
		WD + Reg.	51.25	18.10	17.34	10.23
		SD + Reg.	51.22	**18.26**	**17.40**	**10.39**
MNIST	ResNet-20	CNN	99.68	89.74	45.37	8.92
		WD	99.69	90.45	47.06	10.22
		SD	99.65	**91.23**	54.08	16.80
		WD + Reg.	99.69	90.70	**55.84**	**25.92**
		SD + Reg.	99.69	88.98	50.02	21.71

Table 2: **Comparison against other robustness techniques** (Grabinski et al., 2022a; Lopes et al., 2019) of ResNet-20, ResNet-9, and EfficientNet-B0 on CIFAR-10. We report the mean clean validation accuracy and robust accuracy against adversarial attacks: FGSM, PGD-40, and AutoAttack for $L^\infty, \epsilon = 1/255$, and the mean corruption accuracy on CIFAR-10-C. We report averages over 5 runs.

	Variant	Clean (↑) Val Acc.	Adversarial Acc. (↑) FGSM	PGD-40	AA	CC (↑) Acc.
ResNet-20	CNN	91.29	50.49	30.92	10.78	67.96
	FLC (Grabinski et al., 2022a)	**91.52**	52.49	30.25	8.48	68.75
	PaGA (Lopes et al., 2019)	91.29	50.36	31.50	11.38	67.73
	WD	91.04	48.40	30.37	10.72	66.92
	SD	91.36	50.83	32.98	11.97	67.48
	WD + Reg.	89.86	50.85	41.81	26.79	74.04
	SD + Reg.	90.54	**53.12**	**44.42**	**29.14**	**74.14**
ResNet-9	CNN	94.29	59.58	53.04	37.49	73.38
	FLC (Grabinski et al., 2022a)	94.24	59.64	53.47	38.65	73.81
	PaGA (Lopes et al., 2019)	**94.33**	59.12	52.62	37.50	73.72
	WD	93.73	55.51	49.84	35.23	72.87
	SD	93.97	55.73	50.29	36.00	73.48
	WD + Reg.	93.18	59.25	56.08	43.62	76.41
	SD + Reg.	93.09	**59.87**	**56.89**	**44.80**	**77.72**
Eff.Net-B0	CNN	90.38	53.55	54.05	45.51	68.09
	FLC (Grabinski et al., 2022a)	89.68	51.92	53.09	45.37	69.72
	PaGA (Lopes et al., 2019)	**90.72**	54.18	54.97	46.64	69.31
	WD	90.51	49.87	49.97	40.76	67.10
	SD	90.44	51.04	51.77	43.39	66.65
	WD + Reg.	88.97	**57.91**	59.60	53.30	**72.14**
	SD + Reg.	89.18	57.83	**59.68**	**53.50**	71.87

Note that we have selected models with different kernel sizes - *e.g.* after the stem, ResNets use $k = 3$, EfficientNets-B0 mix $k = 3$ and $k = 5$, and ConvNeXts $k = 7$ (and $k = 2$ downsampling layers). The variance in kernel size allows us to demonstrate the transferability of our proposed regularization beyond the common $k = 3$ kernels.

Table 3: A **benchmark** of a ResNet-20 (CIFAR-10) with various convolution implementations evaluated on a NVIDIA A100 GPU.

Variant	Total Params (↓)	Learnable Params (↓)	Throughput (img/sec) (↑)
CNN	272k	272k	71.3k
WD (+ Reg.)	274k	272k	70.2k
SD (+ Reg.)	323k	272k	23.6k

Robustness evaluation. To understand the effect on robustness and generalization of our proposed decomposition and regularization approaches, we run the standard AutoAttack test suite (AA) (Croce & Hein, 2020a) and additional FGSM-, and PGD-attacks at $\epsilon = 1/255$ ($\epsilon = 16/255$ for MNIST) under the L^∞-norm. We use *Foolbox* (Rauber et al., 2017) to run both FGSM and PGD at the default setting (*e.g.* 40 steps for PGD). We do not include AA results for ImageNet, as these models barely withstand any attacks and measure robust accuracies of 0% even at this small ϵ without adversarial training. Further, we evaluate the robustness of common corruptions of CIFAR-10 and ImageNet models on the respective corrupted datasets (Hendrycks & Dietterich, 2019). In addition, we are interested in the behavior of the methods towards texture bias (Geirhos et al., 2019) and OOD generalization tests (Geirhos et al., 2021). Hence, we evaluate our ImageNet (Deng et al., 2009) models on 5 of these OOD datasets: texture-shape cue-conflict, ImageNet-Sketch, Stylized-ImageNet, and edge-/silhouette-transformations of ImageNet using the implementation of Geirhos et al. (2021).

5.1 Low-resolution datasets

CIFAR-10. As to be expected, switching from regular to either decomposition variant has an insignificant impact on the clean accuracy and a small effect on robust accuracy (at all adversarial attacks) (Tab. 1). However, applying the regularization clearly improves robustness towards all attacks, while slightly decreasing clean accuracy. We can also observe that SD slightly outperforms WD on almost all tested architectures. Hence, it may be tempting to only proceed with SD. However, the additionally created channels account for more parameters, a large memory overhead, and slower inference and training performance. *E.g.* on ResNet-20 we see a 3x slower forward pass and 18% more total parameters, while WD has a minimal overhead, both, in parameters and throughput (Tab. 3).
Regarding robustness, we see the largest gains on models that initially performed worst (+18.36% on ResNet-20 on CIFAR-10). Out of all our tested models, EfficientNet-B0 is the most robust, both, before and after regularization. Noticeably, even the worst hyperparameter combination for ResNet-20 (WD, $\lambda = 0.01, h = 1, \rho_{diff} = 0$) still achieves a 14.22% higher AA accuracy than the baseline. A complete overview of tested hyperparameters is in the appendix.

Common corruptions of CIFAR-10. For common corruptions (CC) (the last column in Tab. 2 corresponds to the CIFAR-10 results in Tab. 1), we analyze the mean accuracy over all corruptions and severities, as well as individual results for corruptions at the highest severity level. A complete overview is given in the appendix. Similar to the results on adversarial robustness we observe that on average both regularized variants outperform the baseline. Additionally, regularized models become significantly more robust against corruptions having predominantly high frequency (HF) perturbations (see Yin et al. (2019) for spectrums) such as *pixelate* and *defocus/glass/gaussian blur*. Perhaps less surprisingly, regularized models become less sensitive to increased *JPEG compression*, as it relies on (quantized) DCT-II coefficients. For corruptions with larger variance in the frequency spectrum, regularized performance remains largely unchanged. We see a slight degradation of performance in low frequency (LF) corruptions such as *brightness*, *saturation*, *contrast*, and *impulse noise*. However, the accuracy drop is relatively low considering the evaluation at the highest severity level.

Other datasets. Although several works reported a shift in the frequency band of adversarial attacks depending on the dataset (Maiya et al., 2021; Abello et al., 2021; Bernhard et al., 2021; Ortiz-Jiménez et al., 2020), we consequently see an improvement due to our HF regularization on multiple datasets (Tab. 1). Arguably, we see smaller improvements for SVHN/Tiny-ImageNet - which are also the datasets that show

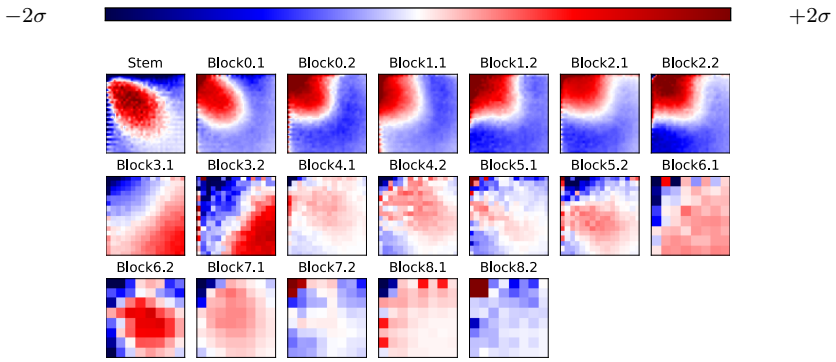


Figure 6: Frequency heat maps of **feature maps**: Regularized layers (top row) of a ResNet-20 show a shift towards LF after regularization, compared to non-regularized layers in rows two and three.

Table 4: Results on **ImageNet** for EfficientNet-B0 and ConvNeXt-Tiny on clean training data, FGSM, PGD-40 ($L^\infty, \epsilon = 1/255$), ImageNet-C and out-of-distribution generalization test. All regularization hyper-parameters are $\lambda = 0.05$ and $h = 3$.

Model	Variant	Clean Test Acc. (↑)		Adversarial Attacks Acc. (↑)		Corruption Error (↓)	Cue-Conflict (↑)	Sketch (↑)		Stylized (↑)		Edge (↑)	Silhouette (↑)
		Top 1	Top 5	FGSM	PGD-40	ImageNet-C		Top 1	Top 5	Top 1	Top 5	Top 1	Top 1
EfficientNet-B0	CNN	75.44	92.86	16.89	2.32	54.54	23.52	65.25	84.62	52.25	79.00	35.00	51.25
	WD	**75.80**	**92.94**	14.86	2.03	53.99	22.58	66.12	86.25	48.25	78.88	**40.62**	55.00
	SD	75.62	92.82	15.05	1.41	52.85	23.67	**66.38**	84.50	**52.50**	78.50	34.38	58.13
	WD + Reg.	75.44	92.15	18.45	4.43	52.03	**29.38**	**66.38**	86.88	47.62	**79.75**	36.25	58.13
	SD + Reg.	74.42	92.19	**18.70**	**5.33**	**51.12**	25.78	64.75	**87.88**	49.12	77.12	32.50	**58.75**
ConvNeXt-Tiny	CNN	**81.32**	95.53	35.53	3.93	**41.92**	24.84	71.50	88.00	56.00	78.38	**48.12**	62.50
	WD	81.11	**95.55**	35.30	2.97	42.98	25.31	73.12	**89.62**	52.00	77.62	47.50	58.75
	WD + Reg.	79.25	94.38	**35.69**	**4.22**	44.31	**32.27**	**73.75**	88.12	**58.00**	**82.38**	38.75	**65.62**

more LF perturbations than HF. Contrary to our CIFAR-10 results, WD outperforms SD on all datasets except SVHN.

Spectrum of feature maps. Further, we aim to understand the implications of the regularization on the computed feature maps. Exemplarily, we compare an *SD + Reg.* ResNet-20 against a CNN baseline and analyze the magnitude shift in the DCT-II coefficients of the feature maps (Fig. 6) of a clean validation batch. Our regularization causes a clear shift towards lower frequencies in regularized layers. Interestingly, in the stem layer, we also see large shifts from entirely vertical or horizontal frequencies to more balanced ones. Contrary, non-regularized (deeper) layers appear to slightly shift towards higher frequencies.

Comparison to other methods. We compare our method to *FrequencyLowCut Pooling (FLC)* (Grabinski et al., 2022a) and *Patch Gaussian Augmentation (PaGA)* (Lopes et al., 2019) (Tab. 2) as these methods also aim at HF-regularization. Regarding AA and CC performance, we observe that our method consistently outperforms both other approaches in standalone comparisons with small degradation of clean validation accuracy. Additionally, imposing our regularization on top of other methods improves the robustness of these significantly. Interestingly, we often get the highest levels of robustness in combination with another method, proving that our regularization can be complementary to other robustness techniques. For a comparison to Wang et al. (2020), please refer to the appendix, where we show favorable behavior of our approach.

5.2 ImageNet

Next, we aim to explore how our regularization performs on the common ImageNet dataset (Deng et al., 2009). In particular, more OOD tests exist for this dataset which allows us to study aspects outside adversarial robustness, and robustness against common corruptions. Similar to our results on other datasets, we see an improvement in adversarial robustness at slight (1-2%) degradation of clean performance (Tab. 4). While we see an improvement in CC performance on EfficientNet, we see an equal decrease for ConvNeXt. This may be due to the larger kernels (7×7) that ConvNeXt utilizes and may, thus, require other hyper-

Table 5: **FGSM-Adversarial Training on CIFAR-10** with L^∞, $\epsilon = 8/255$. We report the mean over 5 runs for the FGSM train and validation accuracy of the epoch of the best PGD-40 validation accuracy as well as the AutoAttack accuracy. We also report the corresponding mean accuracy on CIFAR-10-C and report the difference to the clean-trained evaluations.

	Variant	**Clean**	**FGSM** (↑)	**Adversarial Acc.** (↑)		**Corruption Acc.**	
		Val. Acc. (↑)	**Train Acc.**	**PGD-40**	**AA**	Mean (↑)	Δ (AT-Normal) (↑)
ResNet-20	CNN	**73.73**	**50.39**	46.14	36.09	**66.99**	-0.97
	WD + Reg.	71.69	48.46	45.38	35.64	65.48	-8.56
	SD + Reg.	73.01	49.93	**46.34**	**36.47**	66.73	-7.41
ResNet-9	CNN	81.70	60.27	**52.77**	0.00	74.06	0.68
	WD + Reg.	81.56	61.66	51.52	39.97	74.47	-1.94
	SD + Reg.	**82.56**	**63.39**	51.80	**40.14**	**75.40**	-2.32
EfficientNet-B0	CNN	63.00	42.34	42.49	34.04	57.35	-10.74
	WD + Reg.	68.50	45.87	45.13	36.56	62.66	-9.48
	SD + Reg.	**68.89**	**46.76**	**45.57**	**36.76**	**63.07**	-8.8

Table 6: **PGD-Adversarial Training on ImageNet** of ResNet-50 with L^∞, $\epsilon = 4/255$. We report the train and validation accuracy under PGD attacks, validation accuracy under AutoAttack, corruption error, and cue-conflict. Results are from one run.

Variant	**Clean**	**PGD** (↑)	**Adversarial Acc.** (↑)		**Corruption**	**Cue-Conflict** (↑)
	Val Acc. (↑)	**Train Acc.**	**PGD**	**AA**	**Error** (↓)	
CNN	56.85	33.88	36.04	22.33	78.65	38.83
WD	55.82	33.91	35.06	22.06	78.90	38.83
WD + Reg.	**58.09**	**36.00**	**37.09**	**24.32**	**78.36**	**39.38**

parameters. Importantly we see a significant improvement of the *cue-conflict* in both cases - which is also reflected in the increased accuracy of *silhouette* (LF) and the decrease in performance of *edge* (HF). This indicates that our regularization favorably shifts models toward shape bias (Geirhos et al., 2019).

5.3 Integration into adversarial training and impact on robust overfitting

So far, we investigated the effect of our proposed HF regularization on native robustness. For completeness, we aim to explore the role of our regularization in AT. We train our models against FGSM-adversaries on CIFAR-10 ($L^\infty, \epsilon = 8/255$) (Tab. 5). To mitigate robust overfitting, we use early stopping based on PGD-40 test performance. We observe, that our regularization has a beneficial effect on the out-of-domain attacks (*i.e.* AA) and all runs show an increased performance after regularization. We furthermore observe that our regularization appears to mitigate *robust overfitting* of training attacks (similarly to Grabinski et al. (2022a)) on ResNet-9: without regularization the AT-trained CNN achieves high FGSM train accuracy and high PGD-40 validation accuracy but fails to generalize to other attacks (AA) and stagnates at 0%. With regularization, all runs show comparable or even better accuracy than the best non-regularized models. However, similarly to the observations by Saikia et al. (2021), we generally see a significant decrease in CC accuracy due to AT. Again, this demonstrates that AT is not the cure-all to improve network robustness and there is a need for other approaches such as our proposed frequency regularization.

Additionally, we extend our experiments to AT on ImageNet (Tab. 6). Here we switch to single-step PGD training with the common $\epsilon = 4/255$ and train the ResNet-50 architecture. Again we report PGD, AA, and CC performance but this time also the cue-conflict score. Our regularized WD architecture outperforms the baseline in all metrics: adversarial robustness, corruption error, and cue-conflict. This demonstrates that our method is able to mitigate some of the overfitting aspects of adversarial training and leads to an improved OOD generalization performance. We do not report the (regularized) SD performance due to the lack of tuned hyperparameters but expect similar gains when tuned properly.

6 Conclusion

We have shown a first step towards improving the native robustness of CNNs to multiple distribution shifts such as adversarial attacks, corruptions, and shape-biased datasets, as well as the benefit of our regularization for adversarial training. In particular, our regularization decreases the sensibility to high-frequency perturbations. Albeit our results do not approach SOTA levels, we emphasize that we improve robustness on a wide range of tests, whereas SOTA methods like AT often overfit to one specific type of robustness, such as adversarial attacks, and often even impair performance on other tests compared to normal baselines. Additionally, our method does not rely on OOD examples but intrinsically strengthens the model. Our approach has shown to generalize to different networks with various kernel sizes, that were trained on different datasets, and different measures of robustness. We have also shown that our method can be used in combination with other approaches such as *PaGA* (Lopes et al., 2019), *FLC* (Grabinski et al., 2022a), and even AT (Madry et al., 2018) to further improve robust performance. In combination with AT, our approach shows promise to mitigate *robust overfitting* (Rice et al., 2020).

Limitations. We observed that on some architectures switching to WD/SD introduces a significant drop in accuracy (prior to regularization). Although the forward pass of both methods is mathematically equivalent to baselines, the backward pass is not. *E.g.* weight updates on linear combinations of decomposed convolution filters and feature maps are in different backward pass stages and under different quantization conditions due to limited bit precision. While we observe that our regularization generally improves a multitude of robustness aspects, the regularized counterparts may underperform CNN baselines due to the initial impairment due to the architecture change. We aim to explore more root causes and alternatives in future work.

References

Antonio A. Abello, Roberto Hirata, and Zhangyang Wang. Dissecting the high-frequency bias in convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pp. 863–871. Computer Vision Foundation / IEEE, 2021.

N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23 (1):90–93, 1974.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*. Springer International Publishing, 2020.

Rémi Bernhard, Pierre-Alain Moëllic, Martial Mermillod, Yannick Bourrier, Romain Cohendet, Miguel Solinas, and Marina Reyboz. Impact of spatial frequency based constraints on adversarial robustness. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*. IEEE, 2021.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness, 2022.

Karol Cheinski and Pawel Wawrzynski. Dct-conv: Coding filters in convolutional networks with discrete cosine transform. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, 2020.

Wai Chen. *The Electrical Engineering Handbook*. Academic Press, October 2004. ISBN 978-0-08047748-0.

Wenlin Chen, James T. Wilson, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen. Compressing convolutional neural networks in the frequency domain. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1475–1484. ACM, 2016.

Yaosen Chen, Renshuang Zhou, Bing Guo, Yan Shen, Wei Wang, Xuming Wen, and Xinhua Suo. Discrete cosine transform for filter pruning. *Applied Intelligence*, 2022.

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

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack, 2020b.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Samuel F. Dodge and Lina J. Karam. A study and comparison of human and deep learning recognition performance under visual distortions. *CoRR*, abs/1705.02498, 2017. URL http://arxiv.org/abs/1705.02498.

Ranjie Duan, Yuefeng Chen, Dantong Niu, Yun Yang, A. K. Qin, and Yuan He. Advdrop: Adversarial attack to dnns by dropping information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7506–7515, October 2021.

Paul Gavrikov and Janis Keuper. Cnn filter db: An empirical investigation of trained convolutional filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022a.

Paul Gavrikov and Janis Keuper. Adversarial robustness through the lens of convolutional filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2022b.

Paul Gavrikov and Janis Keuper. On the interplay of convolutional padding and adversarial robustness, 2023.

Paul Gavrikov, Janis Keuper, and Margret Keuper. An extended study of human-like behavior under adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2360–2367, June 2023.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Julia Grabinski, Steffen Jung, Janis Keuper, and Margret Keuper. Frequencylowcut pooling - plug and play against catastrophic overfitting. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XIV*, pp. 36–57. Springer, 2022a.

Julia Grabinski, Janis Keuper, and Margret Keuper. Aliasing and adversarial robust generalization of cnns. *Machine Learning*, 2022b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

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

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Md Tahmid Hossain, Shyh Wei Teng, Dengsheng Zhang, Suryani Lim, and Guojun Lu. Distortion robust image classification using deep convolutional neural network with discrete cosine transform. In *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*, pp. 659–663. IEEE, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale, 2017.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Shao-Yuan Lo and Hsueh-Ming Hang. Exploring semantic segmentation on the DCT representation. In *MMAsia '19: ACM Multimedia Asia, Beijing, China, December 16-18, 2019*, pp. 59:1–59:6. ACM, 2019.

Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D. Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *CoRR*, abs/1906.02611, 2019.

Peter Lorenz, Dominik Strassel, Margret Keuper, and Janis Keuper. Is robustbench/autoattack a suitable benchmark for adversarial robustness? In *The AAAI-22 Workshop on Adversarial Machine Learning and Beyond*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Shishira R. Maiya, Max Ehrlich, Vatsal Agarwal, Ser-Nam Lim, Tom Goldstein, and Abhinav Shrivastava. A frequency perspective of adversarial robustness. *CoRR*, abs/2111.00861, 2021. URL https://arxiv.org/abs/2111.00861.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. *NIPS*, 01 2011.

Guillermo Ortiz-Jiménez, Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Hold me tight! influence of discriminative features on deep network boundaries. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning*, 2017. URL http://arxiv.org/abs/1707.04131.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8093–8104. PMLR, 13–18 Jul 2020.

Daniel L. Ruderman. The statistics of natural images. *Network: Computation In Neural Systems*, 5:517–548, 1994.

Tonmoy Saikia, Cordelia Schmid, and Thomas Brox. Improving robustness against common corruptions with frequency biased models. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 10191–10200. IEEE, 2021.

Nandini C. Singh and Frédéric Theunissen. Modulation spectra of natural sounds and ethological theories of auditory processing. *The Journal of the Acoustical Society of America*, 114:3394–411, 01 2004. doi: 10.1121/1.1624067.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. 2019. doi: 10.48550/ARXIV.1905.11946. URL https://arxiv.org/abs/1905.11946.

Florian Tramèr and Dan Boneh. *Adversarial Training and Robustness for Multiple Perturbations*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Matej Ulicny, Vladimir A. Krylov, and Rozenn Dahyot. Harmonic convolutional networks based on discrete cosine transform. *Pattern Recognition*, 129:108707, 2022. ISSN 0031-3203.

G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019.

Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020.

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

Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 1737–1746. Computer Vision Foundation / IEEE, 2020.

Leonid Yaroslavsky. Fast transforms in image processing: Compression, restoration, and resampling. *Advances in Electrical Engineering*, 2014:1–23, 07 2014. doi: 10.1155/2014/276241.

Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 13255–13265, 2019.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? volume 4, 2014.

Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu. Understanding robust overfitting of adversarial training and beyond. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25595–25610. PMLR, 17–23 Jul 2022.