Supplementary material:
Fourier-basis functions to bridge augmentation gap: Rethinking frequency augmentation in image classification

Puru Vaish* Shunxin Wang* Nicola Strisciuglio
University of Twente
{p.vaish, s.wang-2, n.strisciuglio}@utwente.nl

1. Implementation Details

Below, we report the training setup in detail. For all methods, and a particular dataset and architecture, the same training setup was used unless stated otherwise.

Convolution Neural Networks For CIFAR-100 and Tiny ImageNet we use the SGD optimiser with an initial learning rate of 0.2, Nesterov momentum of 0.9 with a batch size of 128 training for 100 epochs. We use a weight decay of 0.0005 and we do not decay the affine parameters of normalisation. For CIFAR-10, we follow the same setup as above, except we train for 200 epochs with a batch size of 256 and an initial learning rate of 0.1. The learning rate is decayed with a cosine annealing schedule to 0 which is stepped step-wise. For all models, we always employ the standard transformation of random crop with a padding of 4 and random horizontal flip.

For ImageNet, we follow [3] in that we use SGD optimiser with an initial learning rate of 0.1 and Nesterov momentum of 0.9 and train for 90 epochs. We use a weight decay of 0.0001 and we do not decay affine parameters of normalisation. The learning rate decays with a by a factor of 0.1 every 30 epochs. For all models, we employ the standard transformation of random resized crop to image size of 224×224 with bilinear interpolation and random horizontal flip, before other augmentations.

We choose to train all models from scratch (no fine-tuning using AFA) so that we can study the effects of AFA without other underlying factors. Therefore, for fair comparison, we retrain PRIME from scratch as well using our setup. For models trained with JSD, we follow [5] for the regularising coefficient, mainly: $\lambda = 10$ for CIFAR-10 and Tiny ImageNet, $\lambda = 1$ for CIFAR-100 and $\lambda = 12$ for ImageNet.

We only use the main BN layers during testing, similar to AugMax for all convolution models.

Compact Convolution Transformer For CIFAR-10/100 and ImageNet we also train a transformer architecture. For all datasets we use CutMix (alpha=1.0) and MixUp (alpha=0.2 for ImageNet and alpha=1.0 for CIFAR-10/100) with an equal chance of applying one of the two. For CIFAR-10/100, we follow [1]. We train using the AdamW optimiser with max learning rate of 0.0006 and weight decay of 0.06, and we do not decay the affine parameters of the normalisation modules. We train with an effective batch size of 256, and apply learning rate decay following a cosine decay with a warm-up period of 10 epochs and the learning rate scheduler is stepped step-wise. For ImageNet, we use a max learning rate of 0.0005, effective batch size of 1024 and a weight decay of 0.05. The learning rate decay follows a cosine annealing schedule with a warm-up of 25 epochs. The same standard transformations as for convolutional neural networks were applied.

2. Evaluation metrics

Mean corruption error (mCE) measures the robustness of models against image corruptions [2], computed as:

$$\text{mCE} = \frac{1}{|C|} \sum_{c \in C} \frac{\sum_{s=1}^{5} E_{s,c}^{f}}{\sum_{s=1}^{5} E_{s,c}^{baseline}}, \tag{1}$$

where the sum of classification error E of five severity $s \in \{1,2,3,4,5\}$ per corruption c of model f is normalized by that of a baseline model. The normalized classification errors of all corruptions C in the dataset are averaged to obtain mCE. We use AlexNet as baseline in ImageNet experiments and ResNet-18 for Tiny ImageNet. For CIFAR-10/100 there are no baselines advised so we do not report the mCE for these datasets.

Mean flip rate (mFR) evaluates the consistency of model predictions with increasing perturbations [2], computed as

[*] Equal contribution

follows:

$$\mathrm{mFR} = \frac{1}{|C|}\sum_{c\in C}\mathrm{FR}_c^f = \frac{1}{|C|}\sum_{c\in C}\frac{\mathrm{FP}_c^f}{\mathrm{FP}_c^{baseline}}, \tag{2}$$

with

$$\mathrm{FP}_c^f = \frac{1}{m(n-1)}\sum_{i=1}^{m}\sum_{j=2}^{n}\mathbb{1}(f(x_j^{(i)}) \neq f(x_{j-1}^{(i)})). \tag{3}$$

$\mathbb{1}(f(x_j^{(i)}) \neq f(x_{j-1}^{(i)}))$ measures whether the prediction of the model f on a frame x_j is the same as its previous perturbed frame in the i^{th} sequence. If the predictions are the same, $\mathbb{1}(f(x_j^{(i)}) \neq f(x_{j-1}^{(i)}))$ equals to zero, and thus the performance of the model is not affected by the considered perturbations. FP_c^f measures the consistency of predictions over m perturbed sequences, each with n of frames. For a sequence corrupted by noise, the predictions are compared with those of the first frame, as noise is not temporally related. The mFR is obtained by averaging the normalized FP_c^f by that of a baseline model across all the perturbations C. The value of mFR is expected to be close to zero for a robust model.

Mean top-5 distance (mT5D) also measures the consistency of model predictions in terms of increasing perturbations [2]. For a robust model, the top-5 predictions of frames over a sequence should be relevant to those of the previous frames in the sequence. The top-5 distance thus measures the inconsistency of top-5 predictions under consecutive perturbations, computed as follows:

$$\mathrm{T5D}_c^f = \frac{1}{m(n-1)}\sum_{i=1}^{m}\sum_{j=2}^{n} d(\tau(x_j), \tau(x_{j-1})), \tag{4}$$

with

$$d(\tau(x_j), \tau(x_{j-1})) = \sum_{i=1}^{5}\sum_{j=min\{i,\rho(i)\}+1}^{max\{i,\rho(i)\}}\mathbb{1}(1 \leq j-1 \leq 5), \tag{5}$$

where $\rho(\tau(x_j)(k)) = \tau(x_{j-1})(k)$, $\tau(x_j)$ is the ranking of predictions for a perturbed frame x_j and $\tau(x_j)(k)$ indicates the rank of the prediction being k. If $\tau(x_j)$ and $\tau(x_{j-1})$ are the same, then $d(\tau(x_j), \tau(x_{j-1})) = 0$. Averaging the normalized $\mathrm{T5D}$ by that of the baseline over all corruptions obtain $\mathrm{mT5D} = \frac{1}{|C|}\sum_{c\in C}\frac{\mathrm{T5D}_c^f}{\mathrm{T5D}_c^{baseline}}$.

Fourier heatmap evaluates model robustness from a Fourier perspective [6] exploiting Fourier basis functions to perturb test images and measuring the classification error of models. They are constructed as follows. Let $U_{i,j} \in \mathbb{R}^{d_1\times d_2}$ be a real-valued matrix such that its norm equals to 1. The Fourier transform of $U_{i,j}$ has only two non-zero elements located at (i,j) and the corresponding symmetric coordinate with respect to the image center. Given an image X, a perturbed image with Fourier basis noise can be generated by $\tilde{X}_{i,j} = X + rvU_{i,j}$, where r is chosen randomly from a uniform distribution ranging from -1 to 1, and v controls the strength of the added noise. Each channel of the images is perturbed independently with different r and v. The model robustness against Fourier basis noise $U_{i,j}$ is evaluated by the classification error, and the final outcome is in a form of heatmap which records the error of the evaluated model under different Fourier basis noise. Examples are in Fig. 1.

			TIN-C		
-	Main	Auxiliary	SA↑	RA↑	mCE↓
ResNet18	-	✗	63.56	25.86	97.34
	AFA	✗	59.04	28.87	93.45
	-	AFA	62.52	33.35	87.58
	AugMix	✗	62.95	36.26	84.05
	AugMix	AFA	62.51	38.67	80.83
	AugMix†	✗	**64.65**	36.30	83.90
	AugMix†	AFA	64.34	38.52	80.79
	PRIME	✗	63.07	39.67	79.42
	PRIME	AFA	62.48	41.09	77.55
	PRIME†	✗	63.24	41.22	77.44
	PRIME†	AFA	62.65	**43.00**	**73.11**

Table 1. Results for TIN-C with ResNet18. Models with † use loss with JSD.

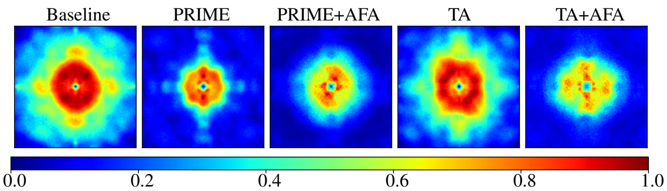


Figure 1. Fourier heatmaps of CCT trained with standard setting, PRIME, PRIME+AFA, TA and TA+AFA.

3. Supplementary results

3.1. Results on Tiny ImageNet

In Tab. 1 we provide the robustness results on Tiny ImageNet (TIN), which are consistent with those presented on other datasets. Models trained with AFA show robustness improvements consistently by significant margin with only negligible reduction of the clean accuracy. We again see that JSD improves robustness slightly, and in AugMix it improves clean accuracy greatly.

3.2. Robustness in the frequency spectrum.

The Fourier heatmaps of CCT trained with standard setting, PRIME, PRIME+AFA, TA and TA+AFA are provided

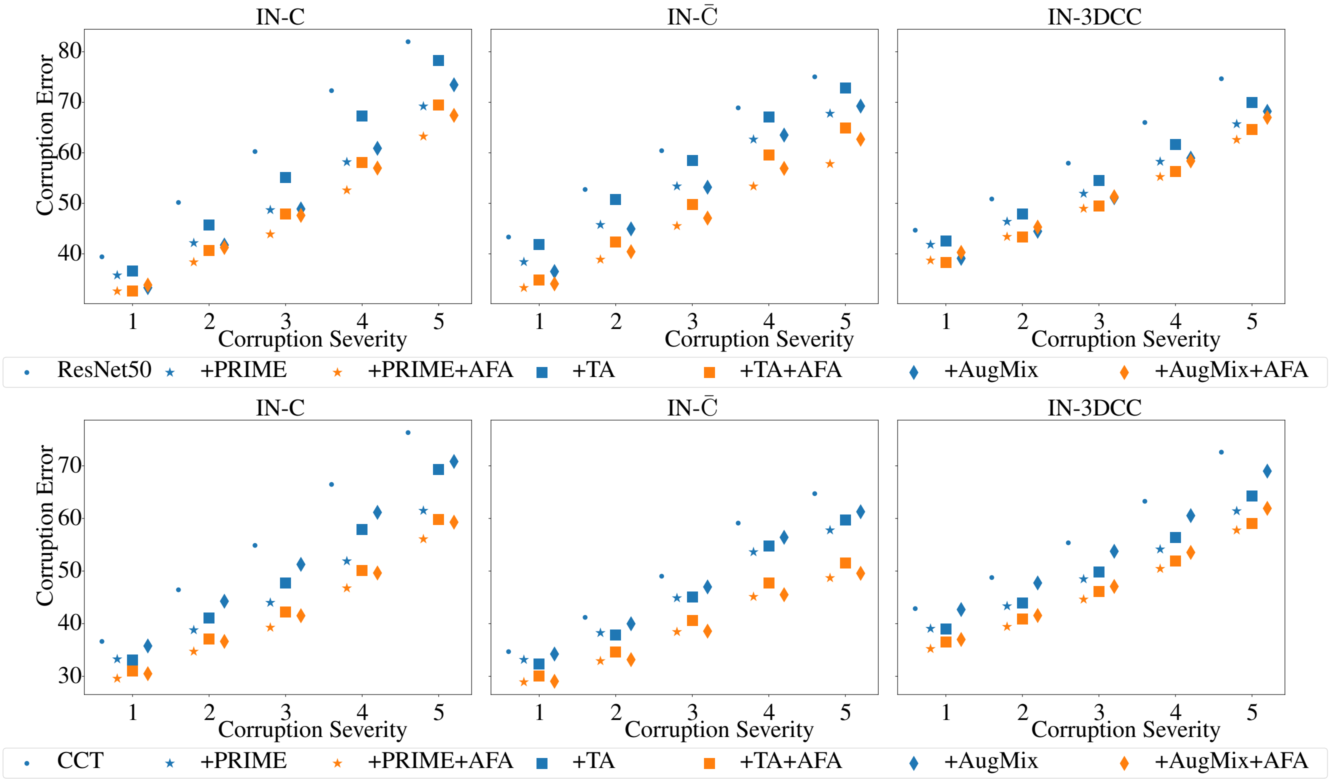


Figure 2. Corruption error of ResNet50 and CCT trained with PRIME, PRIME+AFA, TA, TA+AFA, AugMix and AugMix+AFA. Models trained with AFA (orange points) have lower error at each severity than their counterpart trained with only visual augmentation (blue points), demonstrating the benefit of AFA to corruption robustness.

in Fig. 1. Our observations are consistent with those in the main paper. Also CCT models trained with the contribution of AFA have better robustness to low and middle-high frequency corruptions.

3.3. Robustness per corruption severity.

We report the classification error of models tested under corruptions with different severity levels Fig. 2. The models trained with AFA have consistently lower error than their counterpart trained without AFA, showing that AFA can further boost the robustness of models against common image corruptions, especially in difficult testing conditions with high severity.

3.4. Robustness to each image corruption.

Furthermore, we show the classification error averaged over five corruption severity levels per corruption type in Fig. 4. The error points of model trained with visual augmentations only, and with further use of AFA are connected by a line. A downward trend means models trained with AFA have better robustness performance on specific corruption types. We observe that, in general, models with AFA have better corruption robustness than models trained only with visual augmentations. Significant improvements are especially evident on noise corruptions (Gaussian noise, impulse noise, iso noise, plasma noise, shot noise, single frequency grayscale noise and cocentric sine waves). One exception is ResNet50 trained with AugMix and AFA, for which the model trained without AFA performs better except on few cases. This can be attributed to the less training time (90 epoch vs 180 epochs) than that of ResNet50+AugMix.

4. Evidence of adversarial nature of AFA

4.1. Main and auxiliary batch normalisation

For the ResNet architecture, which includes Batch Normalisation layers, we had replaced the Batch Normalisation layers with DuBIN layers [5] while operating the Auxiliary setting. Assuming that there is no difference in the distribution of images augmented using AFA and a typical visual augmentation technique, there should be no difference in the affine parameters learnt for each individual batch normalisation parameter (the main and the auxiliary).

We show in Fig. 5 the Mean Absolute Difference of the same parameter between the main and the auxiliary component of the DuBIN layer at different depths of the model. We show the results for for models trained with ACE loss

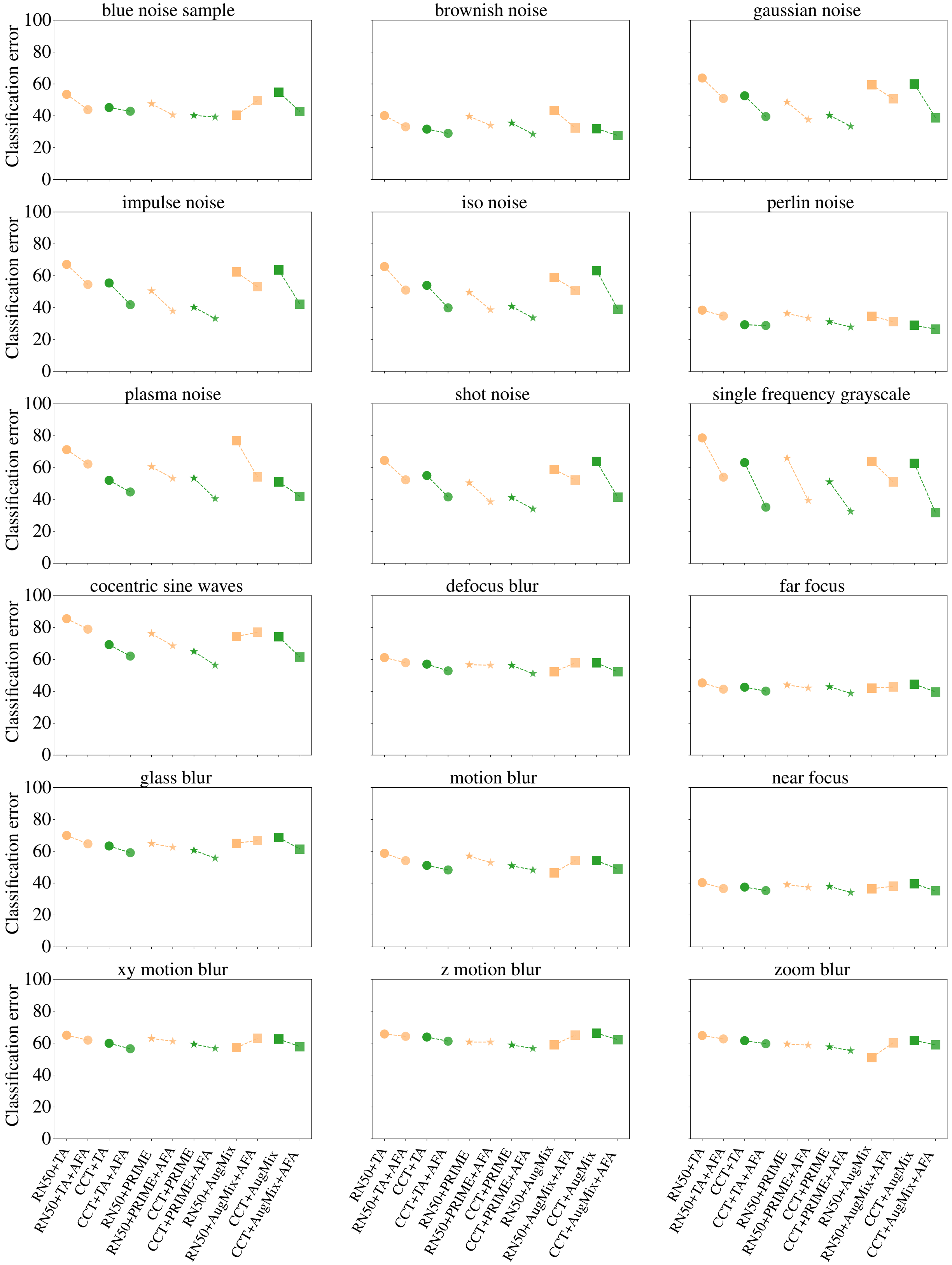


Figure 3. Averaged classification error per corruption of ResNet50s (orange) and CCTs (green). The error points of model trained with visual augmentations and additionally with AFA are connected. A decreasing line indicates better performance when trained additionally with AFA (a).

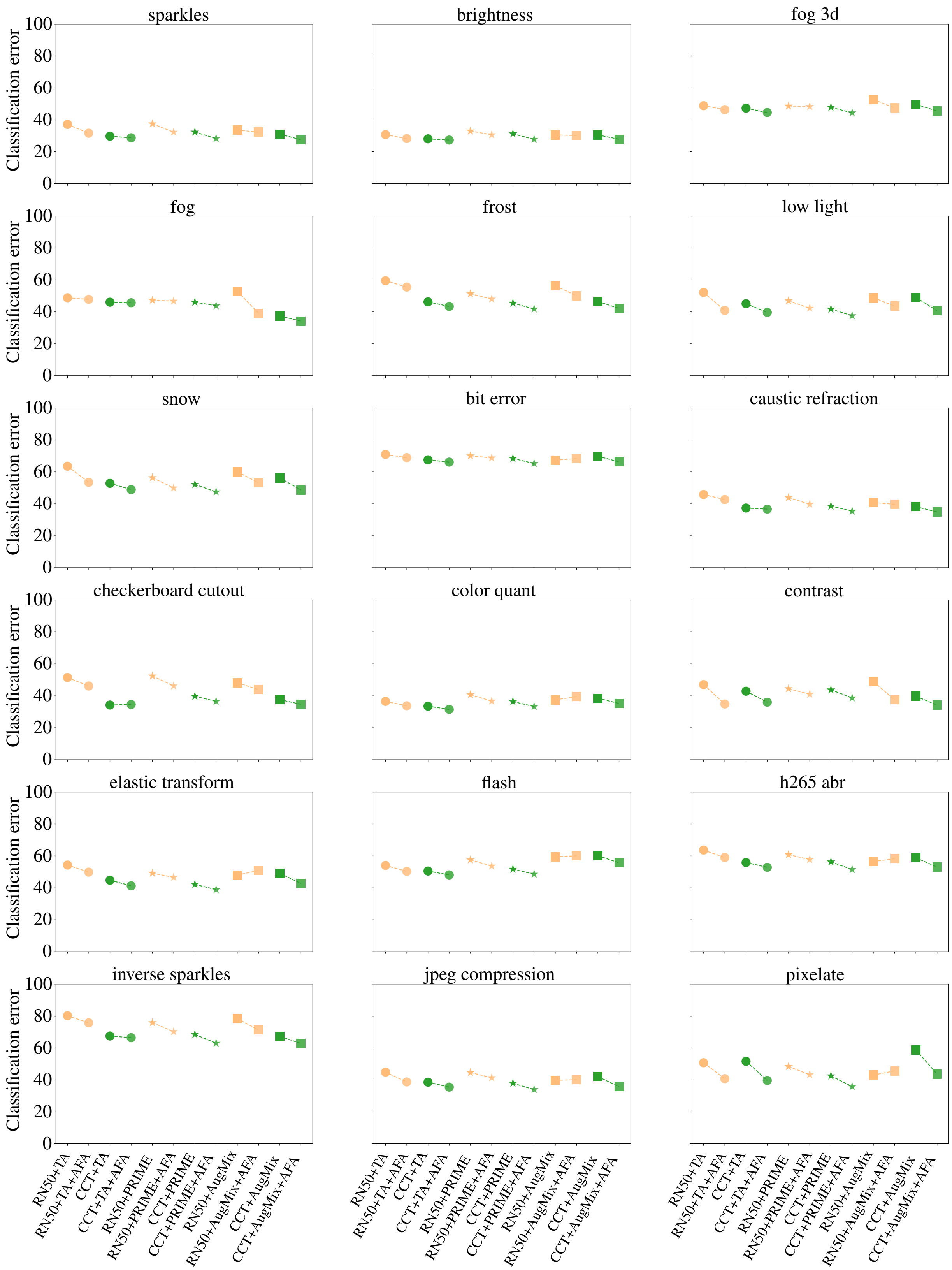


Figure 4. Averaged classification error per corruption of ResNet50s (orange) and CCTs (green). The error points of model trained with visual augmentations and additionally with AFA are connected. A decreasing line indicates better performance when models are trained additionally with AFA (b).

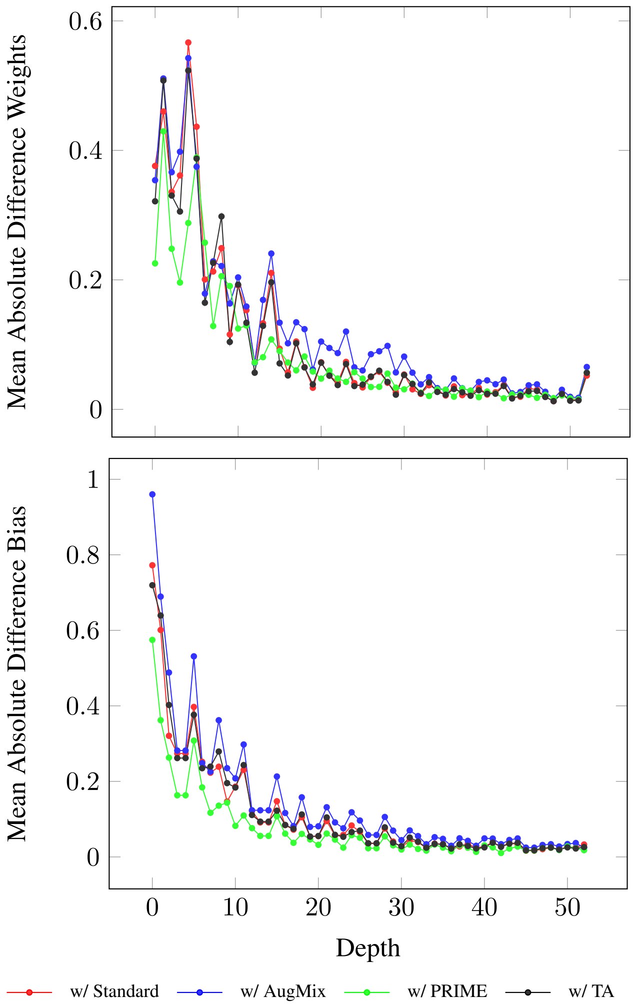


Figure 5. Comparison of the mean absolute difference of the learnt affine parameters for the two batch normalisations in the Dual Batch Norm Layers of ResNet50-DuBIN architecture at different depths.

for ResNet-50 where AFA is paired with just standard transforms, AugMix, PRIME and Trivial Augment (TA).

We can see that at earlier depths the parameter differ largely, which is explained by the difference in distribution of a visually augmented and AFA augmented image. This difference converges to a lower value, which is again explained by the model attempting to extract similar features from the differently augmented images.

4.2. Embedding Space Visualization

We compare how diverse are the augmentations of AFA are with respect to other methods. We follow the procedure in [4]. To reiterate the procedure, we randomly select 3 images from ImageNet, each one belonging to a different class. For each image, we generate 100 transformed instances using Standard Transform, Trivial Augment, PRIME, PGD attack with the following parameters: 5 steps, epsilon of $8/255$ and alpha of $2/255$, and with AFA. Then, we pass the transformed instances of each method through a ResNet-50 pretrained on ImageNet using standard transform and training setup, and extract the features of its embedding space from the penultimate layer before the dense layer. On the features extracted for each method, we perform PCA after whitening and then visualize the projection of the features onto the first two principal components. We visualize the projected augmented space in Fig. 6, which demonstrates that AFA generates which are more akin to an adversarial attack rather than a standard augmentation. This is clear from a visual similarity of AFA's result in Fig. 6e to PGD's result in Fig. 6d and dissimilarity to the other Visual Augmentation techniques.

Finally, we also add in Fig. 6f the embedding space visualisation for the Auxiliary Trained model with AFA augmentation and standard transform for main, following the same procedure as above. We see that the model learns more separable embeddings for images augmented with AFA using the auxiliary setting, therefore is less sensitive to Frequency perturbation. The embeddings also retain a large variance and hardness, therefore showcasing the diversity of the augmentations of AFA.

5. Regularisation Effect

In Fig. 7 we show the norm of the weights of the convolutional kernels for the ResNet50 models trained with and without AFA at each depth. We see that AFA provides a strong regularisation effect that is akin to the regularisation effect of PRIME. Meanwhile, we see that AugMix does not regularise the weights at all compared to the baseline model with only the standard transforms. The weights are however regularised to when AFA is paired with AugMix. Combined with PRIME, there does not seem to be further regularisation of the weights.

6. Proof of Augmenting Fourier Domain

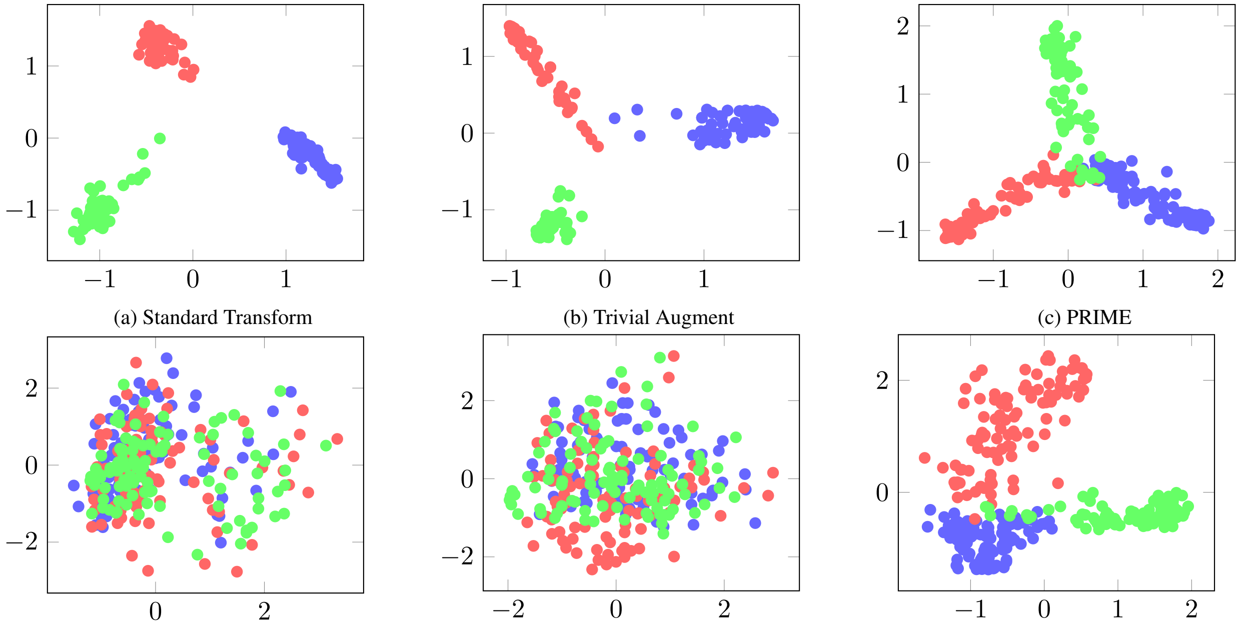
Lemma 1 (Linearity)**.** *Let f, g be functions of a real variable and let $\mathscr{F}(f)$ and $\mathscr{F}(g)$ be their Fourier transforms. Then for complex numbers a and b*

$$\mathscr{F}(af+bg) = a\mathscr{F}(f) + b\mathscr{F}(g), \tag{6}$$

therefore, Fourier transform $\mathscr{F}$ is a linear transformation.

Lemma 2 (Fourier Transform of Plane Wave)**.** *The Fourier transform of the planar wave given by the frequency f and the direction ω, $A_{f,\omega}$ has a fourier transform*

$$\begin{aligned} \mathscr{F}(A_{f,\omega}) &= \mathscr{F}\left(R\cos(2\pi f(u\cos(\omega) + v\sin(\omega)))\right) \quad (7)\\ &= \frac{R}{2}\left(\delta(\hat{x},\hat{y}) + \delta(\bar{x},\bar{y})\right), \quad (8) \end{aligned}$$



(a) Standard Transform

(b) Trivial Augment

(c) PRIME

(d) PGD Attack

(e) AFA on Standard

(f) AFA on Auxiliary Trained

Figure 6. Differences in the Embedding Space for Different Methods and PGD Attack. From (a)-(e) the standardly trained model is used, and for (f) the model trained in the auxiliary setting is used.

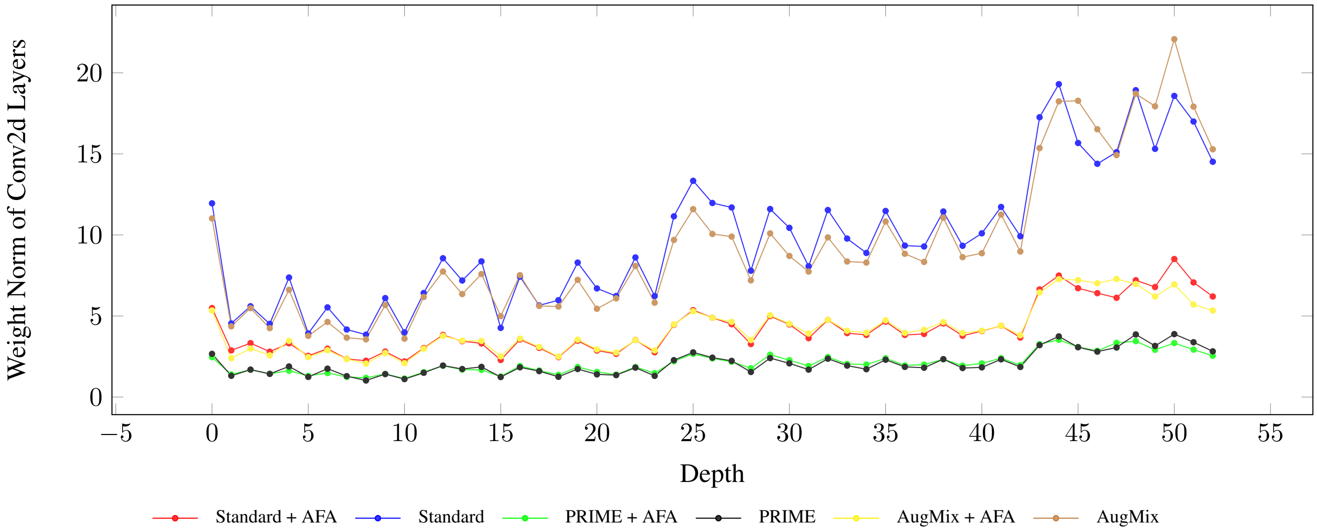


Figure 7. The norm of the Conv2d Layers for ResNet 50 trained with different augmentation techniques with and without AFA. The plot highlights the regularisation effect the methods have on the model weights.

where, $\hat{x} = x - f\cos(\omega)$*,* $\hat{y} = y - f\sin(\omega)$ *and* $\bar{x} = x + f\cos(\omega)$*,* $\bar{y} = y + f\sin(\omega)$.

Theorem 1 (AFA Augments the Fourier Domain)**.** *Given an image sample* s*, an augmentation using AFA produces as augmentation in the Fourier domain of the image for one specific frequency and orientation of the wave* (f, ω)*.*

Proof. Given image s and the randomly sampled planar wave using AFA, $\sigma A_{f,\omega}$, dropping the subscript for the channels for clarity, we have:

$$\begin{aligned}\mathscr{F}(\text{AFA}(s)) &= \mathscr{F}(s + \sigma A_{f,\omega}) \\ &= \mathscr{F}(s) + \sigma \mathscr{F}(A_{f,\omega}) \qquad (9) \\ &\text{(using Lemma 1)} \\ &= \mathscr{F}(s) + \frac{\sigma R}{2}\left(\delta(\hat{x}, \hat{y}) + \delta(\bar{x}, \bar{y})\right). \qquad (10) \\ &\text{(using Lemma 2)}\end{aligned}$$

Therefore, we prove augmenting an image s with AFA corresponds to augmenting the amplitude of a specific frequency component (f,ω) in the 2D Fourier transform of the image. □

References

[1] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the Big Data Paradigm with Compact Transformers. *arXiv*, Apr. 2021. 1

[2] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019. 1, 2

[3] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. *arXiv*, Dec. 2019. 1

[4] Apostolos Modas, Rahul Rade, Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. PRIME: A few primitives can boost robustness to common corruptions. *arXiv*, Dec. 2021. 6

[5] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. AugMax: Adversarial Composition of Random Augmentations for Robust Training. *arXiv*, Oct. 2021. 1, 3

[6] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. A Fourier Perspective on Model Robustness in Computer Vision. *arXiv*, June 2019. 2