HybridAugment++: Unified Frequency Spectra Perturbations for Model Robustness - Supplementary Material

Mehmet Kerim Yucel[1] Ramazan Gokberk Cinbis[2] Pinar Duygulu[1]
[1]Hacettepe University [2]Middle East Technical University
mkerimyucel@gmail.com gcinbis@ceng.metu.edu.tr pinar@cs.hacettepe.edu.tr

1. Supplementary Material

1.1. State of the art comparison on CIFAR-C

In the main text, we provide detailed ablations on CIFAR10/100-C in the form of corruption robustness evaluation. Due to space limitations, we could not provide state-of-the-art results there; we provide these results here. We compare ourselves with methods which share our characteristics; no additional data or models to be used. We choose CutOut [4], Mixup [13], CutMix [12], adversarial training (AT) [7], AutoAugment (AA) [3], Augmix [5] and APR [2]. We take the results of these methods from [2]; we do not include CIFAR-100 clean accuracy results or ResNet18 results here since they are not available.

Corruption Robustness. Table 1 shows mCE values of other methods, as well as the best results provided in Table 1 of the main text. The inclusion of the state-of-the-art methods do not change the takeaway message; $\mathcal{HA}^{++}_{\mathcal{PS}}$ comfortably outperforms others on all datasets and architectures. Note that all variants of $\mathcal{HA}$ and $\mathcal{HA}^{++}$ either outperform or are competitive to all state-of-the-art methods.

Clean Accuracy. Table 1 shows clean accuracy values of other methods, as well as the best results provided in Table 2 of the main text. $\mathcal{HA}^{++}_{\mathcal{PS}}$ outperforms all other state-of-the-art methods, and the best CIFAR-10 result comes with $\mathcal{APR}_{\mathcal{P}} + \mathcal{HA}^{++}_{\mathcal{S}}$. Note that the best result on CIFAR-100 comes with $\mathcal{APR}_{\mathcal{P}} + \mathcal{HA}_{\mathcal{S}}$, which shows the effectiveness of our proposed methods.

	State-of-the-art methods							Single-only			Paired-only			Combined			$\mathcal{APR}_{\mathcal{P}}$[2] *with*	
	Orig	Cutout	Mixup	CutMix	AT	AugMix	AA	$\mathcal{APR}_{\mathcal{S}}$	$\mathcal{HA}_{\mathcal{S}}$	$\mathcal{HA}^{++}_{\mathcal{S}}$	$\mathcal{APR}_{\mathcal{P}}$	$\mathcal{HA}_{\mathcal{P}}$	$\mathcal{HA}^{++}_{\mathcal{P}}$	$\mathcal{APR}_{\mathcal{PS}}$	$\mathcal{HA}_{\mathcal{PS}}$	$\mathcal{HA}^{++}_{\mathcal{PS}}$	$\mathcal{HA}_{\mathcal{S}}$	$\mathcal{HA}^{++}_{\mathcal{S}}$
AllConv	30.8	32.9	24.6	31.3	28.1	15.0	29.2	14.8	16.8	13.9	21.5	20.8	16.7	11.5	12.0	**10.7**	11.9	11.2
DenseNet	30.7	32.1	24.6	33.5	27.6	12.7	26.6	12.3	15.0	11.1	20.3	18.4	14.2	10.3	10.9	**9.5**	10.6	10.2
WResNet	26.9	26.8	22.3	27.1	26.2	11.2	23.9	10.6	13.6	10.0	18.3	16.4	13.2	9.1	9.9	**8.3**	9.2	8.7
ResNeXt	27.5	28.9	22.6	29.5	27.0	10.9	24.2	11.0	13.2	9.9	18.5	17.6	13.2	9.1	10.3	**7.9**	9.3	8.7
Mean	29.0	30.2	23.5	30.3	27.2	12.5	26.0	12.1	14.6	11.2	19.6	18.3	14.3	10.0	10.7	**9.1**	10.2	9.7
AllConv	56.4	56.8	53.4	56.0	56.0	42.7	55.1	39.8	43.0	38.9	47.5	44.7	41.7	35.9	36.5	**34.4**	35.9	35.1
DenseNet	59.3	59.6	55.4	59.2	55.2	39.6	53.9	38.3	41.3	37.3	49.8	45.6	41.8	35.8	36.1	**33.4**	36.3	35.0
WResNet	53.3	53.5	50.4	52.9	55.1	35.9	49.6	35.5	38.1	33.9	44.7	43.1	39.3	32.9	34.2	**31.2**	33.2	31.9
ResNeXt	53.4	54.6	51.4	54.1	54.4	34.9	51.3	33.7	35.6	31.1	44.2	41.2	36.4	31.0	31.5	**28.8**	31.2	29.9
Mean	55.6	56.1	52.6	55.5	55.2	38.3	52.5	36.8	39.5	35.3	46.5	43.6	39.8	33.9	34.5	**31.9**	34.1	33.0

Table 1. Corruption robustness on CIFAR-10 (first 6 rows) and CIFAR-100 with various CNNs. Values show mCE, *lower is better*. The table is divided into groups for easy comparison; single-only augmentation, paired-only augmentation, combined augmentations, etc. *Orig* refers to the standard model.

	State-of-the-art methods							Single-only			Paired-only			Combined			$\mathcal{APR}_{\mathcal{P}}$[2] *with*	
	Orig	Cutout	Mixup	CutMix	AT	AugMix	AA	$\mathcal{APR}_{\mathcal{S}}$	$\mathcal{HA}_{\mathcal{S}}$	$\mathcal{HA}^{++}_{\mathcal{S}}$	$\mathcal{APR}_{\mathcal{P}}$	$\mathcal{HA}_{\mathcal{P}}$	$\mathcal{HA}^{++}_{\mathcal{P}}$	$\mathcal{APR}_{\mathcal{PS}}$	$\mathcal{HA}_{\mathcal{PS}}$	$\mathcal{HA}^{++}_{\mathcal{PS}}$	$\mathcal{HA}_{\mathcal{S}}$	$\mathcal{HA}^{++}_{\mathcal{S}}$
AllConv	93.9	93.9	93.7	93.6	81.1	93.5	93.5	93.5	94.1	93.9	**94.5**	93.9	94.0	94.3	**94.5**	94.3	94.3	94.3
DenseNet	94.2	**95.2**	94.5	94.7	82.1	95.1	95.2	94.9	94.7	95.0	95.0	93.1	93.2	**95.2**	94.9	94.8	95.1	95.1
WResNet	94.8	95.6	95.1	95.4	82.9	95.1	95.2	95.0	95.3	95.4	95.2	93.2	92.0	95.7	95.0	95.3	95.4	**95.8**
ResNeXt	95.7	95.6	95.8	96.1	84.6	95.8	96.2	95.5	95.3	95.7	95.5	93.5	92.9	**96.1**	95.2	95.9	95.6	**96.1**
Mean	94.2	95.0	94.7	94.9	82.6	94.8	95.0	94.9	94.9	95.1	95.0	92.9	92.3	95.2	95.0	95.1	95.1	**95.3**
AllConv	74.9	-	-	-	-	-	-	75.3	75.0	**75.8**	74.8	74.08	74.7	75.2	**75.8**	75.2	75.7	75.1
DenseNet	71.4	-	-	-	-	-	-	75.8	76.0	75.6	71.5	71.4	71.7	75.6	74.9	75.9	**76.1**	**76.1**
WResNet	72.1	-	-	-	-	-	-	76.2	76.8	76.2	70.4	71.3	71.7	76.8	74.8	76.0	**77.2**	76.5
ResNeXt	75.0	-	-	-	-	-	-	78.8	79.4	79.4	71.1	73.5	74.3	79.1	77.3	78.8	**79.9**	79.3
Mean	72.9	-	-	-	-	-	-	76.6	76.9	76.8	70.3	71.1	70.8	76.5	75.6	76.4	**77.1**	76.6

Table 2. Clean accuracy values on CIFAR-10 (first 6 rows) and CIFAR-100. *Higher the better*. The table is divided into groups for easy comparison; single-only augmentation, paired-only augmentation, combined augmentations, etc.*Orig* refers to the standard model.

1.2. More on $\mathcal{HA}$ and $\mathcal{HA}^{++}$

We provide the pseudo-code of $\mathcal{HA}^{++}_{\mathcal{P}}$ and $\mathcal{HA}_{\mathcal{P}}$ in Algorithm 1. Also provided is the pseudo-code for $\mathcal{HA}^{++}_{\mathcal{S}}$ and $\mathcal{HA}_{\mathcal{S}}$ in Algorithm 2. Our code and pretrained models will be made publicly available.

Note that in Algorithm 2, we decompose into low and high frequency bands both augmented images (lines 22-23 and 25-26), and also amplitude-phase swap low-frequency bands (`lfc_f` and `lfc_s`) of both augmented images (lines 42 and 56). We then randomize the selection of which low/high frequency components will come from which image for the final result (lines 58 to 63). Figure 1 of the main text shows a simplified version of this, where only the execution of line 61 is shown. In practice, we use the implementation provided in Algorithm 2.

1.3. Detailed results - transformer

We provide the detailed results of our corruption robustness experiments with Swin-Tiny [6]. The result in Table 1.3 shows that $\mathcal{HA}^{++}_{\mathcal{PS}}$ consistently improves on all types of corruptions, regardless of their frequency characteristics.

```
1  def hybrid_augment_paired(x_batch, prob, blur_fnc
       , is_ha_plus):
2      #x_batch: batch of training images
3      #prob: probability value [0,1]
4      #blur_fnc: blurring function
5      #is_ha_plus: True for HA++, false for HA
6      #fft: fourier transform
7      #ifft: inverse fourier transform
8
9      p = random.uniform(0,1)
10     if p > prob:
11         return x
12
13     batch_size = x_batch.size()[0]
14     index = torch.randperm(batch_size)
15
16     lfc = blur_fnc(x_batch)
17     hfc = x - lfc
18     hfc_mix = hfc[index]
19
20     if is_ha_plus:
21         #Based on the APR method.
22         p = random.uniform(0,1)
23         if p > 0.6:
24             lfc  = lfc
25         else:
26             index_p = torch.randperm(batch_size)
27             phase1, amp1 = fft(lfc)
28             lfc_mix = lfc[index_p]
29             phase2, amp2 = fft(lfc_mix)
30             lfc = ifft(phase1, amp2)
31
32     hybrid_ims = lfc + hfc_mix
33     return hybrid_ims
```

Listing 1. PyTorch-style pseudocode for $\mathcal{HA}_{\mathcal{P}}$ and $\mathcal{HA}^{++}_{\mathcal{P}}$.

```
1  def hybrid_augment_single(x, prob, blur_fnc,
       sample_augs, is_ha_plus):
2      #x: a single training image
3      #prob: probability value [0,1]
4      #blur_fnc: blurring function
5      #sample_augs: randomly sample augmentations
6      #is_ha_plus: True for HA++, false for HA
7      #fft, ifft: fourier and inverse fourier
           transform
8
9      p = random.uniform(0,1)
10     if p > prob:
11         return x
12
13     #First augmented view.
14     ops1 = sample_augs()
15     x_aug1 = ops1(x)
16
17     #Second augmented view.
18     ops2 = sample_augs()
19     x_aug2 = ops2(x)
20
21     lfc_f = blur_fnc(x_aug1)
22     hfc_f  = x_aug1 - lfc_f
23
24     lfc_s = blur_fnc(x_aug2)
25     hfc_s  = x_aug2 - lfc_s
26
27     if is_ha_plus:
28         #For lfc_f.
29         p = random.uniform(0, 1)
30         if p > 0.6:
31             lfc_f = lfc_f
32         else:
33             ops3 = sample_augs()
34             lfc_aug = ops3(lfc_f)
35             ops4 = sample_augs()
36             lfc_aug_2 = ops4(lfc_f)
37
38             phase1, amp1 = fft(lfc_aug)
39             phase2, amp2 = fft(lfc_aug_2)
40             lfc_f = ifft(phase1, amp2)
41
42         #For lfc_s.
43         p = random.uniform(0, 1)
44         if p > 0.6:
45             lfc_s = lfc_s
46         else:
47             ops5 = sample_augs()
48             lfc_aug = ops5(lfc_s)
49             ops6 = sample_augs()
50             lfc_aug_2 = ops6(lfc_s)
51
52             phase1, amp1 = fft(lfc_aug)
53             phase2, amp2 = fft(lfc_aug_2)
54             lfc_s = ifft(phase1, amp2)
55
56     p = random.uniform(0, 1)
57
58     if p > self.prob:
59         hybrid_im = lfc_f + hfc_s
60     else:
61         hybrid_im = lfc_s + hfc_f
62
63     return hybrid_im
```

Code 2. PyTorch-style pseudocode for $\mathcal{HA}_{\mathcal{S}}$ and $\mathcal{HA}^{++}_{\mathcal{S}}$.

		Noise			Blur				Weather				Digital				
Method	Test Error	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Brightness	Contrast	Elastic	Pixel	JPEG	mCE
Standard	**18.8**	52	54	53	68	81	65	72	57	52	47	48	45	74	61	63	59.5
$\mathcal{HA}^{++}_{\mathcal{PS}}$	19.4	**44**	**48**	**42**	**63**	**78**	**59**	**71**	**49**	**48**	**46**	**46**	**39**	**71**	**60**	**59**	**54.8**

Table 3. Swin-Tiny Clean error and corruption robustness (mCE) on ImageNet. *Lower is better.*

1.4. Related work continued

The robustness literature is vast, and it is difficult to cover all methods, therefore in the main text we opted to cover and compare ours against the most relevant ones (i.e. frequency-centric augmentations). Here, we discuss additional, more recent methods.

We focus on recent methods, such as [1, 11, 9, 8]. [1] uses an extra model to generate new training samples, which makes the method significantly more complex than ours. Despite this added complexity, we outperform it on ImageNet-C without extra data (75.03 vs 65.8 mCE) and with extra data (62.9 vs 58.9 mCE), even though they use additional augmentations (i.e. AugMix). [11] extends AugMix by making parts of the cascade augmentation pipeline learnable. We outperform it on CIFAR-10/100-C on all architectures. Note that we could not compare against them on ImageNet-C as they use a different architecture (i.e. ResNet18). [9] outperforms us on ImageNet, but it uses model ensembles during training, which are finetuned on some of the test-time corruptions of ImageNet-C (i.e. noise and blur finetuning for high-frequency model, contrast finetuning for low-frequency model). We believe this violates the assumption of not using test-time corruptions in training. PRIME [8] mixes several max-entropy transforms to augment the training distribution. We outperform it on CIFAR-10/100, are competitive on ImageNet-$\bar{C}$ but behind on ImageNet-C. Despite its results, PRIME has three key disadvantages compared to our method; it i) requires per-dataset hyperparameter tuning for its transforms, ii) manual tuning of these parameters are required to preserve semantics after augmentation and iii) shows that their augmented images look similar to test-time corruptions, which might be (inadvertently) violating the assumption of not using test-time corruptions in training.

1.5. Adversarial robustness on ImageNet

We evaluate ResNet-50 models trained with $\mathcal{HA}^{++}_{\mathcal{PS}}$, $\mathcal{APR}_{\mathcal{PS}}$ and standard training. We use the model checkpoints shown in Table 3 (main text); we do not train new models. Table 4 shows $\mathcal{HA}^{++}_{\mathcal{PS}}$ improves robust and clean accuracy (RA, CA) on ImageNet, and comfortably outperforms our baseline. Note that we use a smaller $\epsilon = 1/255$ value, as higher epsilon evaluation would require adversarial (re)training.

	Orig.	$\mathcal{APR}_{\mathcal{PS}}$	$\mathcal{HA}^{++}_{\mathcal{PS}}$
CA	76.10	75.60	**76.30**
RA	51.02	54.22	**56.44**

Table 4. AutoAttack results.

1.6. Transfer learning performance

As reported in [10], robust models tend to transfer better to downstream tasks. In the same vein, we perform a wide range of finetuning experiments, where a standard ResNet50 and $\mathcal{HA}^{++}_{\mathcal{PS}}$-trained ResNet50 are finetuned on various datasets by changing the final layer. Note that we do not train new models; we use the model checkpoints shown in Table 3 (main text). Table 5 shows we comfortably outperform standard training on majority of other classification tasks. This shows the transferability of the features learned by our augmentation schemes.

CIFAR10	CIFAR100	Aircraft	CTech101	DTD	Flowers	Pets	CTech256	Birds	Cars	SUN	Food
96.8	83.4	**86.6**	**94.0**	74.1	96.3	93.2	81.5	73.6	90.9	62.1	**87.5**
97.4	**84.9**	84.5	92.7	**75.1**	**96.8**	**93.3**	**83.0**	**73.7**	**91.0**	**63.3**	87.4

Table 5. Transfer learning acc. (top-1) of standard ResNet50 (top) and $\mathcal{HA}^{++}_{\mathcal{PS}}$ (bottom) on 12 other classification datasets.

References

[1] Dan Andrei Calian, Florian Stimberg, Olivia Wiles, Sylvestre-Alvise Rebuffi, András György, Timothy A Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. In *International Conference on Learning Representations*, 2022.

[2] Guangyao Chen, Peixi Peng, Li Ma, Jia Li, Lin Du, and Yonghong Tian. Amplitude-phase recombination: Rethinking robustness of convolutional neural networks in frequency domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 458–467, 2021.

[3] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[4] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[5] 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 preprint arXiv:1912.02781*, 2019.

[6] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

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

[8] Apostolos Modas, Rahul Rade, Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Prime: A few primitives can boost robustness to common corruptions. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 623–640, Cham, 2022. Springer Nature Switzerland.

[9] Tonmoy Saikia, Cordelia Schmid, and Thomas Brox. Improving robustness against common corruptions with frequency biased models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10211–10220, October 2021.

[10] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

[11] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[12] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[13] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.