Do ImageNet-trained models learn shortcuts? The impact of frequency shortcuts on generalization

Supplementary Material

1. HFSS configurations

This section details the HFSS configurations on C-10, designed to analyze the trade-off between efficiency and effectiveness in identifying shortcuts. We discuss the rationale behind the choice of frequency patch size and the percentage of sampling frequencies. We then present tracking statistics of the lowest loss for an ImageNet model as frequency subsets are incrementally sampled for shortcut evaluation, which guides the selection of B_s for the HFSS configuration on IN-1k.

1.1. Efficiency and effectiveness of HFSS

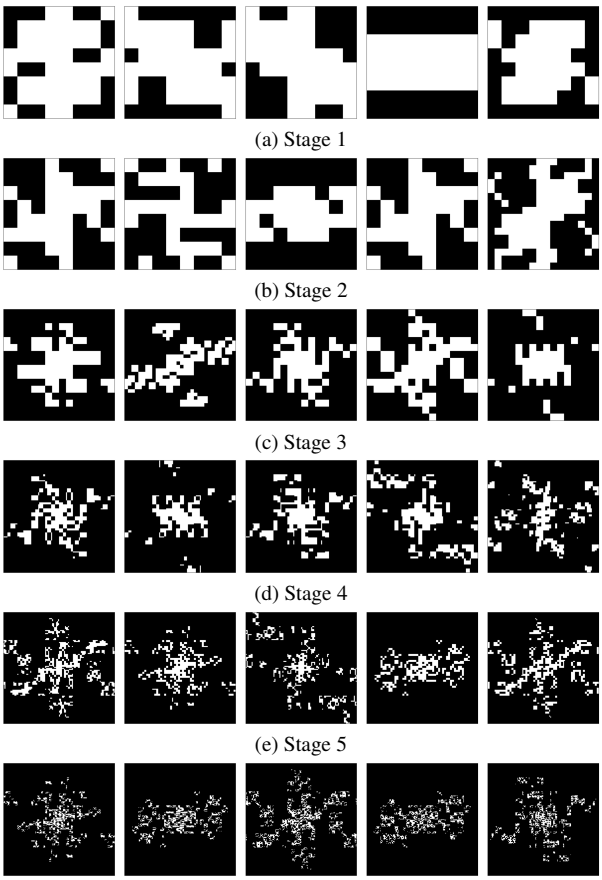
We perform 10 experiments on C-10, each with different number of B_s as shown in Tab. 1. The initial configuration is noted as CF-1 which generates in total 15000 candidate frequency subsets for shortcut relevance evaluation. The fastest (last) configuration is noted as CF-2.10, which only generates in total 70 candidate frequency subsets for evaluation. From the results in the paper, with CF-2.10 HFSS manages to uncover most shortcuts found by CF-1 at low thresholds. There exists a trade-off between the efficiency and effectiveness of HFSS in finding shortcuts.

Table 1. Experiment configurations on C-10.

	No. of sampled candidates				Total
CF-	B_1	B_2	B_3	B_4	
1	1000	2000	4000	8000	15000
2.1	200	800	4000	4000	9000
2.2	200	800	2000	2000	5000
2.3	200	800	500	500	2000
2.4	200	400	500	500	1600
2.5	200	200	500	500	1400
2.6	200	200	300	300	1000
2.7	100	100	200	200	600
2.8	50	50	100	100	300
2.9	20	20	50	50	140
2.10	10	10	25	25	70

1.2. Patch size selection across stages

In the first stage, we design the patch size to ensure the image spectrum can be evenly separated in 4×4 patches. This split results in a manageable number of combinations of frequency subsets, facilitating an effective initial coarse exploration of frequencies that contribute significantly to classification. From the second stage onward, the patch size is halved compared to that of the previous stage. This progressive refinement improves the precision of the frequency subsets explored that contain shortcut information. Examples of sampled frequency subsets are shown in Fig. 1. The frequency patches progressively decrease in size and the frequency maps become more refined as the search progresses.



(a) Stage 1

(b) Stage 2

(c) Stage 3

(d) Stage 4

(e) Stage 5

(f) Stage 6

Figure 1. Sampled frequency subsets at each stage.

1.3. Frequency sampling percentage

At each stage, we sample 60% of the frequency patches. This uniform sampling ratio allows for investigating frequency shortcuts formed by different percentage of frequencies. For instance, applying DFMs searched at stage 3 allows us to analyze the impact of potential shortcuts that contain approximately 22% of the frequencies on OOD data. In

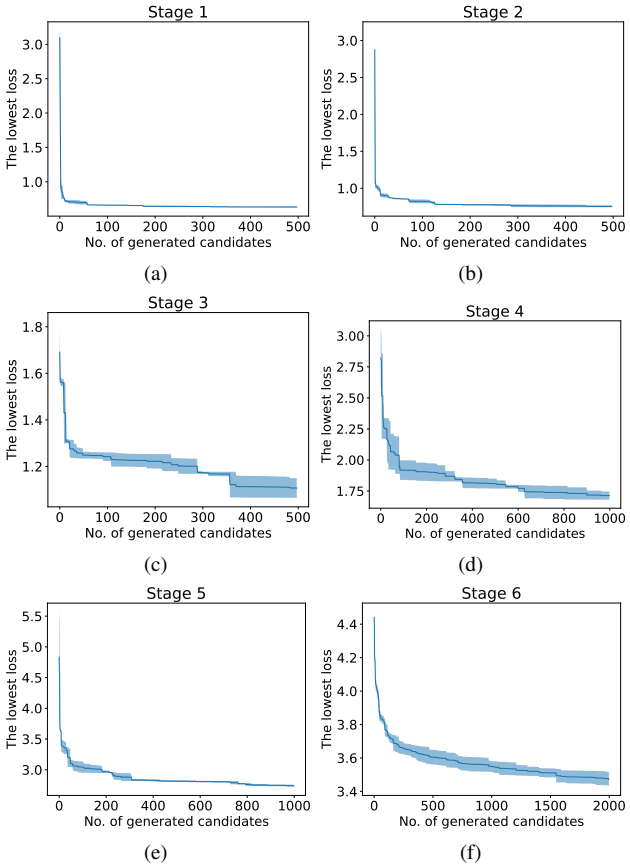


Figure 2. Tracking statistics of IN-1k configuration of ResNet18.

subsequent stages, the resulting DFMs indicate around 13% of the frequencies across full spectrum in stage 4, and about 8% in stage 5.

1.4. Configurations on IN-1k

As IN-1k has 1000 classes, we increase the number of sampled candidates in each stage. We set B_1=B_2=B_3=500, B_4=B_5=1000 and B_6=2000. We run this configuration three times with random seeds 42, 125 and 666. We track the lowest loss (averaged over all 1000 classes), calculating the mean and standard deviation over the three trials. We show the tracking statistics of ResNet18 in Fig. 2. For stages 1 and 2, sampling 500 frequency subsets are sufficient as the loss does not decrease significantly as the number of sampled candidates increases. Starting from stage 3, although the standard deviations are relatively higher than the previous stages, the lowest loss does not decrease much after sampling more frequency subsets. Considering time-efficiency (around 9 days to run HFSS once), we use this setup for all ImageNet experiments, with slightly reduced effectiveness in finding strong shortcuts.

2. Experiment setups

2.1. Datasets

ImageNet-v2 (IN-v2) [5]. This dataset has the same structure as ImageNet-1k, containing 1000 classes. The data creation process of IN-v2 is the same as that of IN-1k. This can evaluate model performance on images collected in different time points, i.e. generalization to statistical distribution shifts.

ImageNet-C (IN-C) [3]. It contains 19 types of synthetic corruption effect, which are Gaussian noise, impulse noise, shot noise, defocus blur, glass blur, motion blur, zoom blur, brightness, contrast, elastic transform, jpeg compression, pixelate, fog, frost, and snow. The dataset contains 19 subsets, each containing IN-1k test images corrupted by one type of corruption, with five levels of corruption severity. High severity indicates high strength of corruption applied to the original test images.

ImageNet-R (IN-R) [4]. It contains images with different renditions, such as cartoon, art, graphics, painting, etc. These allow for a strong model generalizability assessment, as some abstract renderings exclude important features like natural textures that models rely on for classification.

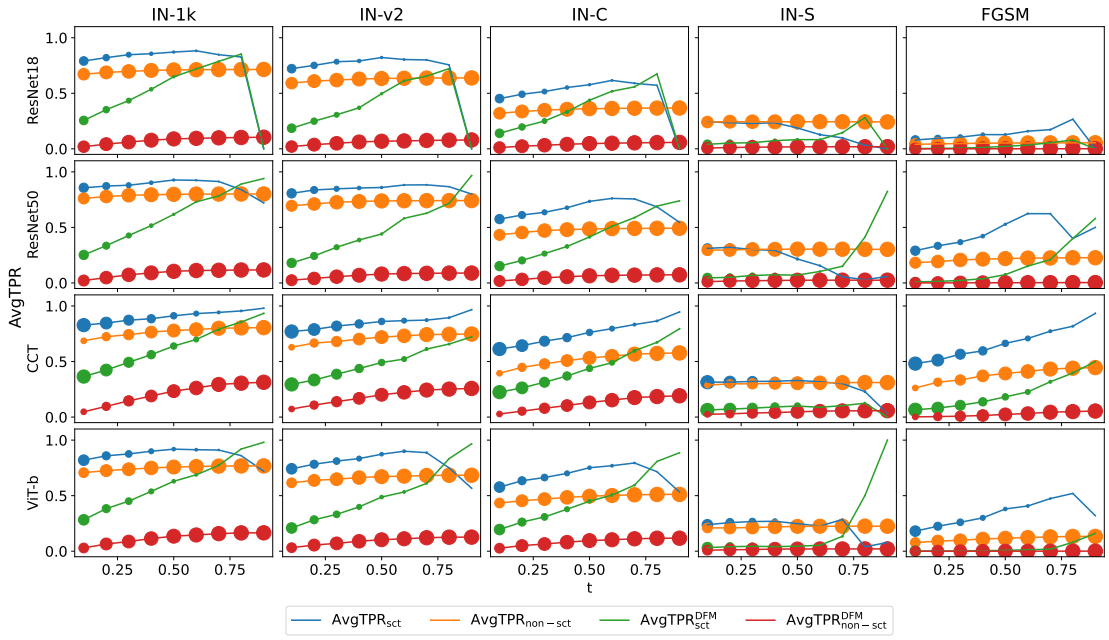
ImageNet-S (IN-S) [6]. The dataset contains 1000 classes, each with 50 validation images, the same as IN-1k. Differently, the images are sketches of objects, which may have texture information loss.

ImageNet-SCT (IN-SCT) [7]. This OOD dataset is constructed to evaluate the impact of frequency shortcuts on generalization performance. It contains 10 classes, sharing similar shape or texture characteristics to the 10 classes in IN-10. Each class has 70 images with seven renditions, e.g. cartoon, painting, sketch, etc.

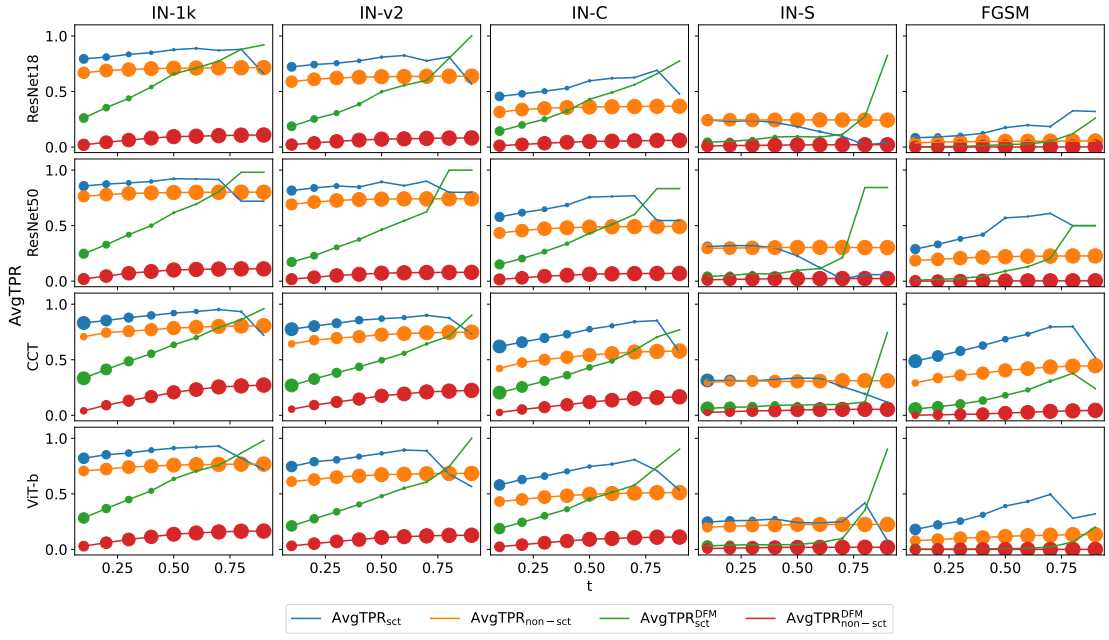
2.2. Training

C-10. Models with ResNet [2] architecture are trained for 200 epochs on the C-10 dataset. The initial learning rate is 0.01, reduced by a factor of 10 if the validation loss does not decrease for 10 epochs. We use SGD optimizer with momentum 0.9 and weight decay 10^{-4} and batch size 128.

IN-10. Models with ResNet(s) [2] architectures are trained for 200 epochs. The initial learning rate is 0.01 and is reduced by a factor of 10 if the validation loss does not decrease for 10 epochs. We use SGD optimizer with momentum 0.9 and weight decay 10^{-4} and batch size 16.



(a)



(b)

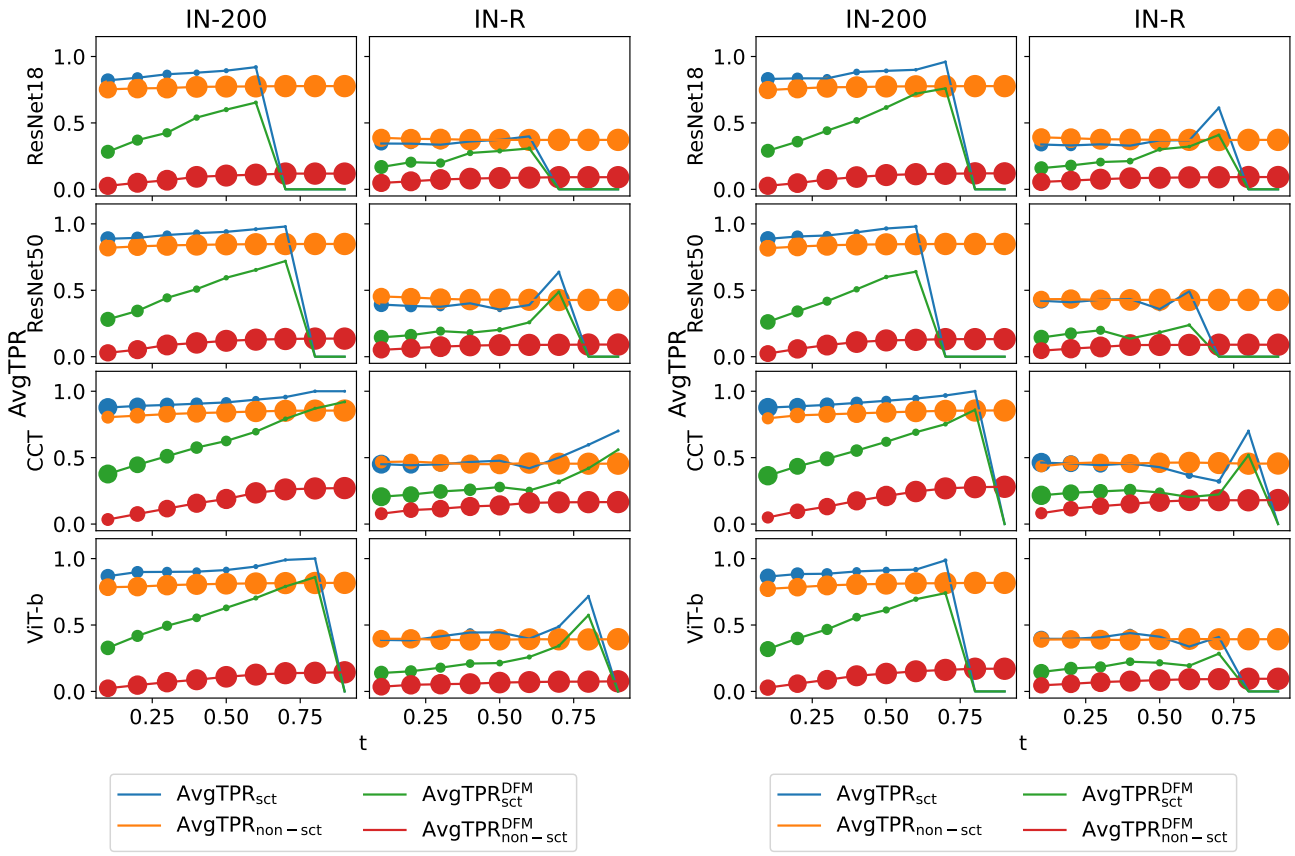
Figure 3. Impact of shortcuts uncovered in (a) the second and (b) the third run on OOD data: average TPR of shortcut and non-shortcut classes given different thresholds. In general, models perform better on images of shortcut classes than non-shortcut classes.

IN-1k. We use the pre-trained weights of ResNet18, ResNet50 and ViT-b from timm [8] and the weights of CCT from the official repository [1].

3. Additional results

3.1. IN-1k

Results across multiple runs. We conducted ImageNet experiments for each model three times, with results from the additional two trials presented in Figs. 3 and 4.



(a) (b)

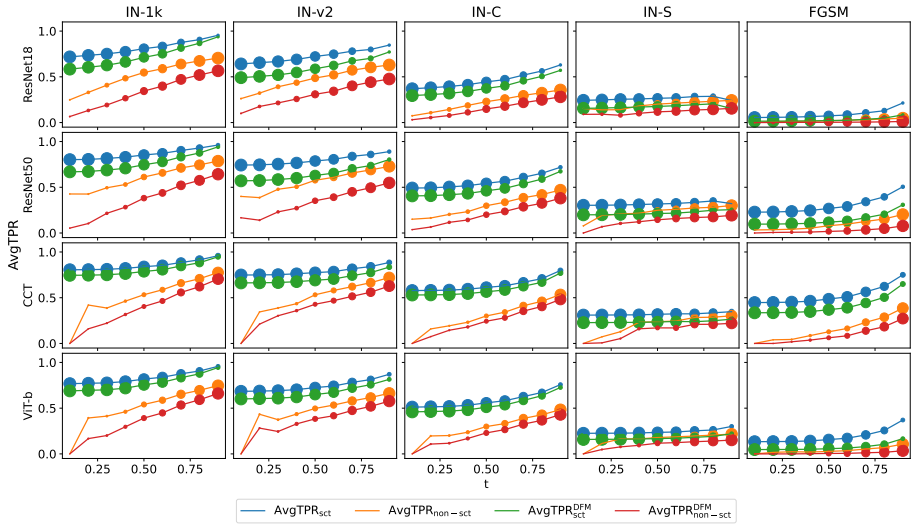
Figure 4. Impact of shortcuts uncovered in (a) the second and (b) the third run on IN-R: average TPR of shortcut and non-shortcut classes given different thresholds. In general, models perform worse on images of shortcut classes than non-shortcut classes.

Similar to the findings from the first run, the subsequent trials also manage to identify classes influenced by shortcuts. Models consistently perform better on shortcut-classes than non-shortcut classes across datasets such as IN-1k, IN-v2, IN-C and under FGSM attacks but worse on IN-R. These results align with the observations that models excel on texture-preserved datasets, as they exploit the shortcuts present in OOD data. Notably, CCT shows the strongest tendency toward shortcut learning among the evaluated models.

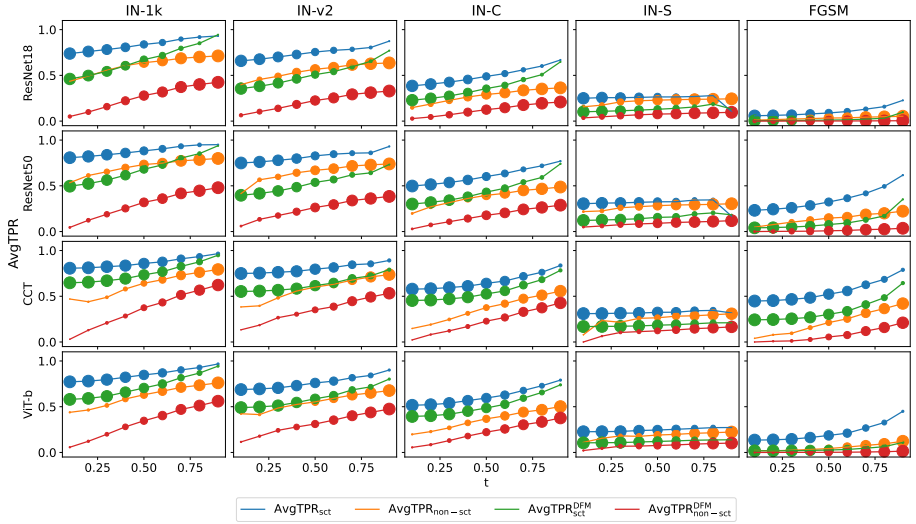
Although the current HFSS configuration applied to ImageNet might overlook some strong shortcuts (see the green lines of ResNet18 in Fig. 3a), HFSS stably uncovers shortcuts at low thresholds (weak shortcuts). As our focus is on the general impact of shortcuts on generalization and robustness, rather than precise prediction performance on specific classes, the configuration of HFSS provides analyzable results for such investigation. For more detailed analyses, one could increase the number of sampling operations B_s, allowing a broader evaluation of frequency subset combinations and obtaining more stable search of strong shortcuts.

Results at different stages. We analyze the impact of frequency shortcuts searched in different stages, which correspond to different percentage of frequencies, on the ID and OOD test sets, as shown in Figs. 5 and 6. Sampling 60% frequency patches at each stage results in around 22% of frequencies at stage 3, 13% at stage 4 and 8% at stage 5 of the full image spectrum.

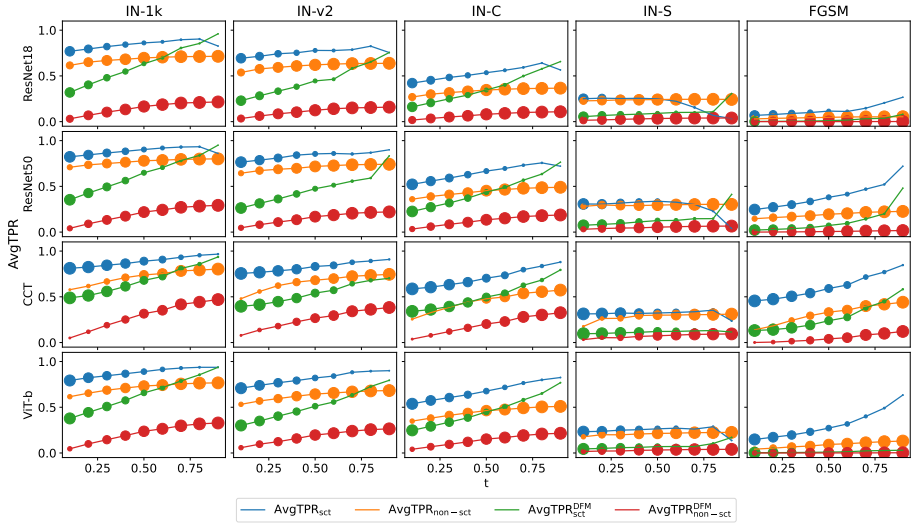
In Figs. 5a and 6a, we present the average TPR of models using DFMs obtained at stage 3 (approximately 22% of frequencies). For shortcut classes, the models achieve performance comparable to that on full-spectrum images. But for non-shortcut classes, performance is generally worse. On IN-S and IN-R datasets, models perform slightly better on shortcut classes than non-shortcut classes at low thresholds. We inspect images filtered by the DFMs from stage 3 and see that retaining only 22% of frequencies results in minimal visual differences compared to the original images, aside from some artifacts caused by filtering. As this retained information is sufficient for classification, most classes are considered subject to shortcuts at low thresholds. This explains the slightly higher Average TPR values, par-



(a)

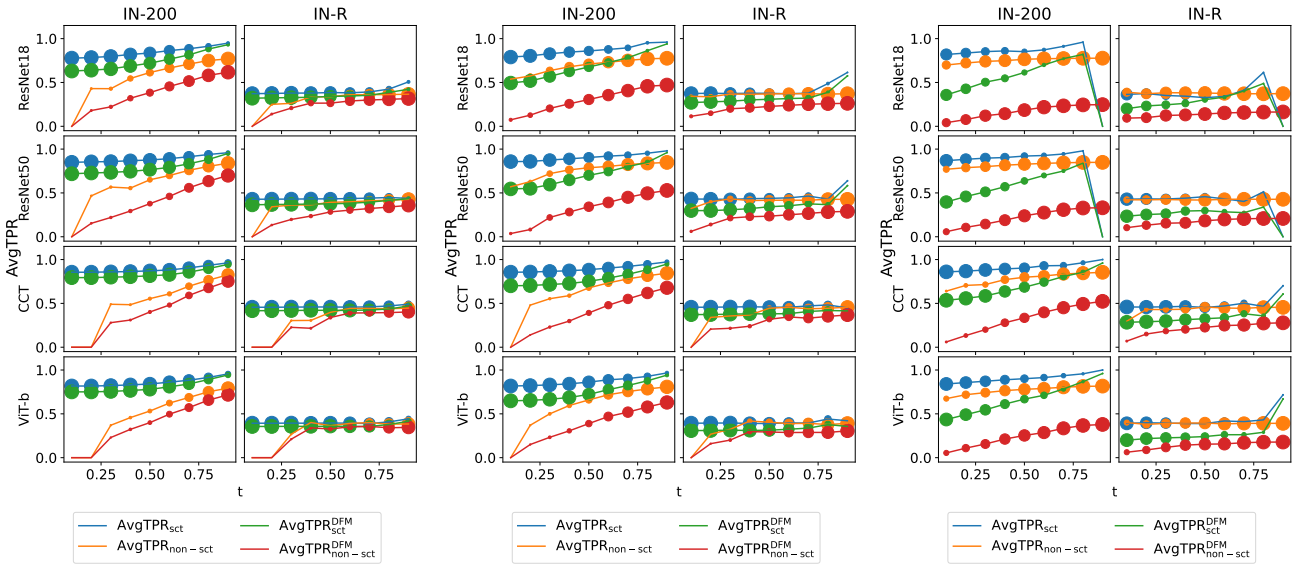


(b)



(c)

Figure 5. Average TPR of shortcut and non-shortcut classes given different thresholds, using DFMs containing around (a) 22%, (b) 13% and (c) 8% of frequencies. In general, models perform better on images of shortcut classes than non-shortcut classes.



(a) (b) (c)

Figure 6. Average TPR of shortcut and non-shortcut classes given different thresholds on IN-R, using DFMs containing around (a) 22%, (b) 13% and (c) 8% of frequencies. In general, models perform similarly on images of shortcut classes and non-shortcut classes.

ticularly for AvgTPR@0.1 in CCT (see Fig. 5a), where almost no non-shortcut classes remain. Despite this, the performance decline of shortcut classes from IN-1k to In-S and IN-R is notably more pronounced compared to non-shortcut classes, showing that reliance on frequency shortcuts does not aid model generalization. The larger performance drops on IN-S (compared to the drop on IN-R) can also be attributed to shortcuts such as color-related cues, as IN-S only contains black-and-white sketches. Similar trends are observed for stages 4 and 5, as shown in Figs. 5 and 6.

Number of classes per stage. Tab. 2 presents the number of shortcut classes at each stage. As the threshold t increases, the count of shortcut classes declines. Notably, at stage 6 with $t = 0.9$, the number drops to 1-3, indicating that such strong shortcuts are uncommon.

3.2. C-10

We report the test results of ResNet models trained on C-10 in Tab. 3. All models trained on C-10 learn shortcuts to classify images in classes *airplane* and *bird*. Based on the threshold value of TPR (0.6), ResNet34 and ResNet50 are less subject to frequency shortcuts, although they still learn them, indicating that larger model capacity is not sufficient to avoid shortcut learning, in line with [7].

The OOD test results of C-10 models are provided in Tab. 4. Models exhibit close-to or above-average TPR for classes *airliner* and *humming bird*, which is attributable to the presence of shortcuts in the OOD data.

Table 2. The number of shortcut classes per stage in IN-1k.

Stage \ t	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ResNet18									
3	993	950	894	792	642	516	317	175	40
4	922	826	701	539	381	276	133	62	9
5	647	445	301	203	111	61	23	12	3
6	343	196	116	60	22	16	7	3	1
ResNet50									
3	997	990	956	904	782	673	469	305	82
4	976	903	792	639	457	339	175	95	12
5	809	604	445	308	174	110	45	18	2
6	421	253	149	78	33	18	5	1	1
CCT									
3	1000	997	989	958	890	820	648	475	166
4	998	985	948	885	755	636	421	256	72
5	973	906	778	622	442	336	152	82	15
6	781	566	408	277	142	88	34	15	3
ViT-b									
3	1000	989	977	930	820	720	518	343	101
4	987	962	900	772	633	491	297	171	33
5	861	674	514	368	225	149	70	25	6
6	520	317	183	110	54	30	7	1	1

Table 3. TPRs on C-10 and DFM-filtered images. TPR≥0.6 are highlighted in bold.

Model \ Class	airplane	auto	bird	cat	deer	dog	frog	horse	ship	truck
ResNet18	**0.709**	0.581	**0.86**	0.585	**0.679**	0.482	**0.713**	0.395	**0.675**	**0.682**
ResNet34	**0.988**	0.183	**0.935**	**0.641**	0.585	0.538	0.393	0	0.203	0.467
ResNet50	**0.995**	0.365	**0.858**	0.526	0.361	0.432	0.251	0.259	0.338	0.526

3.3. IN-10

We provide the comparison of the results of ResNet50 using DFMs searched by HFSS and the algorithm in [7] in Tab. 5. ResNet50 learns strong shortcuts for classes *airliner*, *siamese cat*, *ox*, *zebra* and *container ship*. Although it has larger model capacity than ResNet18, HFSS confirms that it still exploits shortcuts for many classes, in line with

Table 4. TPRs of C-10 models tested on resized IN-10 (first row of each model) and corresponding DFM-filtered images (second row of each model). TPRs higher than or close to average TPR (ResNet18-0.62, ResNet34-0.62 and ResNet50-0.64) are highlighted in bold.

Method \ Class	airliner	wagon	hum-bird	Siam-cat	ox	golden retriever	frog	zebra	Con-ship	truck
						ResNet18				
–	**0.96**	**0.7**	**0.62**	**0.7**	0.24	**0.74**	**0.72**	0.34	**0.82**	0.4
HFSS	**1**	0.14	**0.78**	**0.64**	0.14	0.22	0.18	0.1	0.32	0.14
						ResNet34				
–	**0.96**	**0.72**	**0.64**	**0.82**	0.24	**0.66**	**0.68**	0.2	**0.84**	0.48
HFSS	**1**	0.08	**0.9**	**0.68**	0.44	0.46	0.28	0	0.18	0.32
						ResNet50				
–	**0.96**	**0.76**	**0.6**	**0.78**	0.28	**0.74**	**0.74**	0.22	**0.84**	0.44
HFSS	**0.98**	0.28	**0.86**	0.48	0.26	0.26	0.32	0.22	0.28	0.32

Table 5. TPR results of ResNet50 on DFM-filtered IN-10 images. TPR$\geq$0.6 (a strong frequency shortcut) is highlighted in bold.

Method \ Class	airliner	wagon	hum-bird	Siam-cat	ox	golden retriever	frog	zebra	Con-ship	truck
HFSS	**1**	0.06	0.3	**0.94**	**0.76**	0.46	0.46	**0.78**	**0.82**	0.08
[7]	0.54	0	0	0.42	0	0.2	0	0.16	**0.7**	0.1

Table 6. Models trained on IN-10 are tested on IN-SCT. TPRs higher than or close to average TPR (0.374) are highlighted in bold.

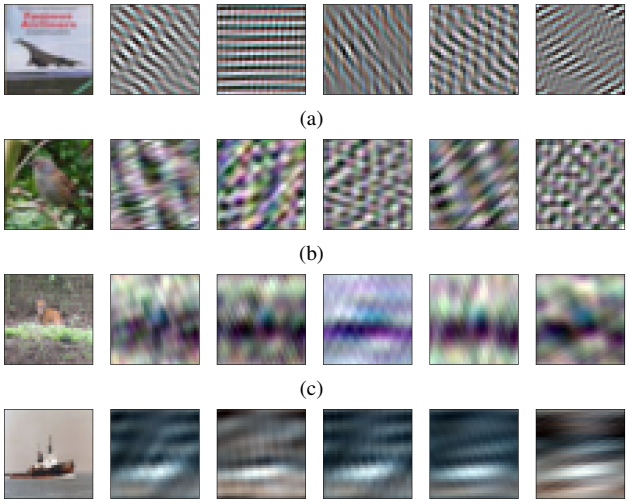
Method \ Class	Mil-aircraft	car	lorikeet	tabby cat	holstein	Lab-retriever	tree frog	horse	fishing vessel	fire truck
–	**0.429**	**0.486**	**0.414**	0.2	**0.37**	0.3	0.3	0.057	**0.44**	**0.743**
HFSS	0.257	**0.514**	**0.5**	0.3	**0.372**	**0.386**	**0.371**	0	**0.486**	**0.7**
[7]	0.243	0	0.057	0.043	0	0.2	0	0	**0.486**	0.043

the observation in [7]. By comparing the TPR values on IN-10 images processed by DFMs obtained through our HFSS algorithm and through that in [7], our algorithm is more effective at finding shortcuts (more TPRs are highlighted in bold).

We report the TPRs of ResNet50 tested on IN-SCT in Tab. 6. The model achieves higher or close to average TPR of classes *holstein* and *fishing vessel* in IN-SCT, which is attributable to the shortcuts for classifying classes *ox* and *container ship* in IN-10. Comparing the TPR values of HFSS and [7], we observe that weak shortcuts for some classes e.g. *frog* and *golden retriever* are still present in the OOD data, but [7] fails to recognize them, demonstrating the effectiveness of HFSS in finding shortcuts.

3.4. Visualization of images filtered by DFMs obtained over five trials

Due to random sampling of candidate frequency subsets, the outcomes of HFSS might deviate slightly for each run. However, from the visualization of the image filtered by DFM obtained over five trials, we observe similar texture shortcuts (see Fig. 7). This indicates that frequency shortcuts are not formed by a fixed set of frequencies, but correspond to similar spatial patterns.



(a) (b) (c) (d)

Figure 7. Images of classes (a) *airplane*, (b) *bird*, (c) *deer* and (d) ship in C-10 filtered by corresponding DFM obtained through five trials. We normalize the images to a range of 0 to 1 for visualization purpose.

References

[1] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. 2021. 3

[2] 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. 2

[3] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 2

[4] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8340–8349, 2021. 2

[5] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019. 2

[6] 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*, pages 10506–10518, 2019. 2

[7] Shunxin Wang, Raymond Veldhuis, Christoph Brune, and Nicola Strisciuglio. What do neural networks learn in image classification? a frequency shortcut perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1433–1442, 2023. 2, 6, 7

[8] Ross Wightman. Pytorch image models (timm). 3