Can Biases in ImageNet Models Explain Generalization?

Supplementary Material

Overview

A. Broader Impact

This study underscores the critical importance of conducting experiments rigorously to derive valid conclusions. By addressing the limitations in prior research and cautioning against overreliance on bias measures without comprehensive validation, this work highlights the necessity of robust experimental methodologies to advance our understanding of neural network generalization. It is well understood that correlation does not prove causality. While this study mainly challenges established correlations for a more nuanced understanding of the influences of biases on generalization, it also presents new ones (*e.g.*, high-frequency bias) - it is important that future works equally rigorously evaluate our findings and try to break the causality.

Transferability and Domain-Specific Considerations. This study explores biases in the ImageNet classification problem. While this dataset and problem are representative of a significant portion of computer vision research, our (and previous) findings may be limited in their transferability to other problems. For example, while there is a theoretical grounding for shape bias in object recognition, it is not intuitively clear if this applies to medical classification tasks such as melanoma detection. We ask researchers to exercise caution when extrapolating findings from specific contexts to broader applications and encourage a rigorous evaluation of model performance on their specific problem - in particular, in safety-critical domains where human lives are at stake.

B. Tables of Results

Table 1 contains an overview of all our models with detailed performance on every benchmark (*i.e.*, all datasets and the adversarial attack). It also serves as a legend for the markers in all our scatter plots (main paper and Appendix). Table 2 contains the corresponding measurements of our studied biases.

C. Details about Generalization Benchmarks

In distribution (ID).

- The *ImageNet (IN)* [45] validation set is the standard test dataset, containing 50,000 images with 1,000 different classes.
- *ImageNet v2 (IN-v2)* [43] is a newer 10,000 images test set sampled a decade later following the methodology of the original curation routine.
- *ImageNet-ReaL (IN-ReaL)* [2] is a re-annotated version of the original ImageNet validation set. It assigns multiple labels per image and contains multiple corrections of the original annotations.

Robustness.

- *ImageNet-C (IN-C)* [21] is a dataset consisting of 19 synthetic corruptions applied to the original ImageNet validation set under increasing severity. The original protocol suggests averaging the error over all corruptions and severities normalized by the error of AlexNet [30] on the same. Contrary, we simply report the mean accuracy over all 19 corruptions and severity levels[2].
- *ImageNet-$\bar{C}$ (IN-$\bar{C}$)* [36] extends IN-C by adding 10 new corruptions which were chosen to be perceptually dissimilar but conceptually similar to the corruptions in IN-C. We report the mean accuracy akin to IN-C.
- *ImageNet-A (IN-A)* [24] contains 7,500 additional images belonging to 200 ImageNet classes that are naturally hard to classify for ImageNet models and are, thus, posing natural adversarial examples.

Concepts.

- *ImageNet-Renditions (IN-R)* [23] is a dataset of 30,000 images of 200 different ImageNet classes in different styles, such as cartoons, paintings, toys, etc.
- *ImageNet-Sketch (IN-S)* [60] is a dataset of over 50,000 hand-drawn sketches belonging to all ImageNet classes. Semantically it can be seen as a subset of IN-R.

[2] The results remain comparable by a simple linear transformation

- *Stylized ImageNet (SIN)* [17] contains ImageNet validation images that have been stylized using different artistic filters to destroy texture information. We use the official 16-class subset given in [18].

Adversarial Robustness. We use a Project Gradient Descent (PGD) [35] attack to adaptively evaluate robustness. As models trained without adversarial training [35] are highly susceptible to such attacks, we attack with a reduced budget of $\epsilon = 0.5/255$ under ℓ_∞ norm, using 40-steps with $\alpha = 2/255$. This benchmark is an important data point due to its adaptive nature. While in theory, a model may overfit a static dataset due to a finite number of test samples, (ideal) adaptive benchmarks would not affected.

D. Results on ViT

In line with our results on ResNet-50, we provide results on ViT-B/16 [11] in Fig. 8. Unlike most ViTs, these models are exclusively trained on the ILSVRC2012 subset of ImageNet. The models originate from AugReg [50], Masked Autoencoders (MAE) [20], DINO [6], data-efficient image transformers (DeiT) [57], and sharpness-aware minimizers (SAM) [9].

E. Detailed Plots for the Analysis

In Sec. 5.1 we discuss the correlation between the performance on IN-A and the shape bias. We provide the plot for this in Fig. 9. As discussed, only strong texture-biased models show improvements in IN-A performance.

Additionally, we discuss in Sec. 5.2 the non-causal - and in particular non-functional/non-injective - relationship between high-frequency bias and generalization for AT models. In Fig. 10 we show the same results as in the main paper (Fig. 5) but limited to AT models to show the trend more clearly. Also, note how there is also a non-functional/non-injective relationship to the attack budget ϵ (corresponding to the marker size).

We also discuss the statistical correlation between our benchmarks in the main paper (Sec. 4). In Fig. 15 we additionally provide scatter plots between all benchmark pairs. We also show the distribution of reached accuracy on all benchmarks in Fig. 11, and provide an overview of the statistical correlation between our biases in Fig. 12 which we use to discuss the relationship between shape bias and spectral biases in Sec. 6.

F. Changes to the Critical Band Test

A cornerstone of the initial study [52] is the evaluation of classification accuracy on noise-modified subsets of ImageNet. Each subset contains on average only 30 random ImageNet samples that are gray-scaled, contrast-reduced, and introduced to Gaussian noise at specified frequencies and varying intensities. In contrast, we use all 50,000 ImageNet samples for each subset (and apply the same transformations). This way, we get less noisy results by testing more samples and making sure that all subsets contain the exact same images. Then, the original test measures the accuracy against 16 super-classes for each subset. This was mainly done for comparability reasons for the same study with the human subjects. Since we do not compare to human trials,

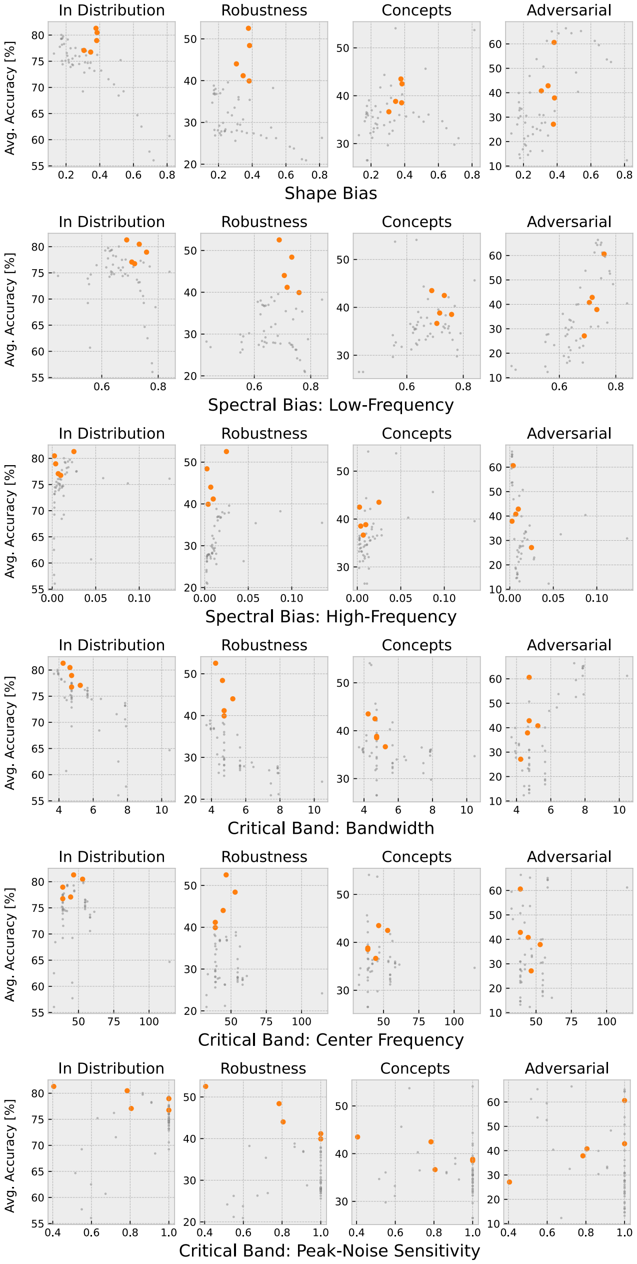


Figure 8. Biases vs. Generalization on ViT-B/16-based models. Orange markers represent the ViTs, gray ones are ResNet-50s.

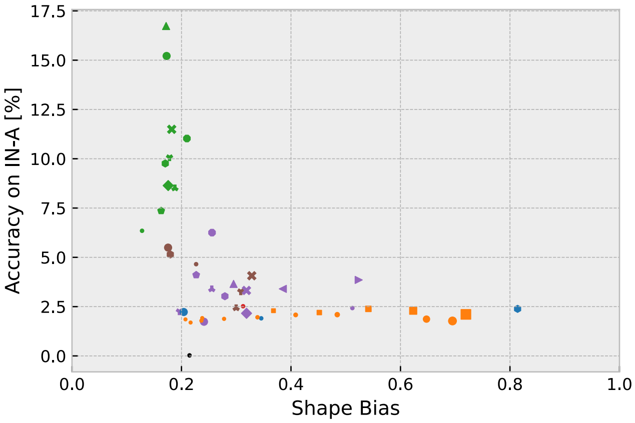


Figure 9. **Shape Bias vs. IN-A.** Only strongly texture-biased models show significant improvements but the most texture-biased model is not the best IN-A model. Markers indicate models as described by the legend in Tab. 1.

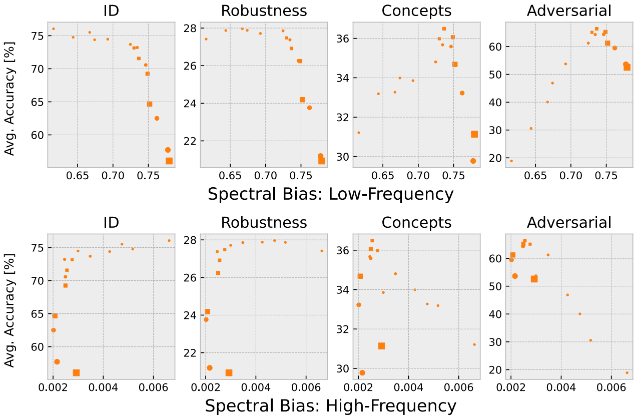


Figure 10. **Spectral Biases vs. Generalization only on adversarially-trained (AT) models.** Markers indicate models as described by the legend in Tab. 1. Marker size correlates with the attack budget ϵ during training.

we measure the common top-1 accuracy against all 1,000 classes. Finally, the critical band is measured by a fitted Gaussian function to the resulting heatmaps. The authors binarized the heatmaps by performance with a threshold of 50%. As we discussed in Sec. 3 this is not ideal for models that are contrast-sensitive (*e.g.*, AT models) and does not allow evaluation of such models. Thus, we normalize the heatmap by the maximum accuracy over all tests prior to the curve fitting. Under our methodology, this corresponds to normalization with the accuracy on the contrast-reduced images (but not under the original test where the random samples for each subset introduce noise).

For completeness, we have also evaluated our model zoo with the original method and the results are shown in Fig. 16. Additionally, to the exact same evaluation (Fig. 16a) we also apply normalization (Fig. 16b). Under the original condition, we see no reasonable correlation for any bias except when limiting the study to AT models. For the normalized study, we again see no correlations between any bias for all models, except for ID and Robustness which show some correlation to the center frequency. However, the correlation is mostly determined by the tail of AT models - removing these models would break the correlation and, thus, make a causal connection highly unlikely.

In Fig. 16c we show the scatter plots between IN-1k and IN-16 obtained results and do not see a correlation indicat-

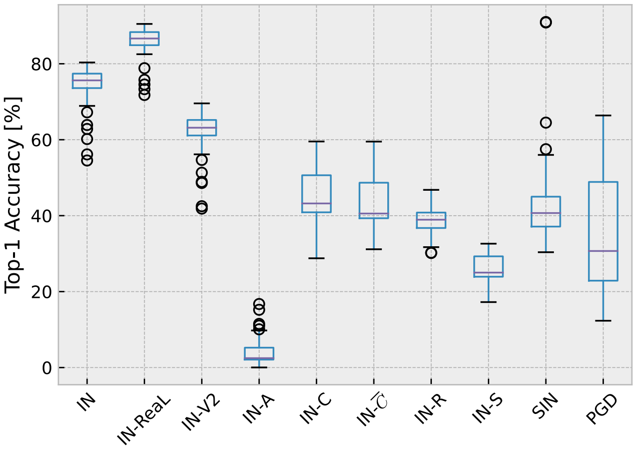


Figure 11. **Performance distribution** of the model zoo on all individual benchmarks.

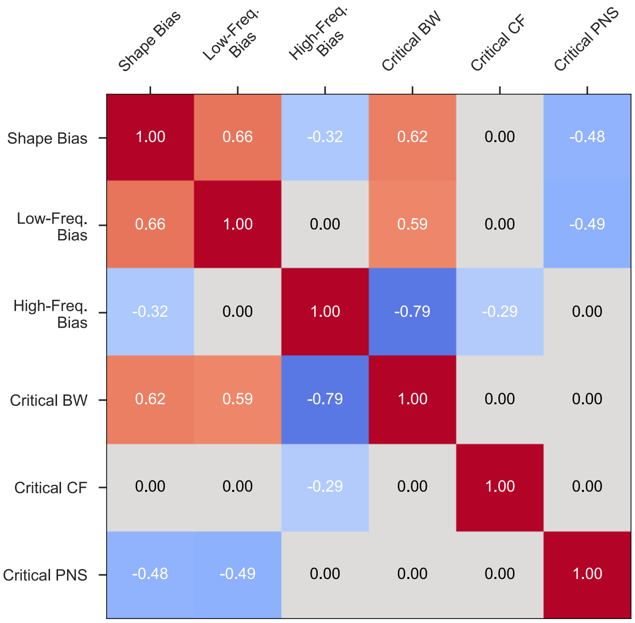


Figure 12. **Correlations between biases.** Correlations measured by Spearman r. We set non-significant correlations with $p \geq 0.05$ to 0.

ing that the results obtained by our method deviate from the original test. While our modifications may not be perfect either (*e.g.*, both our and Subramanian et al. [52] arbitrarily pick 50% as threshold) our modifications are theoretically grounded and, thus, introduce an improved measurement of the critical band for models.

G. General Observations on the Low/High-Pass Data

In this section, we want to provide some high-level findings on our low/high-pass data test (based on Fig. 13). Contrastive learning models underperformed the baseline in low-frequency bias but performed on par for high-frequency bias. We cannot prove this to be indicative of a shortcoming of contrastive learning, as we primarily benchmark older techniques that perform worse than supervised learning because newer methods are almost exclusively designed for ViTs. Still, this may deserve some attention in future works. Some augmentation techniques lead to an unreasonably strong high-frequency bias. This frequency band contains limited information and is almost imperceivable to humans without normalization (Fig. 7). Nonetheless, these models seem to be able to classify a non-negligible amount of samples. This may be related to frequency shortcuts [62] and, in fact, be less desirable. On the other hand, we also see that augmentation improves low-frequency bias and overall the strongest performance there is achieved by an augmentation model (DEEPAUGMENT + AUGMIX [23]). This training category also contains the most models that significantly deviate from the otherwise prevalent low-frequency bias. Newer training recipes seem to mostly improve high-frequency biases, without significant changes to the low-frequency bias. SIN-only training reduces low-frequency bias but significantly raises performance in mid-bands and sometimes even outperforms augmentation on some specific cutoffs. Notably, all models make significant improvements on the lowest 1% of the spectrum, with training recipes showing the least and AT the largest leaps. Gains from additional high-frequency information saturate much earlier for almost all models.

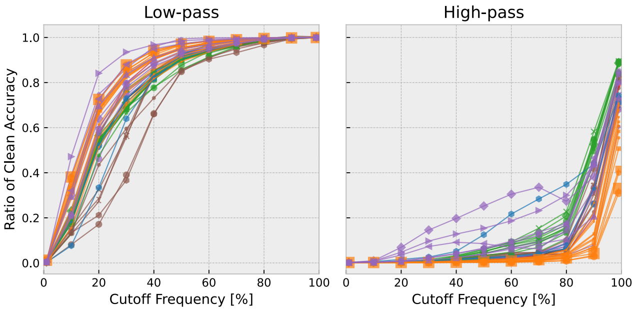


Figure 13. Frequency band-pass test on ImageNet accuracy using low-pass (left) and high-pass (right) filters with increasing cutoff frequency. The distance to the original image decreases with increasing cutoff.

H. Implementation Details

All evaluations were performed with *Python* 3.10.12, *PyTorch* 2.0.1, *CUDA* 11.7, and *cuDNN* 8500 on 4x *NVIDIA A100-SXM4* GPUs.

H.1. Data Preprocessing Pipeline

We use the same data processing pipeline for all models to ensure a fair comparison. We resize the smaller edge of the inputs to 256 px and the other edge with the same ratio using bilinear interpolation, then center-crop to 224 × 224 px. Channel-wise normalization is applied as done during training - typically, this is the mean and std over all samples of the ImageNet dataset.

The samples in IN-C̄/C are preprocessed by default, there we skip the the resizing and cropping. We want to point out that some prior also apply the above transformations to these datasets. As discussed in Sec. 3 this is questionable because it results in undersampling, and thus loss of details, and inconsistent evaluation compared to the clean ImageNet dataset and approaches using "correct" preprocessing.

H.2. Implementation of the Frequency Filter

Let $\mathcal{F}$ denote the Fast-Fourier Transformation (including shifting the zero-frequency component to the center) and $\mathcal{F}^{-1}$ the inverse operation, then we obtain the frequency filtered sample X' from an input sample X as follows:

$$X' = \mathcal{F}^{-1}(\mathcal{F}(X) \circ M_f) \tag{1}$$

M_c denotes the frequency mask (filter) in the Fourier space parameterized by the cutoff frequency f implemented as follows:

```
h, w = X.shape[-2:]
cy, cx = h // 2, w // 2
ry = int(cutoff_freq * cy)
rx = int(cutoff_freq * cx)
if lowpass:
    mask = torch.zeros_like(X)
    mask[:, cy-ry:cy+ry, cx-rx:cx+rx] = 1
else:
    mask = torch.ones_like(X)
    mask[:, cy-ry:cy+ry, cx-rx:cx+rx] = 0
```

Listing 1. Frequency Mask Computation

An example of the resulting samples can be found in Fig. 14.

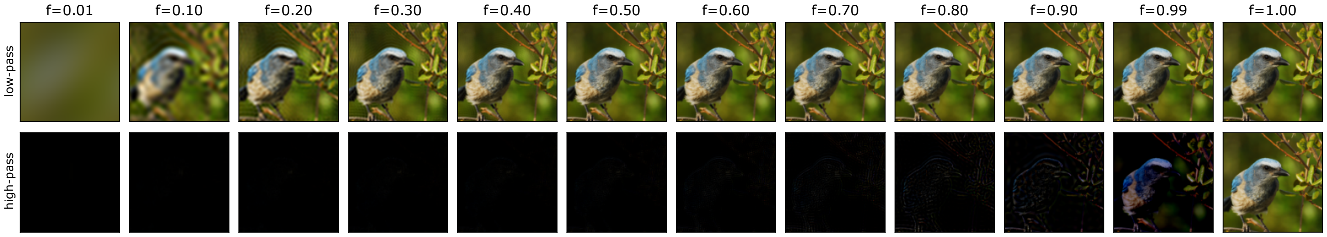


Figure 14. Visualization of the low/high-pass filtered data at cutoff frequency f on one ImageNet sample.

Table 1. Performance of ResNet-50 models on our generalization benchmarks.

			Top-1 Test Accuracy [%] (↑)									
			In Distribution			*Robustness*			*Concepts*			*Adv.*
	Model	Category	IN	IN-ReaL	IN-V2	IN-A	IN-C	IN-$\bar{\text{C}}$	IN-R	IN-S	SIN	PGD
●	Original Baseline [19]	baseline	76.15	86.50	63.14	0.03	41.12	39.70	36.16	24.09	37.12	18.39
·	PGD-AT (ℓ_2, ϵ=0) [35, 47]	adversarial training	75.81	88.65	63.70	1.85	40.90	39.48	35.76	23.50	34.38	18.88
·	PGD-AT (ℓ_2, ϵ=0.01) [35, 47]	adversarial training	75.67	84.97	63.64	1.69	42.13	39.78	36.85	24.22	38.50	30.56
·	PGD-AT (ℓ_2, ϵ=0.03) [35, 47]	adversarial training	75.77	87.42	63.33	1.92	42.25	39.72	36.71	24.60	38.50	40.05
·	PGD-AT (ℓ_2, ϵ=0.05) [35, 47]	adversarial training	75.58	84.66	62.93	1.79	41.66	40.18	37.28	24.69	40.00	46.86
·	PGD-AT (ℓ_2, ϵ=0.1) [35, 47]	adversarial training	74.79	86.20	62.44	1.88	41.91	39.35	37.61	24.70	39.25	53.76
·	PGD-AT (ℓ_2, ϵ=0.25) [35, 47]	adversarial training	74.14	85.28	61.65	1.96	42.02	39.58	38.23	25.31	40.88	61.23
•	PGD-AT (ℓ_2, ϵ=0.5) [35, 47]	adversarial training	73.17	86.50	59.97	2.08	40.82	39.23	38.94	24.21	43.88	64.30
•	PGD-AT (ℓ_2, ϵ=1) [35, 47]	adversarial training	70.42	84.36	56.95	2.09	38.79	37.90	38.95	23.68	44.12	64.37
●	PGD-AT (ℓ_2, ϵ=3) [35, 47]	adversarial training	62.83	75.77	48.91	1.87	34.60	34.83	36.99	20.93	41.75	59.47
●	PGD-AT (ℓ_2, ϵ=5) [35, 47]	adversarial training	56.14	74.54	42.49	1.77	30.65	31.15	33.09	17.24	39.00	53.63
▪	PGD-AT (ℓ_∞, ϵ=0.5) [35, 47]	adversarial training	73.74	84.36	61.38	2.29	40.11	40.04	39.39	24.68	43.88	65.11
▪	PGD-AT (ℓ_∞, ϵ=1.0) [35, 47]	adversarial training	72.04	83.44	59.21	2.20	38.82	39.72	40.96	24.51	44.00	66.39
▪	PGD-AT (ℓ_∞, ϵ=2.0) [35, 47]	adversarial training	69.09	82.52	56.15	2.39	37.49	38.85	39.33	23.10	45.75	65.25
■	PGD-AT (ℓ_∞, ϵ=4.0) [35, 47]	adversarial training	63.87	78.83	51.31	2.29	33.71	36.56	38.92	21.87	43.25	61.22
■	PGD-AT (ℓ_∞, ϵ=8.0) [35, 47]	adversarial training	54.53	71.78	41.86	2.11	28.78	31.91	34.84	18.57	40.00	52.57
▲	AugMix (180ep) [22]	augmentation	77.53	88.96	65.42	3.65	50.77	46.16	41.03	28.49	45.50	30.96
◀	DeepAugment [23]	augmentation	76.65	86.81	65.20	3.40	54.40	48.39	42.25	29.50	49.12	32.51
▶	DeepAugment+AugMix [23]	augmentation	75.80	86.20	63.65	3.85	59.53	51.34	46.79	32.62	57.50	40.40
•	Noise Training (clean eval) [29]	augmentation	67.22	83.44	54.67	2.43	44.40	39.48	36.64	19.99	47.12	48.27
×	NoisyMix [12]	augmentation	77.05	89.57	64.28	3.32	54.23	50.62	45.77	31.18	49.38	50.70
●	OpticsAugment [38]	augmentation	74.22	86.50	62.03	1.73	42.90	40.39	37.50	24.69	43.88	16.08
◆	PRIME [37]	augmentation	76.91	87.12	64.34	2.16	55.27	49.00	42.20	29.83	46.62	30.82
⬢	PixMix (180ep) [25]	augmentation	78.09	88.65	65.89	6.25	52.99	59.51	40.31	29.21	40.25	23.02
⬟	PixMix (90ep) [25]	augmentation	77.36	89.88	65.20	4.11	51.87	57.76	39.92	28.57	45.00	22.28
⬣	Shape Bias Augmentation [31]	augmentation	76.21	87.42	64.20	3.03	47.60	44.46	40.64	27.92	64.50	25.18
⅄	Texture Bias Augmentation [31]	augmentation	75.27	86.81	63.18	2.25	41.82	40.26	36.76	24.28	35.50	16.83
Y	Texture/Shape Debiased Augmentation [31]	augmentation	76.89	86.20	65.04	3.39	48.28	45.47	40.77	28.42	56.00	25.99
⬢	DINO V1 [6]	contrastive	75.28	85.28	62.70	5.15	39.61	35.88	30.17	18.75	30.63	13.26
×	MoCo V3 (1000ep) [8]	contrastive	74.60	87.42	62.01	4.07	43.53	40.76	37.05	25.51	35.50	27.79
Y	MoCo V3 (100ep) [8]	contrastive	68.91	82.52	56.28	2.43	37.75	36.62	31.71	20.48	36.75	24.61
⅄	MoCo V3 (300ep) [8]	contrastive	72.80	84.97	60.74	3.27	41.97	39.00	35.41	24.00	36.75	27.57
•	SimCLRv2 [7]	contrastive	74.90	85.58	61.24	4.65	44.32	40.73	35.16	23.55	43.88	14.57
●	SwAV [5]	contrastive	75.31	87.73	62.15	5.49	41.48	37.63	30.24	18.94	30.38	14.75
•	Frozen Random Filters [14]	freezing	74.76	87.12	62.47	2.52	45.22	40.98	37.52	25.36	40.62	16.18
⬣	ShapeNet: SIN Training [17]	stylized	60.18	73.31	48.61	2.39	39.76	36.76	40.17	30.09	90.88	12.33
•	ShapeNet: SIN+IN Training [17]	stylized	74.59	87.12	62.43	1.91	46.91	43.33	41.55	29.70	91.00	22.47
●	ShapeNet: SIN+IN Training + FT [17]	stylized	76.72	88.34	64.65	2.23	43.55	41.86	38.93	26.92	45.00	21.23
⬢	timm (A1) [64, 65]	training recipes	80.10	88.65	68.73	11.03	50.93	49.01	40.60	29.22	36.88	27.74
●	timm (A1H) [64, 65]	training recipes	80.10	89.26	68.47	15.21	49.36	48.57	40.99	29.64	39.00	36.11
⬟	timm (A2) [64, 65]	training recipes	79.80	86.20	67.29	7.36	48.98	47.83	38.39	27.27	38.38	27.03
•	timm (A3) [64, 65]	training recipes	77.55	86.81	65.04	6.35	41.03	43.20	35.93	24.61	34.75	23.32
×	timm (B1K) [64, 65]	training recipes	79.16	88.34	67.41	8.51	51.64	50.30	43.04	31.22	42.88	33.40
⅄	timm (B2K) [64, 65]	training recipes	79.27	87.42	67.79	8.64	52.25	50.05	42.44	30.40	40.75	32.82
◆	timm (C1) [64, 65]	training recipes	79.76	89.88	68.54	10.07	50.60	49.40	41.54	30.29	37.12	33.72
Y	timm (C2) [64, 65]	training recipes	79.92	90.49	68.80	11.49	51.62	50.92	40.73	29.85	37.12	30.38
⬣	timm (D) [64, 65]	training recipes	79.89	89.26	68.73	9.76	51.26	49.53	40.61	29.85	36.00	29.62
▲	torchvision (V2) [41, 59]	training recipes	80.34	90.18	69.57	16.73	50.02	49.67	41.62	28.44	38.38	39.90

Table 2. Bias measurements of ResNet-50 models. Columns with gray background were not directly used in the main paper.

			Bias								
						Critical Band					
			Shape	Spectral		*(IN-1k non-normalized)*			*(IN-1k normalized)*		
	Model	Category	Bias	Low-Freq.	High-Freq.	C-BW	C-CF	C-PNS	C-BW	C-CF	C-PNS
●	Original Baseline [19]	baseline	0.21	0.63	0.01	11295.35	3681.95	1.0	5.67	54.68	1.00
•	PGD-AT (ℓ_2, ϵ=0) [35, 47]	adversarial training	0.21	0.62	0.01	11295.35	3681.95	1.0	5.67	54.68	1.00
•	PGD-AT (ℓ_2, ϵ=0.01) [35, 47]	adversarial training	0.22	0.64	0.01	11295.35	3681.95	1.0	5.67	54.68	1.00
•	PGD-AT (ℓ_2, ϵ=0.03) [35, 47]	adversarial training	0.24	0.67	0.00	11295.35	3681.95	1.0	5.67	54.68	1.00
•	PGD-AT (ℓ_2, ϵ=0.05) [35, 47]	adversarial training	0.24	0.67	0.00	11295.35	3681.95	1.0	5.91	41.85	1.00
•	PGD-AT (ℓ_2, ϵ=0.1) [35, 47]	adversarial training	0.28	0.69	0.00	11295.35	3681.95	1.0	6.45	40.65	1.00
•	PGD-AT (ℓ_2, ϵ=0.25) [35, 47]	adversarial training	0.34	0.72	0.00	11295.35	3681.95	1.0	7.86	58.20	1.00
•	PGD-AT (ℓ_2, ϵ=0.5) [35, 47]	adversarial training	0.41	0.73	0.00	11295.35	3681.95	1.0	7.86	58.20	1.00
•	PGD-AT (ℓ_2, ϵ=1) [35, 47]	adversarial training	0.48	0.75	0.00	11295.35	3681.95	1.0	7.86	58.20	1.00
●	PGD-AT (ℓ_2, ϵ=3) [35, 47]	adversarial training	0.65	0.76	0.00	11295.35	3681.95	1.0	7.47	32.63	0.60
●	PGD-AT (ℓ_2, ϵ=5) [35, 47]	adversarial training	0.69	0.78	0.00	11295.35	3681.95	1.0	7.92	45.71	0.55
▪	PGD-AT (ℓ_∞, ϵ=0.5) [35, 47]	adversarial training	0.37	0.73	0.00	11295.35	3681.95	1.0	7.86	58.20	1.00
▪	PGD-AT (ℓ_∞, ϵ=1.0) [35, 47]	adversarial training	0.45	0.74	0.00	11295.35	3681.95	1.0	7.37	39.19	0.73
▪	PGD-AT (ℓ_∞, ϵ=2.0) [35, 47]	adversarial training	0.54	0.75	0.00	11295.35	3681.95	1.0	7.92	45.71	0.55
■	PGD-AT (ℓ_∞, ϵ=4.0) [35, 47]	adversarial training	0.62	0.75	0.00	11295.35	3681.95	1.0	10.46	114.50	0.52
■	PGD-AT (ℓ_∞, ϵ=8.0) [35, 47]	adversarial training	0.72	0.78	0.00	11295.35	3681.95	1.0	7.47	32.63	0.60
▲	AugMix (180ep) [22]	augmentation	0.30	0.74	0.02	9.95	34.70	1.0	4.74	38.85	1.00
◀	DeepAugment [23]	augmentation	0.39	0.77	0.06	11295.35	3681.95	1.0	5.51	56.31	0.72
▶	DeepAugment+AugMix [23]	augmentation	0.52	0.84	0.09	9.95	34.70	1.0	4.67	38.28	0.63
•	Noise Training (clean eval) [29]	augmentation	0.51	0.79	0.01	11295.35	3681.95	1.0	6.35	33.58	0.93
×	NoisyMix [12]	augmentation	0.32	0.75	0.01	9.95	34.70	1.0	4.74	38.85	1.00
●	OpticsAugment [38]	augmentation	0.24	0.63	0.01	11295.35	3681.95	1.0	4.45	61.21	1.00
◆	PRIME [37]	augmentation	0.32	0.71	0.13	7.34	42.82	1.0	4.74	38.85	1.00
⬢	PixMix (180ep) [25]	augmentation	0.26	0.68	0.03	5.91	41.85	1.0	4.74	38.85	1.00
⬟	PixMix (90ep) [25]	augmentation	0.23	0.67	0.03	6.45	40.65	1.0	4.74	38.85	1.00
⬣	Shape Bias Augmentation [31]	augmentation	0.28	0.66	0.01	11295.35	3681.95	1.0	4.74	38.85	1.00
⅄	Texture Bias Augmentation [31]	augmentation	0.20	0.64	0.01	11295.35	3681.95	1.0	5.67	54.68	1.00
Y	Texture/Shape Debiased Augmentation [31]	augmentation	0.26	0.67	0.01	11295.35	3681.95	1.0	5.67	54.68	1.00
⬣	DINO V1 [6]	contrastive	0.18	0.44	0.01	7.34	42.82	1.0	4.74	38.85	1.00
×	MoCo V3 (1000ep) [8]	contrastive	0.33	0.56	0.01	7.34	42.82	1.0	4.74	38.85	1.00
Y	MoCo V3 (100ep) [8]	contrastive	0.30	0.55	0.01	8.22	33.66	1.0	4.74	38.85	1.00
⅄	MoCo V3 (300ep) [8]	contrastive	0.31	0.55	0.01	7.34	42.82	1.0	4.74	38.85	1.00
•	SimCLRv2 [7]	contrastive	0.23	0.55	0.01	11295.35	3681.95	1.0	4.74	38.85	1.00
●	SwAV [5]	contrastive	0.18	0.43	0.01	7.34	42.82	1.0	4.74	38.85	1.00
•	Frozen Random Filters [14]	freezing	0.31	0.68	0.01	11295.35	3681.95	1.0	4.74	38.85	1.00
⬢	ShapeNet: SIN Training [17]	stylized	0.81	0.56	0.04	11295.35	3681.95	1.0	4.42	45.57	0.67
•	ShapeNet: SIN+IN Training [17]	stylized	0.35	0.63	0.01	11295.35	3681.95	1.0	4.33	39.67	1.00
●	ShapeNet: SIN+IN Training + FT [17]	stylized	0.20	0.64	0.01	11295.35	3681.95	1.0	5.67	54.68	1.00
⬢	timm (A1) [64, 65]	training recipes	0.21	0.63	0.02	5.67	54.68	1.0	3.96	42.09	1.00
●	timm (A1H) [64, 65]	training recipes	0.17	0.61	0.02	5.67	54.68	1.0	3.71	46.56	1.00
⬟	timm (A2) [64, 65]	training recipes	0.16	0.62	0.01	5.67	54.68	1.0	4.33	39.67	1.00
•	timm (A3) [64, 65]	training recipes	0.13	0.58	0.01	9.95	34.70	1.0	4.67	54.89	1.00
×	timm (B1K) [64, 65]	training recipes	0.19	0.64	0.02	5.67	54.68	1.0	4.14	47.02	0.91
⅄	timm (B2K) [64, 65]	training recipes	0.18	0.65	0.02	5.67	54.68	1.0	4.14	47.02	0.91
◆	timm (C1) [64, 65]	training recipes	0.18	0.64	0.02	5.67	54.68	1.0	3.96	42.09	1.00
Y	timm (C2) [64, 65]	training recipes	0.18	0.62	0.02	5.67	54.68	1.0	3.92	53.43	0.86
⬣	timm (D) [64, 65]	training recipes	0.17	0.63	0.01	5.67	54.68	1.0	3.96	42.09	1.00
▲	torchvision (V2) [41, 59]	training recipes	0.17	0.66	0.01	5.67	54.68	1.0	3.92	53.43	0.86

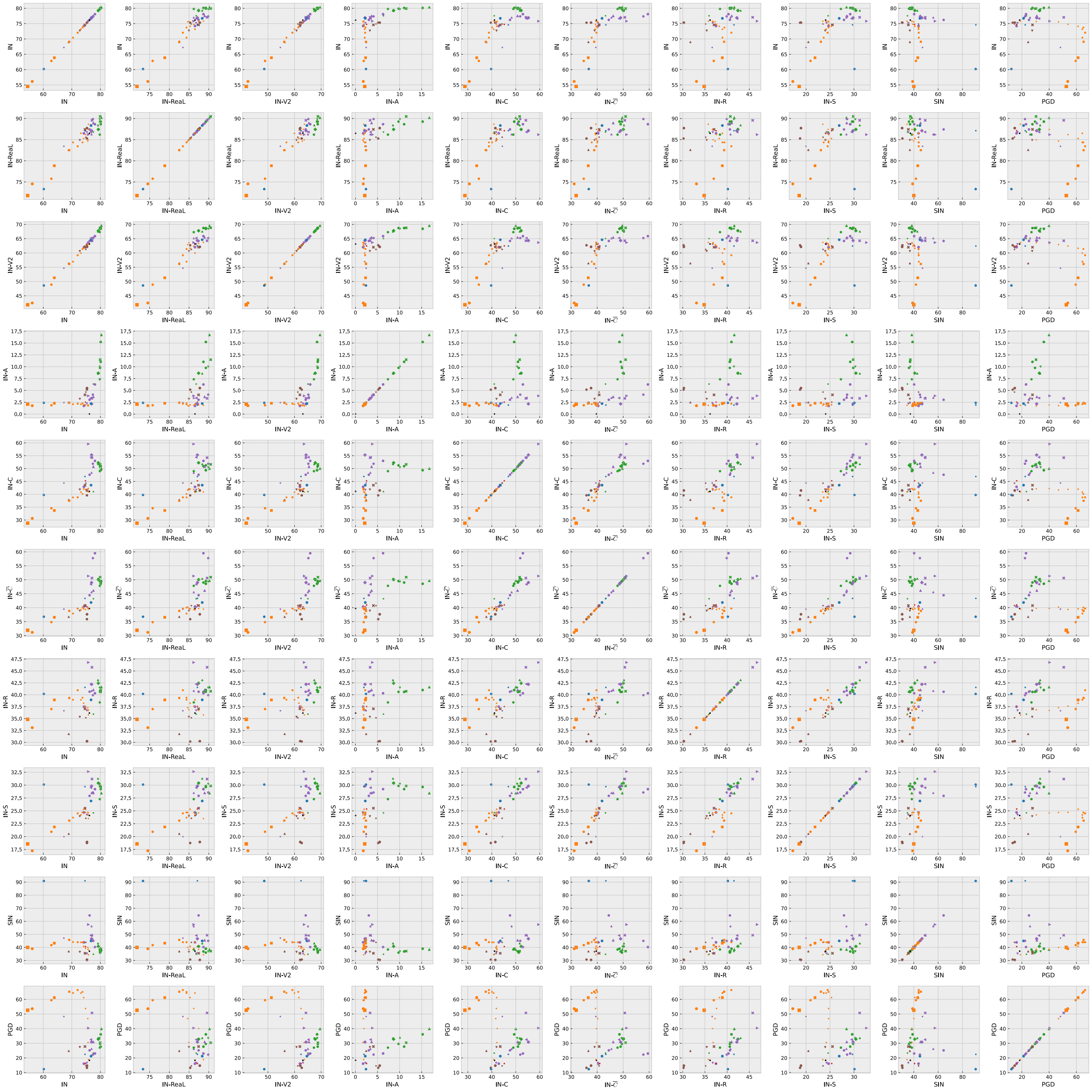
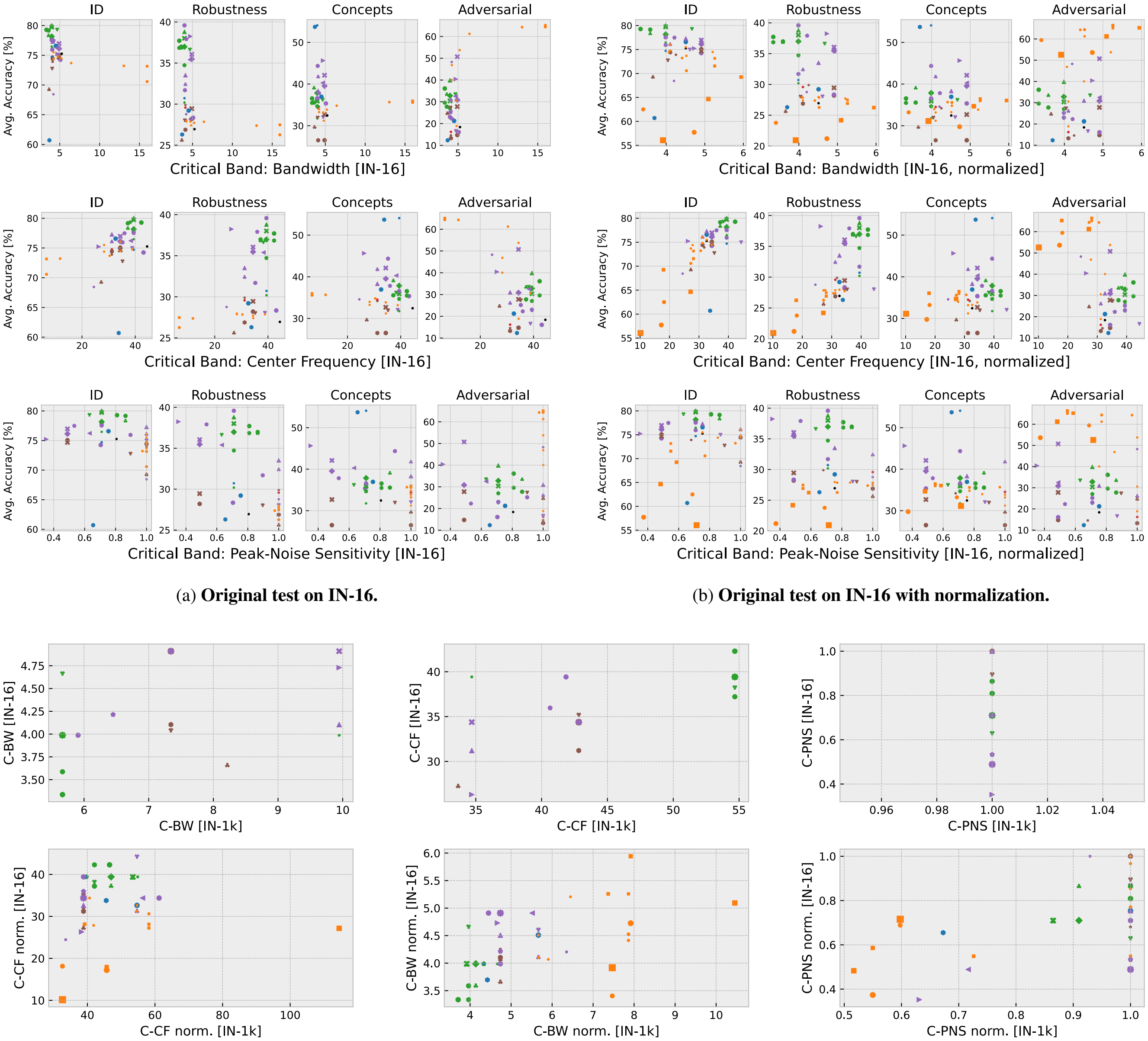


Figure 15. Performance comparison on all dataset pairs. Markers indicate models as described by the legend in Tab. 1.



(a) **Original test on IN-16.**

(b) **Original test on IN-16 with normalization.**

(c) **Critical band evaluation on IN-1k and IN-16 in comparison.** We compare original (top) and normalized (bottom) evaluations.

Figure 16. Measurement of the critical band following the original methodology of Subramanian et al. [52]. **(a)** original test; **(b)** original test with normalized accuracy; **(c)** Comparison between results in ImageNet (IN-1k) as in the main paper and the 16-super-class subset (IN-16). Models with unreasonable measurements (C-BW $\geq$ 100) were removed. Markers indicate models as described by the legend in Tab. 1.