LOW-FREQUENCY SHORTCUTS IN TEXTURE-DRIVEN VISUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks suffer from shortcut learning, where learned features generalize well to the training set but not to in-distribution (ID) or out-of-distribution (OOD) test sets. Existing studies are all based on a few standard benchmarks, which are shape-driven. Numerous application domains, however, are texture-driven. In this work, we present shortcut learning analysis for texture-driven domains, and compare it with that of a standard benchmark. We show that texture-driven domains suffer from low-frequency shortcuts. They make the majority of their decisions based on a few low-frequency components (LFCs) with a skewed spectral behavior, despite that higher-frequency components (HFCs) have a higher predictive power. Pruning LFCs from training and test sets mitigates the shortcut and provides a more balanced spectral behavior, improving the ID accuracy by up to 10% and OOD accuracy by up to 40% under algorithmic and real-world domain shifts. We show that general-purpose and domain-specific foundation models can also suffer from low-frequency shortcuts. While large models can mitigate the shortcuts, they incur a high computational cost and may result in a significantly lower accuracy than shortcut-pruned from-scratch trained small models. We show that reduced image resolutions amplify the degree of shortcuts; large frequency-transformation block sizes capture low-frequency shortcuts better than small block sizes; and, low-frequency shortcuts persist across different color spaces. Our findings provide valuable insights, which we hope will be useful for practitioners working on new, understudied domains.

1 INTRODUCTION

Simplicity bias & shortcut learning. Neural networks are biased towards learning simple functions that solve the optimization problem fast, rather than complex functions that are faithful to the semantics of the problem [24; 36; 56; 64; 67; 72; 91; 93; 100]. This results in shortcut learning, where learned features are strong predictors in the validation set, but do not generalize well into out-of-distribution (OOD) test sets. For example, neural networks might learn decision rules based on superficial cues, such as the source tag in horse images in the Pascal VOC dataset [19], and hence suffer from low generalization performance and unreliable predictions.

Frequency analysis for shortcut learning. Simplicity bias might occur due to visible [19; 20] or invisible superficial cues [91; 93]. While visible features are easy to detect via visual inspection, invisible features are deeply embedded in the data and require formal tools to analyze [18; 23; 61; 68; 84; 91; 93; 108]. One such useful tool is Fourier analysis [47; 89; 91]. Fourier analysis transforms the data from its original domain, e.g., the spatial domain for images, into the frequency domain. Frequency representation provides a structured view of the data and enables principled analysis of learning dynamics by leveraging the data's frequency structure.

Existing studies are limited to shape semantics. Existing frequency analysis studies are mostly based on a few standard benchmarks, such as CIFAR-10 and ImageNet [47; 89; 91]. Standard benchmarks consist of natural images of everyday objects, where classification information is centered around specific objects with well-known global shapes. Therefore, known shortcuts, such as spectral bias [67], texture bias [35], or shape sensitivity [21], are all relative to object-centric representations, which assume that shape is the dominant cue and analyze how frequency components interfere with or support the shape semantics.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

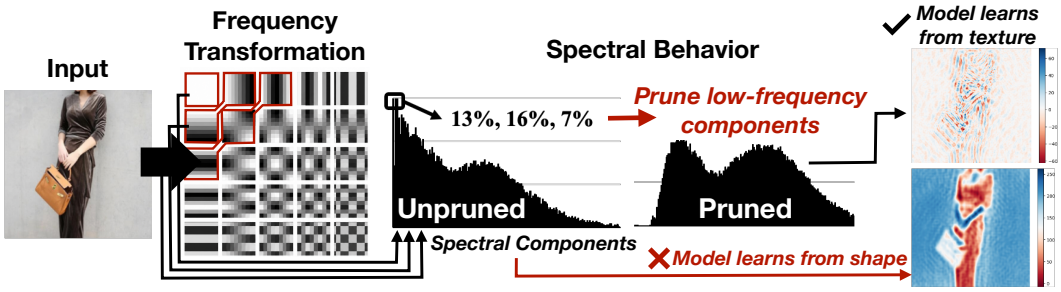


Figure 1: Spectral behavior of models trained on texture-driven domains show that models make majority of their decisions based on a few low-frequency components (first three figures above, from left). Once we prune low-frequency components (LFCs) from training and test sets, spectral behavior shifts towards higher-frequencies (forth figure above), and we obtain a significantly higher ID accuracy, up to 10% (Section 5) These results show that texture-driven domains suffer from low-frequency shortcuts, where models learn from LFCs despite that higher-frequency components have a higher predictive power. We show that low-frequency shortcuts persist across different model sizes, architectures, and hyperparameters (Section 5), under various OOD conditions (Section 6 & 7), for pretrained foundation models (Section 8), and different resolutions, transformation block sizes, and color spaces (Section 9).

Numerous application domains are texture-driven. Computer vision applications, however, span a wide range of domains that are texture-driven, such as histopathology [57] and textile classification [110]. These domains include images, such as microscopic tissue scans or textile samples, where the notion of "shape" is ill-defined or irrelevant (see Figure 3). Classification information is distributed across the image, with repetitive fine-grained patterns rather than being centered on well-defined objects. Spectral cues that drive model predictions shift from global shape structures to fine-grained spatial patterns. As a result, whether known or new shortcuts exist in texture-driven domains, and, if so, how they affect model predictions and robustness, is unclear.

Shortcut learning for texture-driven domains. We study shortcut learning in texture-driven domains and compare it to a standard benchmark using frequency analysis tools. We consider histopathology, textile classification, ground terrain recognition, AI-generated images, morphological classification of galaxies, and land use/cover classification, analyzing one image classification task per domain. As a standard benchmark, we use CIFAR-10 [48] due to its widespread adoption in frequency-based studies [13; 47; 89]. We perform a comprehensive frequency-spectrum analysis covering low, mid, and high frequencies, using different model architectures, sizes, and hyperparameters, as well as pretrained, general-purpose and domain-specific vision foundation/language models, based on in-distribution (ID) and out-of-distribution (OOD) performance. In summary, our **contributions** are as follows.

- We show that texture-driven domains suffer from low-frequency shortcuts. Neural networks make the majority of their decisions based on a few low-frequency components (LFCs), despite that higher-frequency components (HFCs) have higher predictive power. This results in severely suboptimal ID as well as OOD generalization performance.
- We show that pruning LFCs can mitigate the shortcuts and provide up to 10% higher ID and 40% higher OOD performance across different model architectures, sizes, optimizers, and hyperparameters, under various OOD conditions. To the best of our knowledge, we are the first study showing that shortcuts can hurt ID performance, as well as OOD performance.[1].
- We study general-purpose and domain-specific VFM/VLMs and show that they too can suffer from low-frequency shortcuts. While large models can mitigate the shortcuts, they incur a high computational cost and may result in a significantly lower accuracy than shortcut-pruned from-scratch trained small models.

[1]In their theoretical study, Shah et al. [72] also have shown that simplicity bias can hurt both ID and OOD performance, but using only a synthetic toy dataset and a single-layer simple neural network (see their Section 5 and Table 2). We show that simplicity bias is not only an issue for highly controlled toy scenarios, but actually a real-world phenomenon across a wide range of real-world datasets and models.

- We study low-frequency shortcuts across different image resolutions, frequency-transformation block sizes, and color spaces. We show that reduced image resolutions increase the degree of shortcuts; small block sizes only partially mitigate the shortcuts; and, shortcuts persist across different color spaces.

Our study provides valuable insights and improves our understanding of the learning dynamics of neural networks. We hope our findings will help practitioners working in new, understudied domains when designing deep learning systems.

2 Diagnostic Framework: Frequency Pruning & Spectral Behavior

Frequency transformations provide a structured view of data [47]. We adopt a pruning-based analysis pipeline (Figure 2, top), where RGB images are transformed to the frequency domain, selectively pruned by zeroing frequency components, and inverse-transformed back to RGB for training/inference. We use the commonly used discrete cosine transform (DCT) [14; 86; 76; 97], applying it to the full image as a single block. Each color channel is transformed independently and pruned identically.

When pruning, we remove frequency components diagonally from the top-left to the bottom-right of the image (or vice versa), since oscillation rates and spatial complexity increase along this direction. Each such diagonal is referred to as a frequency component; the terms diagonal, frequency component, and component are used interchangeably. Pruning is used for **sensitivity analysis** and for **accuracy contributions**.

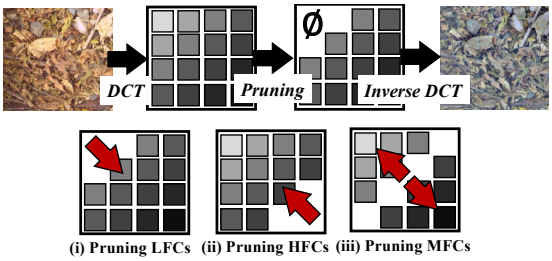


Figure 2: Frequency analysis methodology.

Sensitivity Analysis. We evaluate three pruning strategies (Figure 2, bottom): (i) pruning low-frequency components (LFCs), (ii) high-frequency components (HFCs), and (iii) mid-frequency components (MFCs). Each strategy removes an increasing number of diagonals in a prescribed order. Both training and test sets are pruned, and models are trained from scratch for every level of pruning.

Diagonal-wise Analysis. Existing studies either use broad units of frequency coefficients [13; 47], or perform an analysis one coefficient at a time [91]. Using broad units limits the findings, whereas one coefficient at a time analyses are too costly. Diagonal-wise grouping provides a middle ground.

Spectral Behavior via Accuracy Contributions. We fix the training set to either unpruned or pruned for a specific number of components, and prune test images one component at a time, attributing the resulting accuracy drop to the removed component as its accuracy contribution. The resulting distribution across components provide the spectral behavior. Distribution shifts induced by training-time interventions reflect changes in learning dynamics rather than post-hoc sensitivity.

Low-Frequency Shortcuts. We define a low-frequency shortcut as a skewed spectral behavior towards LFCs and improved ID accuracy via LFC-pruning. Our definition follows [91; 93] in terms of the skewed spectral behavior, but extends it by further examining the ID behavior. We do so as some domains may contain useful information within a narrow band of LFCs. [36] define shortcuts as the generalization gap between ID and OOD performance, i.e., poor OOD performance compared to the ID performance; we define low-frequency shortcuts as a broader term, including poor ID and OOD performance.

3 Application Domains

Histopathology. Histopathology analyzes microscopic tissue images for diagnosis and prognosis of diseases [45; 57; 78; 95]. Classification relies on morphological structures capturing cellular organization (1st image in Figure 3).

Textile Classification. Driven by fashion e-commerce [85], textile classification supports applications ranging from virtual try-ons [39] to recycling [49; 55]. The task requires recognizing repetitive, fine-grained patterns distributed across the image (2nd image in Figure 3).

Ground Terrain Recognition. Ground terrain recognition supports applications such as autonomous driving [73; 103] and robot navigation [38; 112]. Terrains correspond to surface types (e.g., leaves, grass) and are classified using surface spatial cues and texture (3rd image in Figure 3).

AI-Generated Images. With the rapid growth of generative AI, determining whether an image is real or fake has become an important application [111; 62; 59]. The task requires identifying subtle, low-level differences between real and AI-generated images of the same objects (4th image in Figure 3).

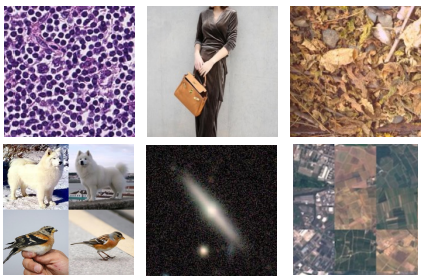


Figure 3: Sample images for the texture-driven domains we study.

Galaxy Morphologies. Morphological structures of galaxies are used for various purposes such as star formation. While morphologies themselves are shape-driven, the pixels creating the morphology, i.e., the stars are texture-driven [87] (5th image in Figure 3).

Land Use/Cover Classification. Land use/cover classification using satellite images have numerous applications such as agriculture and urban development [17; 104]. It includes both texture-driven, e.g., herbaceous vegetation, and shape-driven categories, e.g., high-way [41; 42] (6th image in Figure 3).

4 Analysis Setup

Datasets. We evaluate: SPIDER Colorectal histopathology (77K samples, 14 classes) [44; 57]; TextileNet (fabric subset; 350K samples, 27 classes) [74; 110]; GTOS ground terrain recognition (30K samples, 40 classes) [102]; TinyGenImages (35K samples, 2 classes) [111; 4; 6][2]; Galaxy MNIST (10K samples, 4 classes) [2]; EuroSAT (27K sampels, 10 classes) [1]; CIFAR-10 (60K samples, 10 classes) [48]; NICO++ domain-generalization (89K samples, 60 classes) [107]. All experiments use 70%-10%-20% train/val/test splits. Validation accuracies are reported. Split details are in Appendix A; test accuracies follow validation results (Appendix B).

Models. We use ResNet-50 (25.6M) [40], MobileNet-V3 (2.5M) [46], ViT-Small (22M), and ViT-Tiny (5.7M) [29] as the classifiers. We train models from scratch with PyTorch's default initialization [65]. We use DinoV2 (VFM, 86M) [3], DinoV3 (VFM, 840M) [75], and CLIP (VLM, 151M) [66] as the general-purpose foundation models; and, Hibou-L (VFM, 300M) [57] and CONCH V1.5 (VLM, 307M) [26] as the domain-specific foundation models for histopathology. We train a linear layer on top of their frozen backbones. We defer further details of the foundation models to Appendix A.

Training. We follow the standard CIFAR-10 training protocol for CNNs [13; 47; 89; 91; 93]; ViT & DeiT for ViTs [29; 82][3]; and, linear-probing protocols for DinoV2, DinoV3, CLIP, Hibou-L, and CONCH [7; 9; 10; 12; 8]. Images are resized to 224×224, except EuroSAT (64×64) and CIFAR-10 (32×32). Experiments are repeated with three seeds and reported as mean $\pm$ standard deviation. We report the results with seed of 42 for spectral behaviors and defer other seeds to Appendix C. Complete training details are provided in Appendix A.

5 Evidence of Low-Frequency Shortcuts in Texture-Driven Domains

Figure 4 shows ID accuracy for LFC (dark), MFC (light orange), and HFC (red) pruning for the five classification tasks trained with ResNet-50. The x-axis reports the number of pruned coefficients

[2]Original GenImages dataset contains one million generated images across eight different generative models [4]. TinyGenImages uniformly samples 5K images across seven of the eight generative models [6]. We further tested recognizing images generated by two different generative models, e.g., Midjourney to Glide, using all available images. We observed similar results. We defer these results to Appendix E.

[3]See Table 3 of [29], and Table 9 of [82].

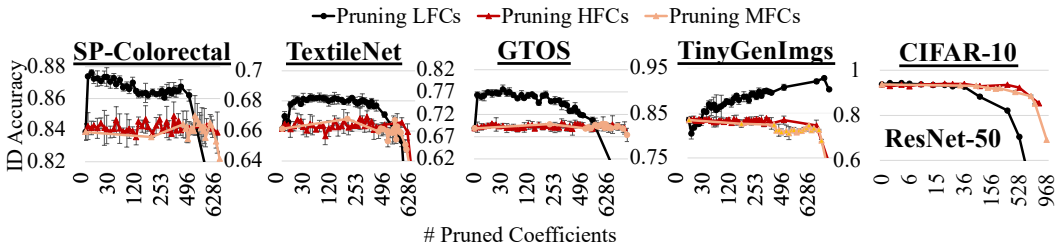


Figure 4: ID accuracy results for pruning LFCs (dark line), MFCs (light orange line), and HFCs (red line). Pruning LFCs eliminates the shortcut and improves accuracy for texture-driven tasks.

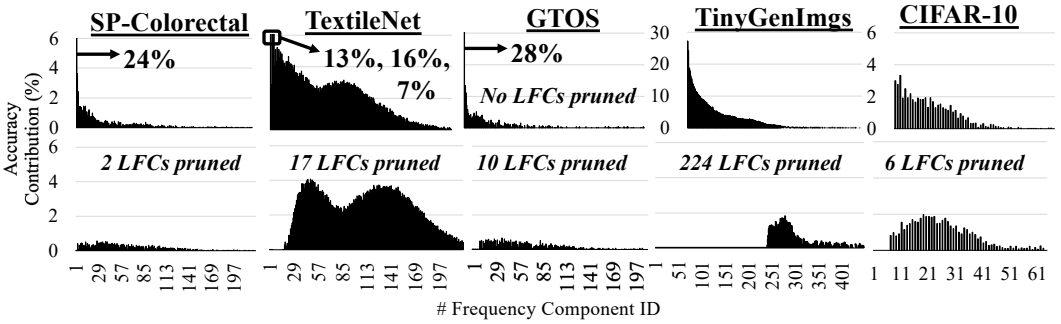


Figure 5: Accuracy contributions with unpruned (top) and pruned (bottom) training and test images.

(unscaled to capture fine-grained per-diagonal effects), and the y-axis shows ID accuracy. As can be seen, pruning HFCs and MFCs do not significantly impact the ID accuracy, as they constitute a small portion of the overall energy in images [76; 97; 96].On the other hand, **pruning LFCs improves the accuracy for all the texture-driven domains**[4].

Figure 5 presents the spectral behavior of ResNet-50 trained on unpruned images (top). Frequency component IDs increase diagonally from the top-left to the bottom-right, with accuracy contributions shown on the y-axis. As shown, **accuracy contributions are highly skewed toward LFCs** for all texture-driven domains. In contrast, CIFAR-10 exhibits a more balanced distribution. Figure 5 presents the spectral behavior for pruned training images (bottom). For each task, we prune the number of LFCs that maximizes ID accuracy in Figure 4. As can be seen, **contributions shift toward higher frequencies** with a more balanced and unbiased distribution across all domain-specific tasks.

These results indicate that **texture-driven domains suffer from low-frequency shortcuts**. Neural networks rely exponentially more on LFCs than they do on HFCs, despite that higher-frequency features have a higher predictive power. Pruning LFCs mitigates the shortcut by shifting the spectral behavior towards higher frequencies, which in turn provides up to 10% higher ID performance.

LFCs are background features, such as shading or camera effects. These features constitute a small set of simple features that neural networks fall into shortcut. HFCs, on the other hand, are large number of fine-grained features, and hence harder to learn from.

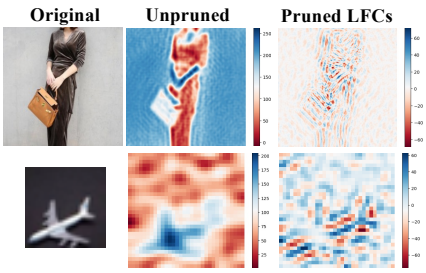


Figure 6: Sample images from TextileNet (top) and CIFAR-10 (bottom).

Prior Theoretical Work. Simple features has been shown to have faster and stronger growing gradients than complex ones [64; 72; 105]. We provide a detailed analysis of existing theoretical work in Appendix D.

[4]CIFAR-10's accuracy also slightly improves (by 0.6%), but only minimally compared to texture-driven tasks and only a little above the standard deviation (0.3%).

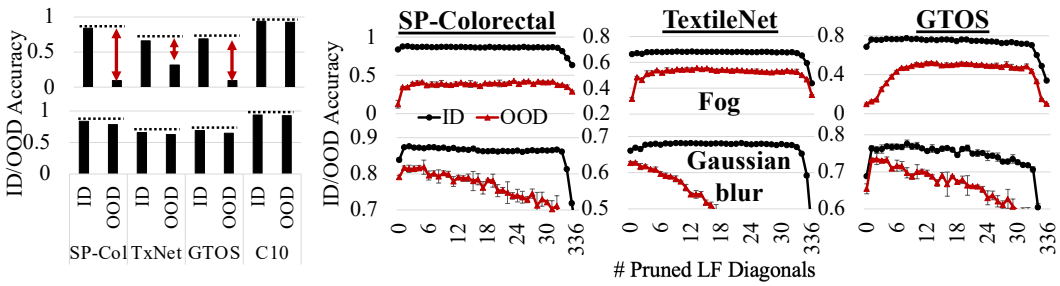


Figure 7: OOD results for fog (top) and Gaussian blur (bottom) corruptions for ResNet-50.

Visuals. Figure 6 presents a reconstructed image from TextileNet (top row) using top-contributing 50 frequency components with unpruned and pruned (17 LFCs) trainings using ResNet-50. Unpruned training learns from background and shape structures, whereas pruned training learns from fine-grained information. Further visuals are in Appendix F.

CIFAR-10. CIFAR-10 obtains most of its accuracy from the lower end of the frequency spectrum (Figure 5). Figure 6 (bottom row) presents a reconstructed sample image using top-contributing 10 frequency components with unpruned and pruned (6 LFCs) trainings. Unpruned training learns from smooth features, whereas pruned training learns from irregular structures.

Impact of Model Architecture, Size, and Hyperparameters. We analyzed MobileNet-V3 (2.5M params), ViT-Small (22M params), and ViT-Tiny (5.7M params), and reached similar conclusions. We further analyzed different optimizers and hyperparameters, such as learning rate, momentum, weight decay, and scheduler, for ResNet-50 and ViT-Small, and reached similar conclusions. We defer the results to Appendix G and M.

Summary. Texture-driven domains suffer from low-frequency shortcuts, where neural networks make majority of their decisions based on a few low-frequency components (LFCs), despite higher-frequency components (HFCs) have a higher predictive power. Pruning LFCs mitigate the shortcuts and provide up to 10% improved ID accuray with a more balanced spectral behavior.

6 Low-Frequency Shortcuts In OOD Classification

6.1 Algorithmic Corruptions

We use ImageNet-C benchmark [43]. We observe three main categories of corruptions in ImageNet-C: (i) low-frequency, (ii) high-frequency, and (iii) mixed corruptions [5]. We select one corruption per category: fog, Gaussian blur, and elastic transform. We report severity 1 results with ResNet-50 and defer the rest to Appendix H and I. We consider two pipelines: (i) corrupt-then-prune and (ii) prune-corrupt-prune. We report results for (i) and defer (ii) to Appendix J.

1. Low-Frequency Corruption: Fog. The left graph at top of Figure 7 presents ID and OOD accuracies for unpruned training. OOD accuracy of texture-driven tasks heavily drops, as high as 70%, compared to their ID accuracy. This is due to the low-frequency shortcuts, which the fog corruption heavily modifies. The right three graphs at top of Figure 7 present ID (dark line) and OOD (red line) accuracies. Pruning LFCs substantially improves OOD performance, by up to 40%, thanks to eliminating the low-frequency shortcuts. CIFAR-10's OOD accuracy follows nearly the same curve as its ID accuracy (see Figure 4). This is because CIFAR-10 has a more balanced spectral behavior, and hence is more robust. We defer CIFAR-10's OOD results to Appendix K.

2. High-Frequency Corruption: Gaussian Blur. The left graph at bottom of Figure 7 presents ID and OOD accuracies for unpruned training. ID and OOD accuracies are close to each other. This is because texture-driven tasks heavily rely on LFCs when trained on unpruned images. As a result, corrupting HFCs has a small impact on models' decisions. The right three graphs at bottom of Figure 7 present ID and OOD accuracies. OOD accuracies exhibit both increasing and decreasing trends. On the one hand, generalization performance improves as HFCs better represent task semantics. On

[5] See Figure 2 of [106] and Figure 31 in Appendix T.

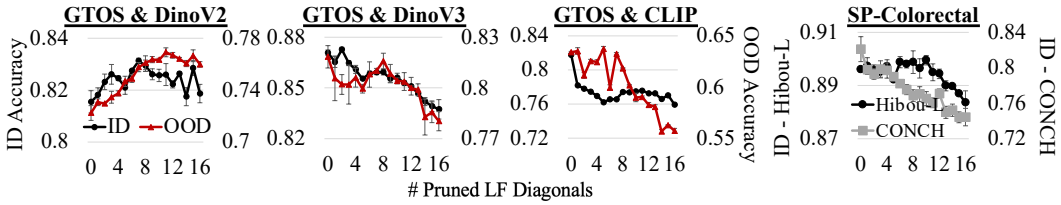


Figure 10: ID/OOD accuracy results for general-purpose (DinoV2, DinoV3, and CLIP) and domain-specific (Hibou-L and CONCH) pretrained foundation models.

the other hand, generalization performance decreases if the corruption corrupts the HFCs that the model has started to rely on. Final accuracy depends on which factor has a greater impact.

3. Mixed Corruption: Elastic Transform. Elastic transform corrupts both low- and high-frequency components, each at lower magnitudes than fog and Gaussian blur. Its behavior is somewhere in-between fog and Gaussian blur. We defer its results to Appendix I.

6.2 Real-World Domain-Generalization

NICO++ includes real-world photos of same category of images, such as rabbits or flowers, from six background domains: autumn, dim, grass, outdoor, rock, and water [107]. We train ResNet-50 with four domains combined and test it with one of the remaining two domains. We use initials of each domain to name each experiment.

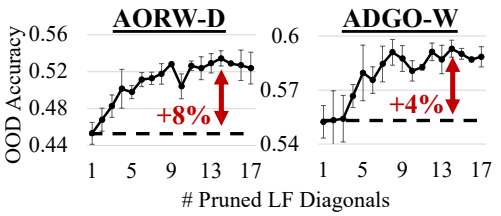


Figure 8: Domain generalization severely suffers from low-frequency shortcuts.

Figure 8 shows that accuracy is significantly improved, by 8% and 4%, as we prune LFCs. We consistently observed a similar behavior for seven other subsets of NICO++[6]. We further observed spectral behavior of all cases are skewed towards LFCs (Appendix L). These show that low-frequency shortcuts play a crucial role in real-world domain-generalization. When there are a class-consistent low-frequency background cues, models are heavily biased towards learning from them, resulting in a highly suboptimal OOD performance. Pruning LFCs allow models to learn from more realistic higher-frequency cues with an up to 8% improved accuracy. We provide all results in Appendix L.

OOD Summary. Low-frequency shortcuts play a crucial role in algorithmic as well as real-world OOD classification problems. Pruning LFCs allow models learn from realistic cues, providing up to 40% and 8% improved accuracy for algorithmic and real-world OOD problems.

7 Low-Frequency Shortcuts in Mixed Texture-Shape Domains

Figure 9 presents ID (left y-axis, dark line) and OOD (right y-axis, red line) accuracy for GalaxyMNIST and EuroSAT. We report average OOD accuracy over the three corruptions used in Section 6.1. ID accuracy drops ∼1%, whereas OOD accuracy is improved by up to 3 to 7%. Both tasks have a skewed spectral behavior towards LFCs (Appendix O). The slight drop in ID performance shows that LFCs do not constitute a shortcut for these tasks. **Nevertheless, their skewed spectral behavior make them vulnerable to OOD corruptions**, where LFC pruning significantly improves OOD performance.

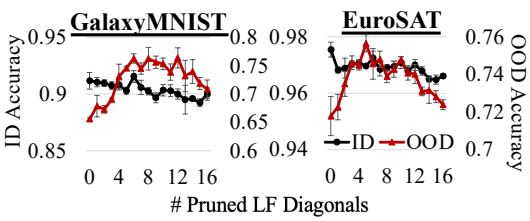


Figure 9: Skewed spectral behavior makes models vulnerable to OOD corruptions.

[6]These are AORW-G, ADGO-R, G-O, O-G, D-W, R-W, and W-D.

8 Impact of Large-Scale Pretraining on Low-Frequency Shortcuts

General-Purpose Pretraining. Left three graphs Figure 10 present ID (left y-axis, dark line) and OOD (right y-axis, red-line) accuracy, as we prune LFCs (x-axis) for DinoV2 (86M), DinoV3 (840M), and CLIP (151M) using GTOS dataset. DinoV2's ID and OOD accuracies are significantly improved by up to 1.5% and 4%, respectively. Furthermore, DinoV2 has a skew in its spectral behavior towards first few LFCs at 6% and 5% (Appendix P). This shows that **DinoV2 suffers from low-frequency shortcuts**, even though it significantly reduces the impact of shortcuts compared to from-scratch trained ResNet-50 (third graphs in Figure 4 and 5). DinoV3's ID accuracy remains stable for the first few LFCs and then follows a decreasing trend. Its spectral behavior contains a slight skew towards first few LFCs at 3% and 2.6%. This shows that **DinoV3 mitigates the shortcuts, which, however, comes at the cost of high computational cost**, as it requires 10x more parameters than DinoV2 we use and might not be feasible for some real-world use-cases such as outdoor robotics. CLIP's ID accuracy sharply drops. Its spectral behavior contains heavy skew towards LFCs at 14%, 8%, 15%, and 11% at first four LFCs. This shows that **CLIP is heavily biased towards LFCs**. Pruning LFCs results in an information loss that CLIP cannot recover. Unlike DinoV2 and DinoV3, CLIP is not trained with complex data augmentations. Furthermore, training with text-image pairs rather than purely with images is likely to amplify the low-frequency bias, as the objective rewards broad textual descriptions of objects.

Domain-Specific Pretraining. Right-most graph in Figure 10 presents ID accuracy of Hibou-L (300M, VFM, left y-axis, dark line) and CONCH (307M, VLM, right y-axis, gray line) using SP-Colorectal dataset. Hibou-L's accuracy remains stable and then follows a decreasing trend. It spectral behavior contains a skew towards first few LFCs at 6 to 9%. This shows that **Hibou-L mitigates the shortcuts**. While it highly relies on LFCs, it is able to recover from the lost information and maintain a similar level of accuracy as we prune LFCs. In contrast, CONCH's accuracy sharply drops, which shows that CONCH is heavily biased towards LFCs, and it cannot recover the lost information, similar to CLIP.

CONCH's accuracy (0.82) is less than from-scratch trained ResNet-50 (0.84). Furthermore, shortcut-pruned ResNet-50 achieves 0.88 accuracy, which is only 2% less than Hibou-L (0.9), despite being >10x smaller (25M). This shows that, while domain-specific pretraining can mitigate the shortcuts, (i) **they may achieve a lower accuracy than from-scratch trained small models**, and (ii) **they are very expensive with only a few percent advantage** compared to shortcut-pruned from-scratch trained models.

9 Impact of Resolution, Block Size, and Color Space on Low-Frequency Shortcuts

Resolution. We analyze 112x112 and 56x56 images (down from 224x224) for GTOS and SP-Colorectal. We observe that **the impact of shortcuts increases as we lower the resolutions**. The accuracy contribution of the lowest-frequency component increases from 28% to 41% and 47% for GTOS, and from 24% to 33% and 35% for SP-Colorectal. Reducing image resolution is equivalent to concentrating information in lower-frequency bands, which results in an increased degree of low-frequency shortcuts [54][7]. We provide complete results in Appendix Q.

Frequency-Transformation Block Size. Frequency transformation is often used with various block sizes [86; 76; 98]. We vary the block size from 8x8, to 16x16, 32x32, and to 224x224. We perform LFC-pruning and observe ID behavior for GTOS, SP-Colorectal, and TextileNet. We observe that **large block sizes mitigate shortcuts better than small block sizes**, as they capture background cues better. We provide the results in Appendix R.

Color Space. We study YCrCb color space. Cr/Cb channels encode color information, and Y channel encode grayscale information. We convert original RGB images into YCrCb images, perform frequency transformation, LFC-pruning, and then convert images back to YCrCb, and then RGB. We use only Y, only CrCb, and all YCrCb channels with SP-Colorectal and GTOS datasets. We observe

[7] See Chapter 5.1.3.1 and Figure 5.5 of [54]. We used PyTorch's default bilinear interpolation and frequency-domain resizing from [54] and observed similar results.

that **pruning LFCs from only Y or only from CrCb is not enough to eliminate the shortcuts**. Pruning LFCs from all YCrCb or RGB channels is similar. Results are in Appendix S.

10 Discussion & Limitations

Optimally Mitigating Low-Frequency Shortcuts. Our work identifies a new shortcut for a class of understudied domains through extensive experimental analysis. While pruning is used as part of our diagnostic methodology, it alone does not stand for an algorithm for optimally mitigating low-frequency shortcuts. Developing such an algorithm is an exciting direction for future work.

Frequency-Pruning as Data Representation. Frequency-pruned images effectively constitute a novel data representation. Section 8 showed that such representations can achieve a near-VFM accuracy. This points to an exciting future work for novel data representations including various transformations and augmentations, for an even higher accuracy at only a fraction of the cost.

Training-Time vs. Inference-Time Images. Images should follow a consistent data representation during training and inference. For example, if images are compressed using JPEG during training, they should also be compressed with JPEG during inference. Otherwise, the model faces a different distribution during inference than it was trained on, which can cause unpredictable performance [37; 32; 80; 28].

Segmentation & Object Detection. We use image classification, similar to existing studies [13; 47; 89; 91; 93; 106], and perform it across a wide range of domains, models, and configurations. It is not clear whether texture-driven segmentation and object detection tasks would also suffer from a similar type of shortcut. Frequency shortcuts analysis for texture-driven segmentation and object detection tasks and architectures is future work and beyond the scope of our work.

11 Related Work

Prior work primarily studies standard, shape-driven benchmarks and their OOD generalization performance. We focus on texture-driven domains and their ID and OOD generalization performance.

Spectral Bias. Theoretical works show that neural networks learn low-frequency functions first due to gradient-descent's simplicity bias and ReLU smoothness [22; 31; 51; 64; 67; 70; 72; 81; 98; 99; 105]. Empirical works show that HFCs can be more informative than LFCs for CIFAR-10. We show that, within the lower half of the spectrum, both LFCs and HFCs contribute to CIFAR-10. This is consistent with prior works, which define HFCs only broadly. A detailed comparison is in Appendix N.

Texture/Shape-Bias. Texture/shape-bias studies show that texture/shape perturbations can both be detrimental to the model's performance [21; 16; 34; 35; 36; 79; 83]. Our work is orthogonal to these studies. Identifying which frequency components define what level of texture, and studying texture/shape perturbations for texture-driven domains are exciting future directions.

Shortcut Learning. To mitigate shortcuts, existing approaches [92] include feature-space control [50; 58; 64; 109], data augmentation [35; 53; 56; 101; 106], layer-wise penalization [20; 88]. Our work is orthogonal to these studies, as it takes the first step and performs an extensive experimental analysis to identify a new shortcut for a new class of domains. Proposing a novel algorithm to optimally mitigate low-frequency shortcuts for texture-driven domains is an exciting future work.

12 Conclusion

We show that texture-driven domains suffer from low-frequency shortcuts, where a small number of low-frequency components (LFCs) dominate the model's decisions with a skewed spectral behavior, despite that higher-frequency components (HFCs) have a higher predictive power. Pruning LFCs shifts the spectral behavior towards higher frequencies, which in turn provides a significantly improved ID and OOD accuracy, by up to 10% and 40%, respectively. We show that low-frequency shortcuts persist across different model sizes, architectures, and hyperparameters, under algorithmic as well as real-world OOD conditions, for various pretrained foundation models, as well as different image resolutions, frequency-transformation block sizes, and color spaces. Our study contains valuable insights that we hope will be useful to practitioners working in new, understudied domains.

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing. We have manually reviewed all AI-proposed text and modified it to make the flow smooth and succinct. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide our codebase along with a readme file explaining how to reproduce our results as a supplementary material. We provide the codebase also in the following anonymized repository: code repository.

REFERENCES

[1] EuroSAT GitHub Repo. https://github.com/phelber/EuroSAT, 2019.

[2] Galaxy MNIST GitHub Repo. https://github.com/mwalmsley/galaxy_mnist, 2022.

[3] DINOv2: Learning Robust Visual Features without Supervision, 2023.

[4] GenImages GitHub Repo. https://github.com/GenImage-Dataset/GenImage, 2023.

[5] NICO++ GitHub Repo. https://github.com/xxgege/NICO-plus, 2023.

[6] TinyGenImages Kaggle Repo. https://www.kaggle.com/datasets/yangsangtai/tiny-genimage, 2024.

[7] CLIP GitHub Repo. https://github.com/openai/CLIP, 2025.

[8] CONCH V1.5 Model. https://huggingface.co/MahmoodLab/conchv1_5, 2025.

[9] DinoV2 GitHub Repo. https://github.com/facebookresearch/dinov2, 2025.

[10] DinoV3 GitHub Repo. https://github.com/facebookresearch/dinov3, 2025.

[11] Galaxy Zoo. https://github.com/mwalmsley/galaxy-datasets, 2025.

[12] SPIDER Colorectal Model. https://huggingface.co/histai/SPIDER-colorectal-model, 2025.

[13] Antonio A. Abello, Roberto Hirata, and Zhangyang Wang. Dissecting the High-Frequency Bias in Convolutional Neural Networks. In *CVPRW*, pp. 863–871, 2021.

[14] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete Cosine Transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.

[15] Jiawang Bai, Li Yuan, Shu-Tao Xia, Shuicheng Yan, Zhifeng Li, and Wei Liu. Improving Vision Transformers by Revisiting High-Frequency Components. In *ECCV*, pp. 1–18, 2022.

[16] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J. Kellman. Deep Convolutional Networks do not Classify based on Global Object Shape. *PLOS Computational Biology*, 14 (12):1–43, 2018.

[17] Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna Nemani. DeepSat: A Learning Framework for Satellite Imagery. In *SIGSPATIAL*, 2015.

[18] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. In *CVPR*, 2017.

[19] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in Terra Incognita. In *ECCV*, 2018.

[20] Christopher Boland, Keith A Goatman, Sotirios A. Tsaftaris, and Sonia Dahdouh. There Are No Shortcuts to Anywhere Worth Going: Identifying Shortcuts in Deep Learning Models for Medical Image Analysis. In *International Conference on Medical Imaging with Deep Learning*, volume 250, pp. 131–150, 2024.

[21] Tom Burgert, Oliver Stoll, Paolo Rota, and Begüm Demir. ImageNet-trained CNNs are not Biased Towards Texture: Revisiting Feature Reliance Through Controlled Suppression. In *NeurIPS*, 2025.

[22] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards Understanding the Spectral Bias of Deep Learning. In *IJCAI*, pp. 2205–2211, 8 2021.

[23] Jaouad Dabounou and Amine Baazzouz. Enhancing Neural Network Interpretability Through Conductance-Based Information Plane Analysis, 2024.

[24] Nikolay Dagaev, Brett D. Roads, Xiaoliang Luo, Daniel N. Barry, Kaustubh R. Patil, and Bradley C. Love. A Too-Good-to-Be-True Prior to Reduce Shortcut Reliance. *Pattern Recognition Letters*, 166:164–171, 2023.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-scale Hierarchical Image Database. In *CVPR*, pp. 248–255, 2009.

[26] Tong Ding, Sophia J. Wagner, Andrew H. Song, Richard J. Chen, Ming Y. Lu, Andrew Zhang, Anurag J. Vaidya, Guillaume Jaume, Muhammad Shaban, Ahrong Kim, Drew F. K. Williamson, Harry Robertson, Bowen Chen, Cristina Almagro-Pérez, Paul Doucet, Sharifa Sahai, Chengkuan Chen, Christina S. Chen, Daisuke Komura, Akihiro Kawabe, Mieko Ochi, Shinya Sato, Tomoyuki Yokose, Yohei Miyagi, Shumpei Ishikawa, Georg Gerber, Tingying Peng, Long Phi Le, and Faisal Mahmood. A Multimodal Whole-Slide Foundation Model for Pathology. *Nature Medicine*, 31:3749–3761, 2025.

[27] Tuan Do, Bernie Boscoe, Evan Jones, Yun Qi Li, and Kevin Alfaro. GalaxiesML: A Dataset of Galaxy Images, Photometry, Redshifts, and Structural Parameters for Machine Learning. 2024.

[28] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression Artifacts Reduction by a Deep Convolutional Network. In *ICCV*, pp. 576–584, 2015.

[29] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021.

[30] Lingyu Du and Guohao Lan. FreeGaze: Resource-efficient Gaze Estimation via Frequency Domain Contrastive Learning. *CoRR*, abs/2209.06692, 2022.

[31] Adam Dziedzic, John Paparrizos, Sanjay Krishnan, Aaron Elmore, and Michael Franklin. Band-limited Training and Inference for Convolutional Neural Networks. In *ICML*, pp. 1745–1754, 2019.

[32] Max Ehrlich and Larry S. Davis. Deep Residual Learning in the JPEG Transform Domain. In *ICCV*, pp. 3484–3493, 2019.

[33] Dan Fu and Gabriel Guimaraes. Using Compression to Speed Up Image Classification in Artificial Neural Networks. 2016. URL `https://www.danfu.org/files/CompressionImageClassification.pdf`.

[34] Paul Gavrikov and Janis Keuper. Can Biases in ImageNet Models Explain Generalization? In *CVPR*, pp. 22184–22194, 2024.

[35] 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 *ICLR*, 2019.

[36] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut Learning in Deep Neural Networks. *Nature Machine Intelligence*, 2:665–673, 2020.

[37] Farhad Ghazvinian Zanjani, Svitlana Zinger, Bastian Piepers, Saeed Mahmoudpour, Peter Schelkens, and Peter H. N. de With. Impact of JPEG 2000 Compression on Deep Convolutional Neural Networks for Metastatic Cancer Detection in Histopathological Images. *Journal of Medical Imaging*, 6(2), 2019.

[38] Tianrui Guan, Divya Kothandaraman, Rohan Chandra, Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, and Dinesh Manocha. GA-Nav: Efficient Terrain Segmentation for Robot Navigation in Unstructured Outdoor Environments. *IEEE Robotics and Automation Letters*, 7(3): 8138–8145, 2022.

[39] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S. Davis. VITON: An Image-Based Virtual Try-On Network. In *CVPR*, pp. 7543–7552, 2018.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pp. 770–778, 2016.

[41] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 204–207. IEEE, 2018.

[42] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[43] Dan Hendrycks and Thomas G. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *ICLR*, 2019.

[44] HistAI. SPIDER-colorectal dataset. `https://huggingface.co/datasets/histai/SPIDER-colorectal`, 2025. Accessed: 2026-01-07.

[45] HMB302. Inflammation. `https://hmb302.ca/chapters/inflammation/`, 2023. Online histology and pathology educational resource. Accessed: 2026-01-07.

[46] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc V. Le, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, and Yukun Zhu. Searching for MobileNetV3. In *ICCV*, pp. 1314–1324, 2019.

[47] Jason Jo and Yoshua Bengio. Measuring the Tendency of CNNs to Learn Surface Statistical Regularities, 2017.

[48] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

[49] Kirsi Laitala and Casper Boks. Sustainable Clothing Design: Use Matters. *Journal of Design Research*, 10(1–2):121–139, 2012.

[50] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking Clever Hans Predictors and Assessing What Machines Really Learn. *Nature Communications*, 10(1), 2019.

[51] Zhiyu Lin, Yifei Gao, and Jitao Sang. Investigating and Explaining the Frequency Bias in Image Classification. In *IJCAI*, pp. 717–723, 2022.

[52] Shao-Yuan Lo and Hsueh-Ming Hang. Exploring Semantic Segmentation on the DCT Representation. In *1st ACM International Conference on Multimedia in Asia (MMASIA)*, pp. 1–6, 2019.

[53] Matthias Minderer, Olivier Bachem, Neil Houlsby, and Michael Tschannen. Automatic Shortcut Removal for Self-supervised Representation Learning. In *ICML*, 2020.

[54] Jayanta Mukhopadhyay. *Image and Video Processing in the Compressed Domain*. Chapman and Hall/CRC, 2011.

[55] Subramanian Senthilkannan Muthu. *Circular Economy in Textiles and Apparel: Processing, Manufacturing, and Design*. Woodhead Publishing, 2018.

[56] Meike Nauta, Robert Walsh, Andrew Dubowski, and Christin Seifert. Uncovering and Correcting Shortcut Learning in Machine Learning Models for Skin Cancer Diagnosis. *Diagnostics*, 12(1), 2022.

[57] Dmitry Nechaev, Alexey Pchelnikov, and Ekaterina Ivanova. SPIDER: A Comprehensive Multi-Organ Supervised Pathology Dataset and Baseline Models, 2025.

[58] Hongjing Niu, Hanting Li, Feng Zhao, and Bin Li. Roadblocks for Temporarily Disabling Shortcuts and Learning New Knowledge. In *NeurIPS*, pp. 29064–29075, 2022.

[59] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards Universal Fake Image Detectors That Generalize Across Generative Models. In *CVPR*, pp. 24480–24489, 2023.

[60] Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Fast Vision Transformers with HiLo Attention. In *NeurIPS*, pp. 14541–14554, 2022.

[61] Nicolas Papernot and Patrick McDaniel. Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning, 2018.

[62] Jeongsoo Park and Andrew Owens. Community Forensics: Using Thousands of Generators to Train Fake Image Detectors. In *CVPR*, pp. 8245–8257, 2025.

[63] Namuk Park and Songkuk Kim. How Do Vision Transformers Work? In *ICLR*, 2022.

[64] Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient Starvation: A Learning Proclivity in Neural Networks. In *NeurIPS*, volume 34, pp. 1256–1272, 2021.

[65] PyTorch. *PyTorch — ResNet-50 Model Documentation*, 2025. URL `https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html`. Accessed: 2026-01-07.

[66] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, pp. 8748–8763, 2021.

[67] 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 Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *PMLR*, volume 97, pp. 5301–5310, 2019.

[68] Vikram V. Ramaswamy, Sunnie S. Y. Kim, Ruth Fong, and Olga Russakovsky. Overlooked Factors in Concept-Based Explanations: Dataset Choice, Concept Learnability, and Human Capability. In *CVPR*, pp. 10932–10941, 2023.

[69] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global Filter Networks for Image Classification. In *NeurIPS*, pp. 980–993, 2021.

[70] Basri Ronen, David Jacobs, Yoni Kasten, and Shira Kritchman. The Convergence Rate of Neural Networks for Learned Functions of Different Frequencies. In *NeurIPS*, volume 32, 2019.

[71] Samuel Felipe dos Santos, Nicu Sebe, and Jurandy Almeida. The Good, The Bad, and The Ugly: Neural Networks Straight From JPEG. In *27th IEEE International Conference on Image Processing (ICIP)*, pp. 1896–1900, 2020.

[72] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The Pitfalls of Simplicity Bias in Neural Networks. In *NeurIPS*, 2020.

[73] Runwu Shi, Shichun Yang, Yuyi Chen, Rui Wang, Jiayi Lu, Zhaowen Pang, and Yaoguang Cao. Road Recognition for Autonomous Vehicles Based on Intelligent Tire and SE-CNN. In *Intelligent Systems and Pattern Recognition*, volume 1589, pp. 291–305. 2022.

[74] Shu Zhong. TextileNet: Material taxonomy-based fashion textile dataset. `https://github.com/hahashu/TextileNet`, 2023. Accessed: 2026-01-07.

[75] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026.

[76] Utku Sirin and Stratos Idreos. The Image Calculator: 10x Faster Image-AI Inference by Replacing JPEG with Self-designing Storage Format. *Proc. ACM Manag. Data*, 2(1), 2024.

[77] Utku Sirin, Victoria Kauffman, Aadit Saluja, Florian Klein, Jeremy Hsu, and Stratos Idreos. Frequency-Store: Scaling Image AI by A Column-Store for Images. In *CIDR*, 2025.

[78] Chetan L. Srinidhi, Ozan Ciga, and Anne L. Martel. Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, 67, 2021.

[79] Ajay Subramanian, Elena Sizikova, Najib J. Majaj, and Denis G. Pelli. Spatial-frequency Channels, Shape Bias, and Adversarial Robustness. In *NeurIPS*, 2023.

[80] Zhun Sun, Mete Ozay, Yan Zhang, Xing Liu, and Takayuki Okatani. Feature Quantization for Defending Against Distortion of Images. In *CVPR*, 2018.

[81] Damien Teney, Armand Mihai Nicolicioiu, Valentin Hartmann, and Ehsan Abbasnejad. Neural Redshift: Random Networks Are Not Random Functions. In *CVPR*, pp. 4786–4796, 2024.

[82] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training Data-Efficient Image Transformers & Distillation Through Attention. In *ICML*, pp. 10347–10357, 2021.

[83] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L. Griffiths. Are Convolutional Neural Networks or Transformers More Like Human Vision? In *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society*, pp. 1844–1850, 2021.

[84] Haiming Tuo, Zuqiang Meng, Zihao Shi, and Daosheng Zhang. Interpretable Neural Network Classification Model Using First-order Logic Rules. *Neurocomputing*, 614(1):128–840, 2025.

[85] Koen van Gelder. E-commerce Worldwide—Statistics & Facts. `https://www.statista.com/topics/871/online-shopping/`, 2025. Accessed: 2026-01-07.

[86] G.K. Wallace. The JPEG Still Picture Compression Standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.

[87] Mike Walmsley, Chris Lintott, Tobias Géron, Sandor Kruk, Coleman Krawczyk, Kyle W Willett, Steven Bamford, Lee S Kelvin, Lucy Fortson, Yarin Gal, William Keel, Karen L Masters, Vihang Mehta, Brooke D Simmons, Rebecca Smethurst, Lewis Smith, Elisabeth M Baeten, and Christine Macmillan. Galaxy Zoo DECaLS: Detailed Visual Morphology Measurements from Volunteers and Deep Learning for 314000 Galaxies. *Monthly Notices of the Royal Astronomical Society*, 509(3):3966–3988, 2022.

[88] Haohan Wang, Songwei Ge, Zachary C. Lipton, and Eric P. Xing. Learning Robust Global Representations by Penalizing Local Predictive Power. In *NeurIPS*, pp. 10506–10518, 2019.

[89] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. High-Frequency Component Helps Explain the Generalization of Convolutional Neural Networks. In *CVPR*, pp. 8681–8691, 2020.

[90] Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-Oversmoothing in Deep Vision Transformers via the Fourier Domain Analysis: From Theory to Practice. In *ICLR*, 2022.

[91] Shunxin Wang, Raymond Veldhuis, Christoph Brune, and Nicola Strisciuglio. What Do Neural Networks Learn in Image Classification? A Frequency Shortcut Perspective. In *ICCV*, pp. 1433–1442, 2023.

[92] Shunxin Wang, Raymond Veldhuis, Christoph Brune, and Nicola Strisciuglio. A Survey on the Robustness of Computer Vision Models against Common Corruptions, 2024.

[93] Shunxin Wang, Raymond Veldhuis, and Nicola Strisciuglio. Do ImageNet-trained Models Learn Shortcuts? The Impact of Frequency Shortcuts on Generalization. In *CVPR*, pp. 25198–25207, 2025.

[94] Zhenyu Wang, Hao Luo, Pichao WANG, Feng Ding, Fan Wang, and Hao Li. VTC-LFC: Vision Transformer Compression with Low-Frequency Components. In *NeurIPS*, pp. 13974–13988, 2022.

[95] Hongming Xu, Qi Xu, Fengyu Cong, Jeonghyun Kang, Chu Han, Zaiyi Liu, Anant Madabhushi, and Cheng Lu. Vision Transformers for Computational Histopathology. *IEEE Reviews in Biomedical Engineering*, 17:63–79, 2024.

[96] Kai Xu. *Learning in Compressed Domains*. PhD thesis, Arizona State University, 2021.

[97] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the Frequency Domain. In *CVPR*, pp. 1740–1749, 2020.

[98] Zhi-Qin John Xu. Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks. *Communications in Computational Physics*, 28(5):1746–1767, 2020.

[99] Zhiqin John Xu and Hanxu Zhou. Deep Frequency Principle Towards Understanding Why Deeper Learning Is Faster. In *AAAI*, volume 35, pp. 10541–10550, 2021.

[100] Zhiqin John Xu and Hanxu Zhou. Deep Frequency Principle Towards Understanding Why Deeper Learning Is Faster. *AAAI*, 35(12):10541–10550, 2021.

[101] Zhuo Xu, Xiang Xiang, and Yifan Liang. Overcoming Shortcut Problem in VLM for Robust Out-of-Distribution Detection. In *CVPR*, 2025.

[102] Jia Xue, Hang Zhang, Kristin Dana, and Ko Nishino. Differential Angular Imaging for Material Recognition. In *CVPR*, pp. 764–773, 2017.

[103] Dongsheng Yang, Dongmin Zhang, Yi Yuan, Zhaoyu Lei, Binlei Ding, and Bo Lei. Road Terrain Recognition Based on Tire Noise for Autonomous Vehicle. *Scientific Reports*, 14(1), 2024.

[104] Yi Yang and Shawn Newsam. Bag-of-Visual-Words and Spatial Extensions for Land-Use Classification. In *SIGSPATIAL*, GIS, pp. 270–279, 2010.

[105] Ping yeh Chiang, Renkun Ni, David Yu Miller, Arpit Bansal, Jonas Geiping, Micah Goldblum, and Tom Goldstein. Loss Landscapes are All You Need: Neural Network Generalization Can Be Explained Without the Implicit Bias of Gradient Descent. In *ICLR*, 2023.

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

[107] Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. NICO++: Towards Better Benchmarking for Domain Generalization. In *CVPR*, pp. 16036–16047, June 2023.

[108] Chenxiao Zhao, P. Thomas Fletcher, Mixue Yu, Yaxin Peng, Guixu Zhang, and Chaomin Shen. The Adversarial Attack and Detection under the Fisher Information Metric. *AAAI*, 33 (1):5869–5876, 2019.

[109] Jiayun Zheng and Maggie Makar. Causally Motivated Multishortcut Identification and Removal. In *NeurIPS*, pp. 12800–12812, 2022.

[110] Shu Zhong, Miriam Ribul, Youngjun Cho, and Marianna Obrist. TextileNet: A Material Taxonomy-based Fashion Textile Dataset, 2023.

[111] Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. GenImage: A Million-Scale Benchmark for Detecting AI-Generated Image. In *NeurIPS*, 2023.

[112] Jannik Zürn, Wolfram Burgard, and Abhinav Valada. Self-Supervised Visual Terrain Classification from Unsupervised Acoustic Feature Learning. *IEEE Transactions on Robotics (T-RO)*, 37(2):466–481, 2021.

Table 1: Datasets.

Dataset	#Classes	Train / Val / Test	Input Size
CIFAR-10	10	45K / 5K / 10K	32×32
SP-Colorectal	14	56K / 8K / 13K	224×224
TextileNet	27	245K / 35K / 70K	224×224
GTOS	40	24K / 2K / 4K	224×224
TinyGenImages	2	24K / 7K / 4K	224×224
GalaxyMNIST	4	8K / 1K / 1K	224×224
EuroSAT	10	19K / 2.7K / 5.4K	64×64
NICO++	60	64.1K / 17.8K / 7.1K	224×224

A Additional Experimental Details

Datasets. We use one image classification dataset for each domain we described in Section 3. Table 1 presents the summary of each dataset. For histopathology, we use the recently proposed multi-organ SPIDER dataset [57]. We use the colorectal organ type, due to its widespread adoption [44]. It is a large-scale dataset with ∼77K images and 14 classes. We use only the central patches. We use the original training-test set split with 80%-20% ratio. We split the training set further into 70% training and 30% validation. We name this dataset as **SP-Colorectal** in our figures. For the textile classification dataset, we use the recently proposed **TextileNet** dataset [110]. We use its fabric subset, which contains 27 classes and ∼350K samples [74]. We use the proposed 70%-10%-20% training-validation-test set splits [110]. For ground terrain recognition, we use the recent ground terrain dataset for outdoor scenes, **GTOS** [102]. It is a large-scale dataset with ∼30K images and 40 classes. We use the first of its five original 70%-30% training-test splits. We further split the test set into validation and test sets by 10%-20%. For AI-generated images, we use **GenImages** and **TinyGenImages** datasets [4; 6; 111]. It includes original images taken from the ImageNet dataset [25], and their corresponding fake images generated by eight different generative models. TinyGenImages is a uniformly sampled version of GenImages. GenImages includes one million generated and one million original images. TinyGenImages includes 17.5K generated and 17.5K original images. We use the original training and test split (28K / 7K), where we split the training set further into training and validation sets of 85% to 15% (size 24K and 4K). We use the basic **Galaxy MNIST** dataset with 10K samples, 4 classes [2] among others [11; 27] due to its convenience. We use the original train/test split, and further split the training set into train/val split by 90-10%. We use their 224x224 version (GalaxyMNISTHighrez). We use **EuroSAT** dataset for satellite images with 27K samples and 10 classes covering shape- and texture-driven categories [1]. We randomly split the dataset into 70-10-20% train/val/test splits, as there is no original train/val/test splits. As the standard benchmark, we use **CIFAR-10** [48] due to its widespread adoption in frequency characterization studies [13; 47; 89]. It has 50K training and 10K test images over 10 classes. We split the training set into 45K-5K training-validation sets. **NICO++** is a domain-generalization dataset, where there are 60 object categories taken in six different domains, such as rock, water, autumn, dim, water, and grass[8]. It has 89K samples. We use the original training and test splits of 80-20%, which then we split the training set into further training and validation of 90-10% [5; 107].

Foundation Models. For DinoV2, DinoV3, and CLIP, we train a linear layer (768, 1280, and 512 units, respectively) on top of their frozen backbones. We use ViT-B/14-Reg (86M params), ViT-H+/16 (840M params) and ViT-B/32 (151M params) backbones, respectively. For CONCH V1.5 and Hibou-L, we train a linear layer (768 and 128 units, respectively) on top of their frozen backbones. We use ViT-L/16 (307 params) and ViT-L/14 (300M params) backbones, respectively.

Training. We train with SGD (momentum 0.9, weight decay 5×10^{-4}), a cosine annealing schedule with initial learning rate 0.1, and batch size 128 for 200 epochs, selecting the checkpoint with highest validation accuracy for ResNet-50 and MobileNet-V3. Following [29; 82], we use AdamW (weight decay=0.3) with initial learning rate 3e-3, 300 epochs of training, 3.4×(number of batches) warmup steps, using linear scheduler for the warmup (start factor 1e-8, end factor 1.0), and cosine scheduler

[8]While the paper refers to 10 domains with 80 categories each [107], the published dataset contains a subset of it with six domains and 60 categories [5].

for the post-warmup training, batch size of 512. We train a linear head on top of a frozen DinoV2 and CLIP backbones (768 & 512 units, respectively) for 100 epochs with a batch size of 32. We use lr=10^{-4}, weight decay=10^{-4} for DinoV2, and lr 10^{-3} and weight decay 10^{-4} for CLIP. We use random-resized-crop and horizontal-flip augmentations for DinoV2, and no data augmentation for CLIP, following the original code bases for linear probing [9; 7] We use seed values of 42, 123, and 999. We make sure we achieve, for each benchmark, an accuracy almost as high as in their original papers. This is 89% for SP-Colorectal, 66% for TextileNet, 77% for GTOS, and 94% for CIFAR-10. We use training-validation-test splits, as explained earlier in this section, and report test-set accuracies in Appendix B. Validation-test set splits, and their accordance are important to keep relevance to image compression studies, which need to perform validation set analysis to decide how many images should be compressed at test time [76; 97]. We normalize each image using the mean and standard deviation of its training set, separately for each pruned version of the training set.

Hardware. We use a cluster of Nvidia A100, H100, and H200 GPUs. We submit jobs as batches and use GPUs as they are available. Each training takes 1-6 hours with a single GPU. Obtaining the spectral behavior requires an absence-test per diagonal, for each diagonal of coefficients. Each such experiment requires one pass over the validation set, which is usually a few minutes on a single GPU. OOD tests are similar to spectral behavior tests, as they also require one pass over the corrupted validation images.

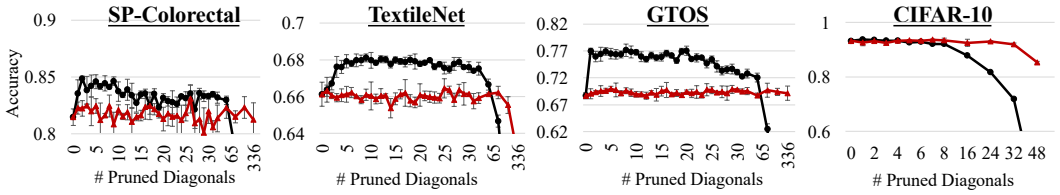


Figure 11: Test set ID accuracy results for pruning LFCs (dark line) and HFCs (red line). Test-set and validation set results are similar. Pruning LFCs improves accuracy for texture-driven tasks by improving representation and reducing frequency shortcuts. LFCs are crucial for CIFAR-10, as it is shape-driven and depends on smooth, high-level structures. Pruning HFCs minimally impacts accuracy, especially for SP-Colorectal and GTOS. TextileNet and CIFAR-10 are more dependent on the higher frequencies at the lower half of the spectrum. Hence, they suffer from observable accuracy losses near the end of HFC pruning.

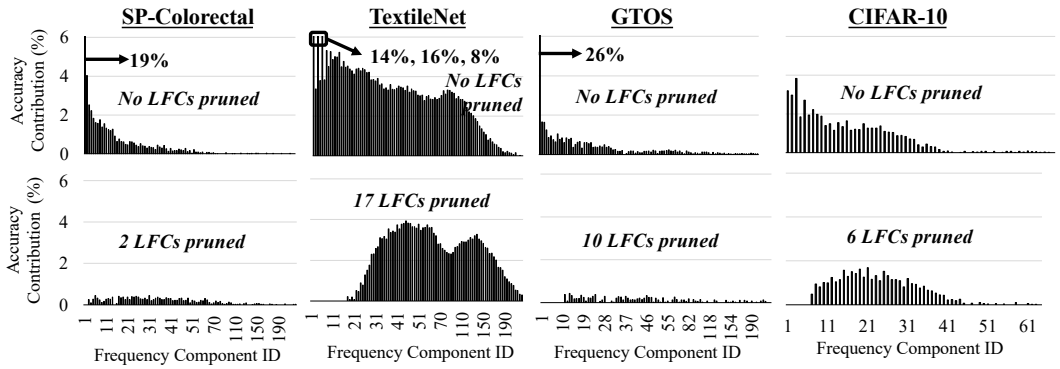


Figure 12: Accuracy contributions based on test-set images. **Top:** Accuracy contributions when models are trained with unpruned images. **Bottom:** Accuracy contributions when models are trained with LFC-pruned images. Results follow similar patterns to the validation-set results. Unpruned training causes a skewed behavior for texture-driven tasks, indicating a low-frequency shortcut. LFC-pruned training eliminates the shortcut and improves representation capacity. CIFAR-10 benefits from a range of components from the lower end of the frequency spectrum.

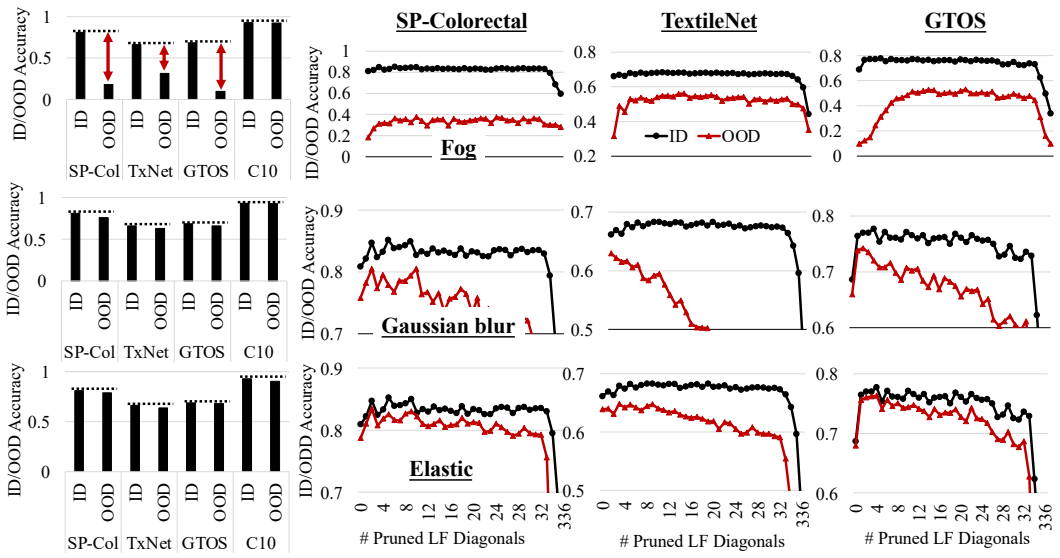


Figure 13: OOD corruption results based on test-set images. **Top:** Low-frequency corruption (fog) results. **Middle:** High-frequency corruption (Gaussian blur) results. **Bottom:** Mixed-frequency corruption (elastic transform) results. Results are similar to those of validation-set images. Fog heavily impacts texture-driven tasks due to the low-frequency shortcuts. Pruning LFCs significantly improves OOD accuracy for fog corruption. Pruning LFCs introduces a trade-off under Gaussian blur and elastic transform corruptions: OOD accuracy improves due to HFCs' improved representation, but decreases due to magnified corruptions at the HFCs. Final accuracy depends on which factor has a greater impact.

B Test-Set Results

This section presents test set results for ID performance, accuracy contributions, and OOD performance. Figure 11 presents test set results for ID performance. As can be seen, results follow similar trends to the validation set accuracies. Texture-driven tasks, SP-Colorectal, TextileNet, and GTOS suffer from low-frequency shortcuts. Pruning LFCs eliminates the shortcut and improves ID generalization performance. CIFAR-10 is shape-driven. As a result, LFCs, defining smooth local structures, are more important than HFCs. HFCs are promising candidates for image compression, as pruning up to 87.5% of them results in no loss of accuracy.

Figure 12 presents accuracy contributions based on test set images. As shown, the results are similar to those of the validation set. All texture-driven domains suffer from low-frequency shortcuts (top row), with a skewed distribution towards LFCs (top row). When models are trained on pruned images (bottom row), the spectral behavior of the texture-driven tasks shifts towards higher frequencies, with a more balanced distribution. CIFAR-10 has a significantly more balanced distribution than texture-driven tasks. CIFAR-10's distribution also shifts toward higher frequencies, although largely remaining within the lower half of the spectrum.

Figure 13 presents test set results for OOD corruptions. The results closely follow the validation-set results. Low-frequency corruptions (fog) heavily impact texture-driven domains, due to the low-frequency shortcuts. Pruning LFCs eliminates the shortcut, resulting in significantly higher OOD accuracy against fog corruption. Nevertheless, the maximum OOD accuracy is significantly lower than the maximum ID accuracy, due to the magnified corruption at HFCs. Fog, despite being low-frequency-heavy, also corrupts HFCs. When models are trained with LFC-pruned images, they rely more on HFCs and LFCs. As a result, they become more sensitive to high-frequency corruptions, and fog's corruptions at the HFCs are magnified. As a result, final OOD accuracy is significantly lower than the ID accuracy.

Gaussian blur and elastic transform minimally affect ID accuracy, as their corruption at the LFCs is not significant. As LFCs are pruned, models latch onto higher frequencies that better represent

application semantics, while becoming increasingly vulnerable to high-frequency corruptions. Hence, OOD accuracy increases, stays stable, or decreases depending on whether the improved representation or magnified corruption is more significant.

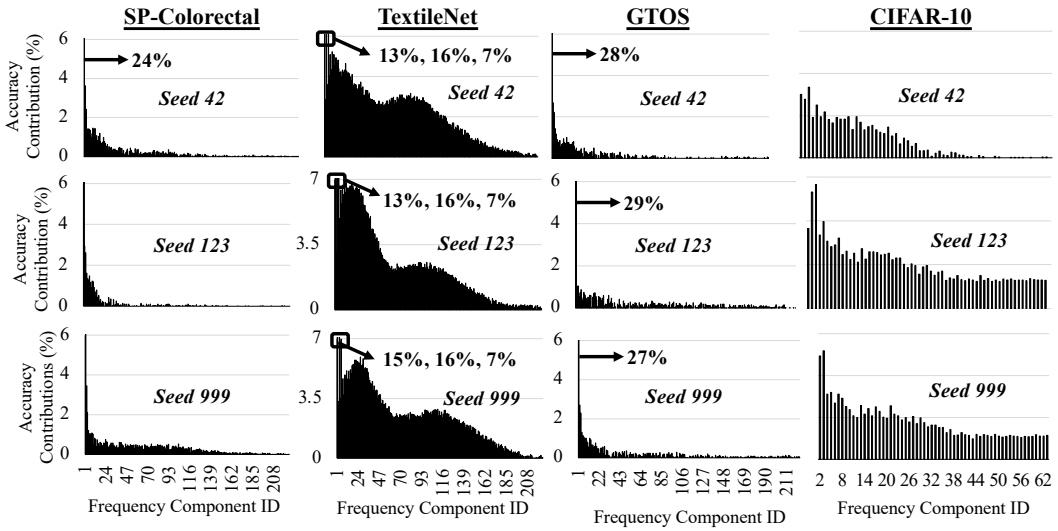


Figure 14: Spectral behavior for ResNet-50 across different seeds when using unpruned images. As can be seen, similar trends are observed across different seeds.

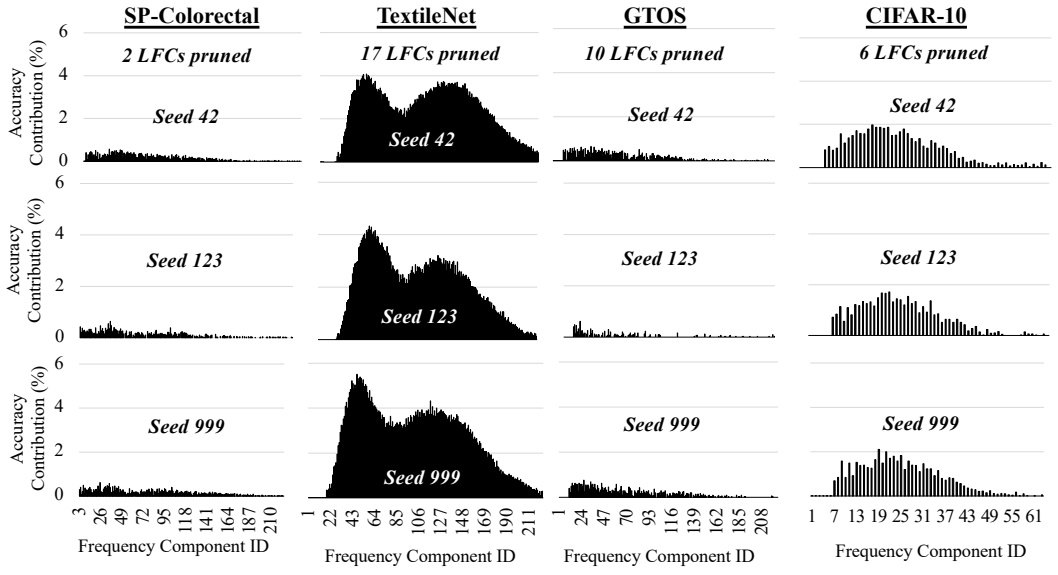


Figure 15: Spectral behavior for ResNet-50 across different seeds when using pruned images. As can be seen, similar trends are observed across different seeds.

C Spectral Behavior Across Different Seeds

We report spectral behavior, i.e., the accuracy contributions for ResNet-50 when trained with unpruned images for SP-Colorectal, TextileNet, GTOS, and CIFAR-10 in Figure 14. As can be seen, similar trends are observed across different seeds. Similarly, Figure 15 presents the spectral behavior for pruned images, where we prune as many LFCs as each model achieves its maximum ID accuracy. Once again, the figure shows that the trends holds true across different repetitions. This shows that low-frequency shortcuts are persistent across different seeds and repetitions.

D THEORETICAL DISCUSSION

Our results closely follow the theoretical results presented by Shah et al. 2020 [72], Pezeshki et al. 2021 [64], and Chiang et al. 2023 [105].

Shah et al. show that neural networks are provably biased towards learning simpler decision boundaries (see their Figure 1, green line) than complex decision boundaries (see their Figure 1, orange line). The authors prove that the gradients for the simpler decision boundary are consistently larger than those for the complex decision boundary (see Section 4.1, Theorem 1). The proof stems from the closed-form gradient expressions for the simple and complex decision boundaries given by Lemmas 4 and 5 in Appendix F.2, respectively.

The first lines of the proof of each lemma have the $y_j x_{1j}$ and $y_j x_{2j}$ terms in them, which is the reason that the two gradients have different final expressions. y stands for the label, x_{1j} stands for the feature value providing a simple decision boundary for sample j, and x_{2j} stands for the feature value providing a complex decision boundary for sample j. $y_j x_{1j}$ always produces the same result for all the samples (1 in the specific scenario), whereas $y_j x_{2j}$ produces different results depending on the feature and label values (1, -1, or 0 in the specific scenario). This results in constant increases in the gradients of the parameters for the simple feature ($\nabla_{w_{1i}}$, where i is the neuron index), whereas gradients of the parameters for the complex feature ($\nabla_{w_{2j}}$ fluctuate (in a one-layer neural network). As a result, parameter values of the simple feature (Linear Coordinate in Theorem 1) are constantly bigger than the parameter values for the complex feature (3-Slab Coordinate in Theorem 1). The neural network then relies on the feature that can be used with a simpler decision boundary to make most of its decisions. The authors show that simple features are chosen over complex features, even when simple features have a less predictive power, indicating the severity of the simplicity bias (see their Section 5).

Pezeshki et al. introduce the concept of Gradient Starvation that supports the findings of Shah et al. The authors use a simplified Neural Tangent Kernel regime, where the output of the network can be approximated as a linear function. The authors use SVD to decompose the label-feature matrix ($Y\Phi_0$) into its principal components and define the strength of a feature by its corresponding eigenvalue. Features that are highly aligned with the labels (e.g., the horizontal feature in their Figure 1) have large eigenvalues, and vice versa.

The authors show that if a feature has a high strength, i.e., is highly correlated with the label, i.e., simple in Shah et al.'s definition, the strength of the feature decreases the response of the network to the other feature (see Theorem 2, Eq. 16 ($\frac{dz_2^*}{ds_1^2} < 0$) where z_2^* is the optimal response of the network to feature 2, and s_1^2 is the square of the strength of feature 1), which the authors define as the Gradient Starvation (although they do not explicitly show that gradients diminish over the iterations).

Finally, Chiang et al. observe that simple solutions are favored by non-gradient-based optimizers as well and attribute this to the volume hypothesis, where simple solutions occupy orders of magnitude more volume in the loss landscape than complex solutions. The authors show that the simple solution has a volume that is 6 orders of magnitude larger than the complex solution for the same example Shah et al. uses (see Chiang et al.'s Section 6 & Figure 3).

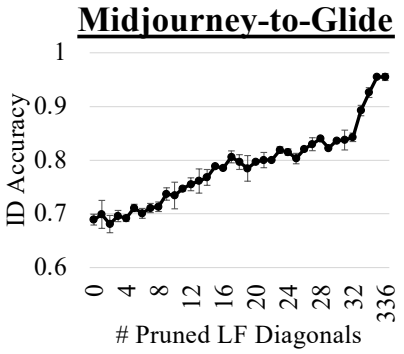


Figure 16: ID results for a subset of original GenImages dataset.

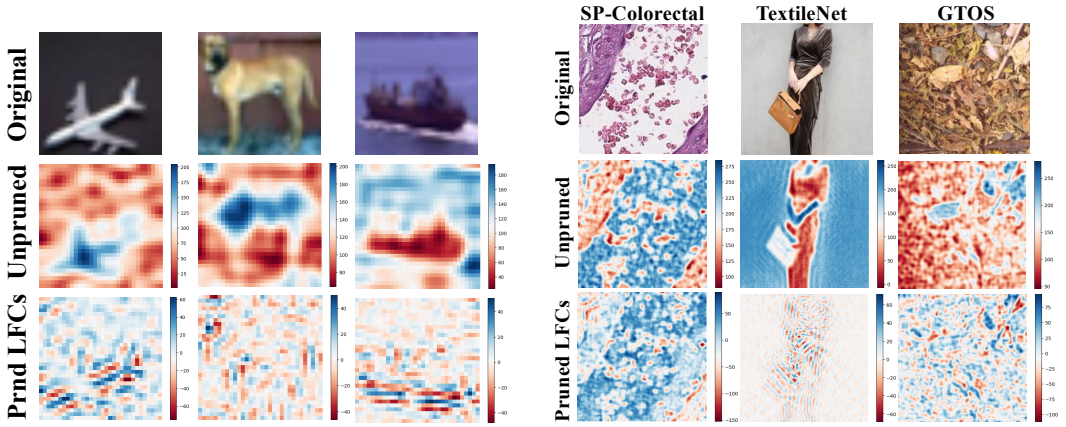
These theoretical results explain what we empirically observe at scale: neural networks learn from a few LFCs rather than a large number of HFCs in texture-driven domains, despite the fact that HFCs have greater predictive power. We observe that neural networks either (i) learn from a few LFCs or (ii) a large number of HFCs. It is never that the network learns from a small number of HFCs. This shows that LFCs provide a simple set of features that are highly correlated with the labels, similar to the simple features that Shah et al. studied in their examples. Once the network starts learning from a few LFCs, the gradients of the LFCs starve the gradients of the HFCs, and eventually, the LFCs dominate the learning process. When we prune the LFCs from

the training set, the network can learn from a complex set of HFCs, as there is no other competitor now.

E Additional Results on AI-Generated Images

In addition to the TinyGenImages results in Section 5, we analyze a specific subset of original GenImages [4; 111] to make sure TinyGenImages' sampled images does not introduce any misleading outcome for our analyses. We observe that the original GenImages dataset contains ∼160K generated images for each generative model, across eight different generative models, such as BigGAN, Midjourney, and Glide. Table 2 of [111] shows that main challenge of the GenImages dataset is to accurately predict images generated by two different generative models. Hence, we randomly choose two different generative models and analyze its ID accuracy behavior as we prune LFCs. We choose Midjourney and Glide. We train the model (ResNet-50) using Midjourney-generated images and test it with Glide-generated images.

Figure 16 presents the results. As can be seen, accuracy constantly improves by up to 25%, even more than that of TinyGenImages shown in Figure 4 in Section 5. This shows that our results with TinyGenImages is representative for full GenImages.



(a) CIFAR-10 visualizations. Models trained with unpruned images latch onto the smooth structures that define object shapes well. Models trained with pruned images (6 LFCs) latch onto the more irregular structures. As we prune a larger and larger number of LFCs, shapes become increasingly distorted, and accuracy drops sharply.

(b) Texture-driven tasks' visualizations. Models trained with unpruned images latch onto large blobs and shapes that do not have any meaningful, low-level structure that matches with the application semantics. Models trained with pruned images latch onto fine-grained details that defines the application semantics more faithfully.

Figure 17: Sample images from CIFAR-10 (left) and texture-driven tasks (right). **Top:** Original images. **Middle:** Reconstructed images based on unpruned training. **Bottom:** Reconstructed images based on pruned training.

F Additional Visualizations

This section presents additional visualizations for CIFAR-10 and the texture-driven domains we studied. Figure 17a presents sample images for CIFAR-10. Images are reconstructed by using the top 10 frequency components in terms of their accuracy contributions in Figure 5. The top row presents the original images from the airline, dog, and ship classes, the middle row presents the reconstructed images from unpruned training, and the bottom row presents the reconstructed images from 6-LFC-pruned training. Our goal is to visualize the data characteristics that neural networks learn from when they are trained with unpruned versus pruned images. We use colormaps because we want to focus on the strength of the signal components and their characteristics.

As can be seen, models trained on unpruned images (middle row) latch onto structures with smooth patterns that define the overall shape of the object, which are largely LFCs, as shown in the accuracy

contributions in Figure 5. When models are trained on LFC-pruned images, they latch onto higher frequencies that still include a significant amount of information, but are much more irregular and significantly distort the overall shape of the object. This explains both how accuracy is preserved when LFCs are pruned and how it starts to drop significantly after pruning 6 LFCs. Up to 6 LFCs; while patterns start breaking, they still contain sufficient information to perform classification. As we prune an increasing number of LFCs, the patterns become increasingly distorted, resulting in a significant loss of accuracy.

Figure 17b presents sample images reconstructed from frequency components with the top 50 frequency components in terms of their accuracy contributions shown in Figure 5, for the texture-driven tasks. The top row presents original images; the middle row, images reconstructed from models trained on unpruned images; and the bottom row, images reconstructed from models trained on pruned images. When pruning, we prune as many LFCs as needed to achieve the highest ID accuracy.

As can be seen, models trained on unpruned images (middle row) latch onto low-frequency information that largely defines broad shapes and blobs, with minimal fine-grained structure. On the other hand, models trained on LFC-pruned images (bottom row) latch onto fine-grained, high-frequency structures that are better aligned with the characteristics of the classification tasks. This allows learning from a range of higher-frequency components with a more balanced, unbiased distribution, resulting in better generalization performance.

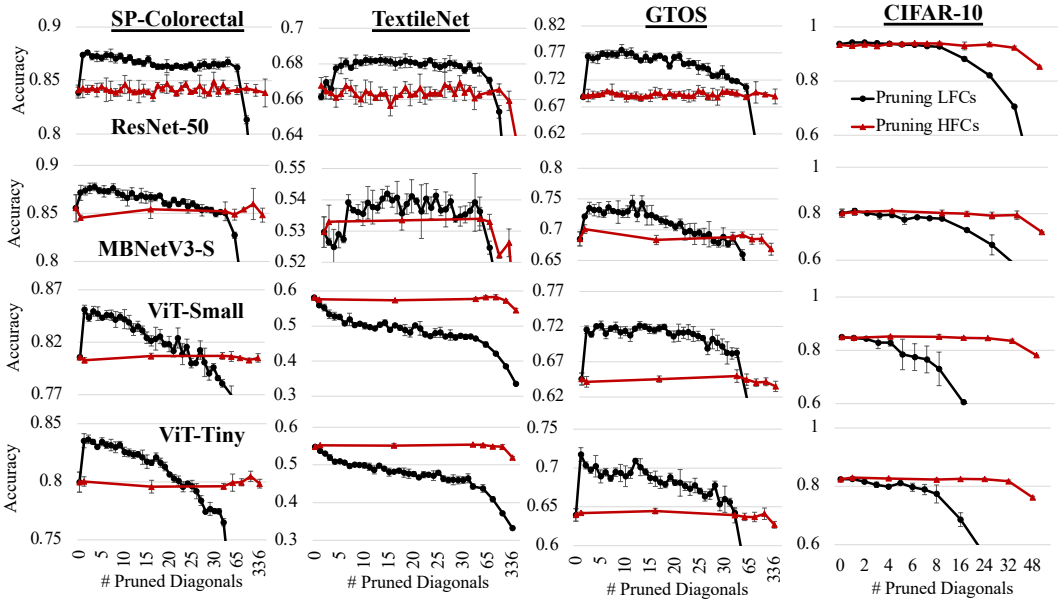


Figure 18: ID Results for ResNet-50, MobileNet-V3, ViT-Small, and ViT-Tony.

G Impact of Model Size and Architecture on ID Results

Figure 18 presents ID accuracy results for pruning LFCs and HFCs for all the four neural architectures we analyzed: ResNet-50, MobileNet-V3, ViT-Small, and ViT-Tiny. Figure 19 and 20 show the spectral behavior for the four models we use. As can be seen, the results are consistent across different neural architectures; pruning HFCs do not significantly impact the ID accuracy, whereas pruning LFCs improves it for all texture-driven tasks, except TextileNet with ViT-Small and ViT-Tiny. Nevertheless, all the models and datasets have a skewed behavior towards the lowest-frequency coefficients.

Decreasing ID results for TextileNet with ViTs is due to two reasons: (i) ViTs have a stronger bias towards LFCs than CNNs, and (ii) TextileNet has a stronger low-frequency noise than SP-Colorectal and GTOS.

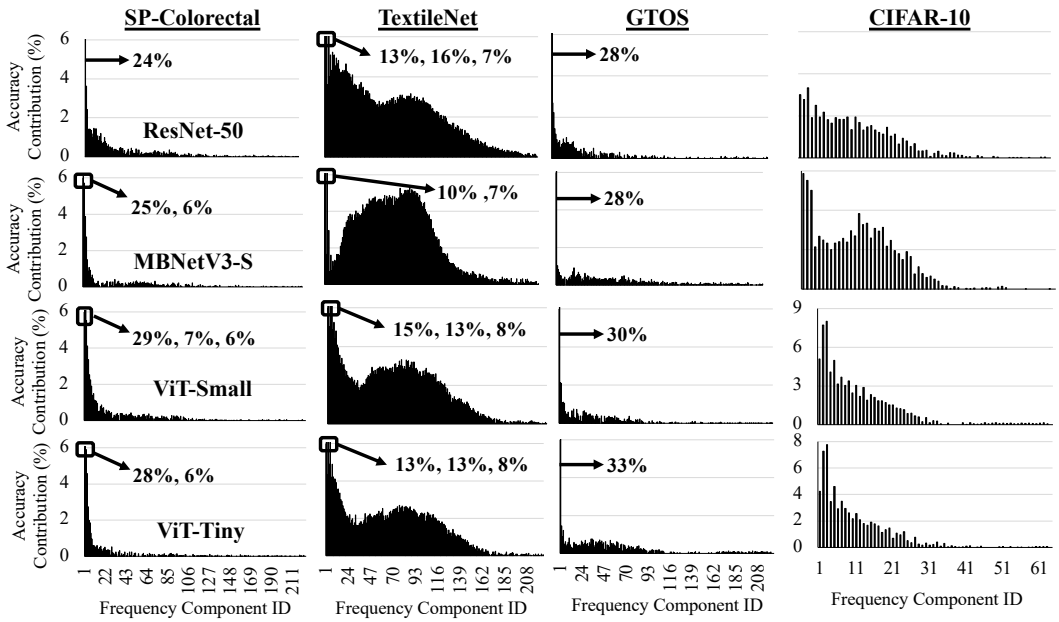


Figure 19: Spectral Behavior for ResNet-50, MobileNet-V3, ViT-Small, and ViT-Tony when trained with unpruned images.

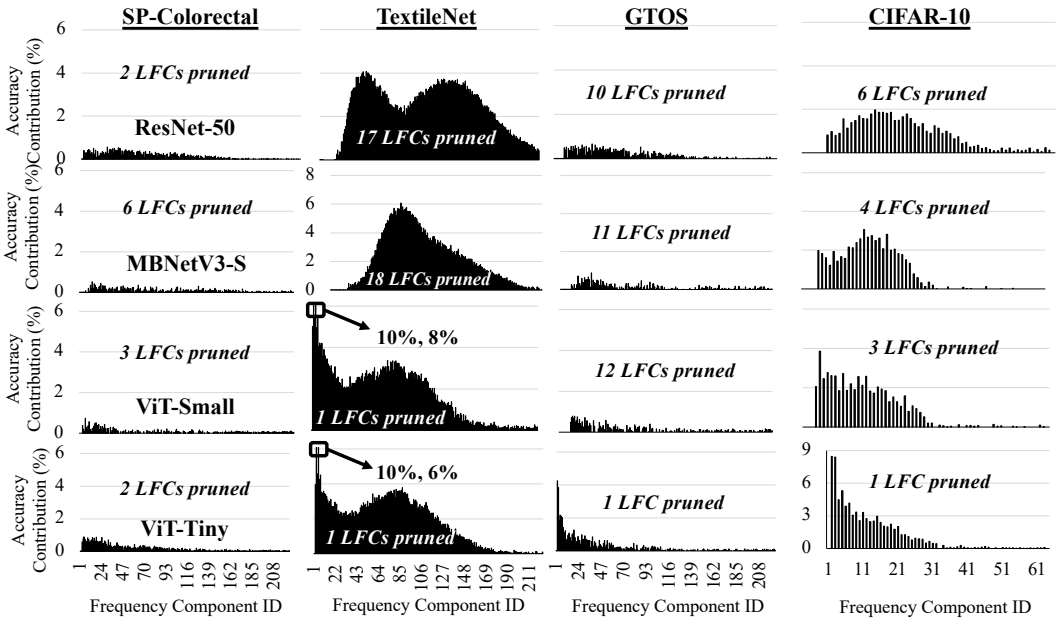


Figure 20: Spectral Behavior for ResNet-50, MobileNet-V3, ViT-Small, and ViT-Tony when trained with pruned images.

Figure 21 presents accuracy contributions for TextileNet with ResNet-50 (top) and ViT-Small (bottom) for an increasing number of pruned LFCs. As shown, pruning LFCs does not immediately eliminate the skewed distribution, i.e., the low-frequency shortcut; whereas, for ResNet-50, where pruning even a single LFC imme-

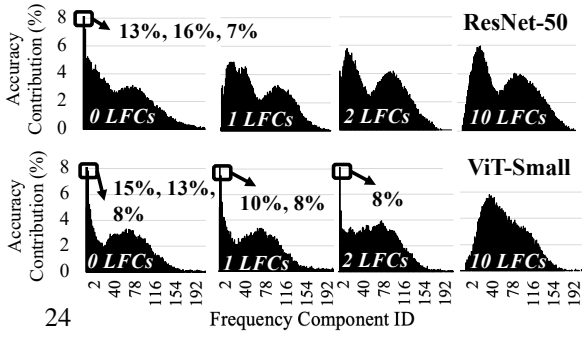


Figure 21: Spectral behavior of TextileNet.

diately eliminates the skewed distribution and shifts the distribution towards HFCs. CNNs use fine-grained strided convolutions over the whole image, which allow them to more easily extract fine-grained, high-frequency information than ViTs, which use disjoint image patches and hence have a stronger bias towards low-frequency noises. There is a recent line of work reporting similar results on ViTs versus CNNs [15; 60; 63; 69; 90; 94].

Secondly, TextileNet images are fashion models wearing garments made from different fabrics (see Figure 3 of our submission). These images contain more severe low-frequency noise, compared to SP-Colorectal and GTOS, e.g., in the photographic background and the body shape. SP-Colorectal and GTOS, on the other hand, have squared images of tissue scans and ground terrains. As a result, ViTs suffer more severely from low-frequency shortcuts for TextileNet than they do for SP-Colorectal and GTOS.

Therefore, low-frequency shortcuts are universal phenomena across different model sizes and architectures for texture-driven tasks. Pruning, however, is not a universally successful method to eliminate the low-frequency shortcut. In particular, when the model architecture has a strong bias towards low-frequency information, such as ViTs, and also the domain includes heavy low-frequency noise, e.g., TextileNet, pruning falls short on eliminating the shortcut. Nevertheless, in 10 of 12 cases (4 models & 3 datasets), pruning LFCs improves the ID performance.

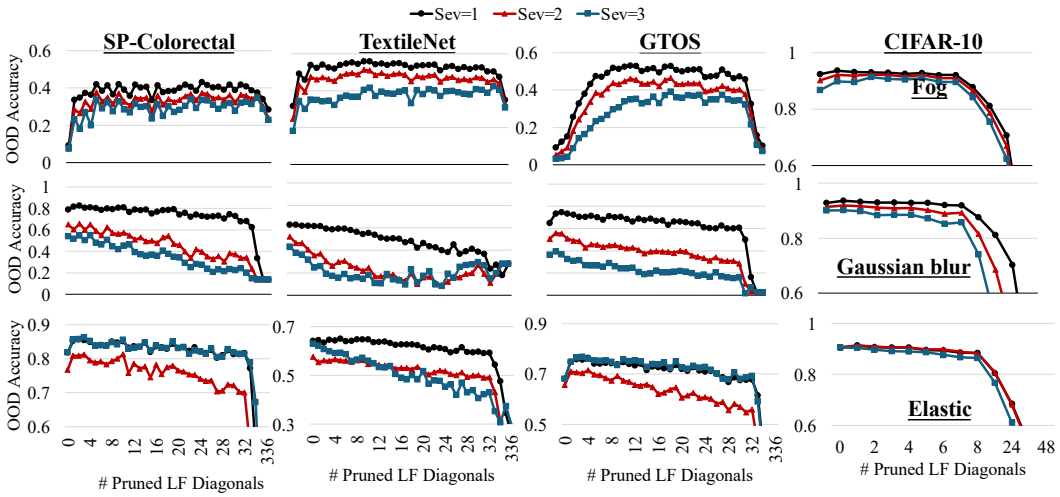


Figure 22: OOD performance under different severity levels. X-axis: number of pruned LFCs. Y-axis: OOD accuracy. Overall patterns remain similar across different severity levels. Fog's OOD accuracy improves as the low-frequency shortcut is eliminated. Gaussian blur and elastic face a trade-off between the improved representation and magnified corruptions at the HFCs.

H Results Across OOD Severity Levels

Figure 22 presents OOD results for the four tasks we study across three severity levels: 1, 2, and 3. The x-axis shows the number of pruned LF diagonals, whereas the y-axis shows the OOD accuracy. As can be seen, the main patterns remain the same across the severity levels. Texture-driven tasks have improved accuracy for fog corruption, as fog is low-frequency-heavy, and texture-driven tasks suffer from low-frequency shortcuts. Gaussian blur and elastic transform can increase, stabilize, or decrease accuracy, depending on the trade-off between improved representation and magnified high-frequency corruption. CIFAR-10 has a small accuracy drop at the severity of 1, whereas a more visible accuracy degradation at the severity of 2 and 3.

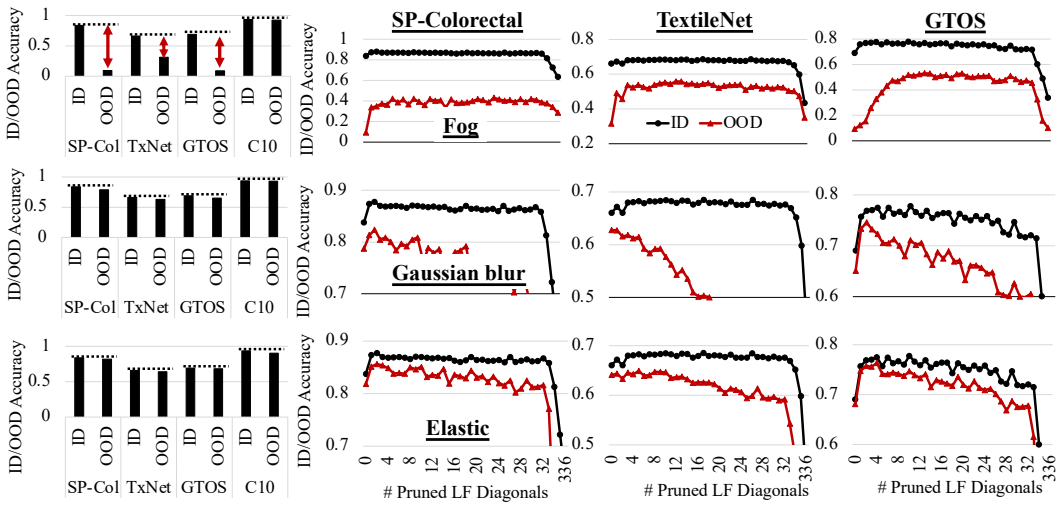


Figure 23: OOD Results for ResNet50.

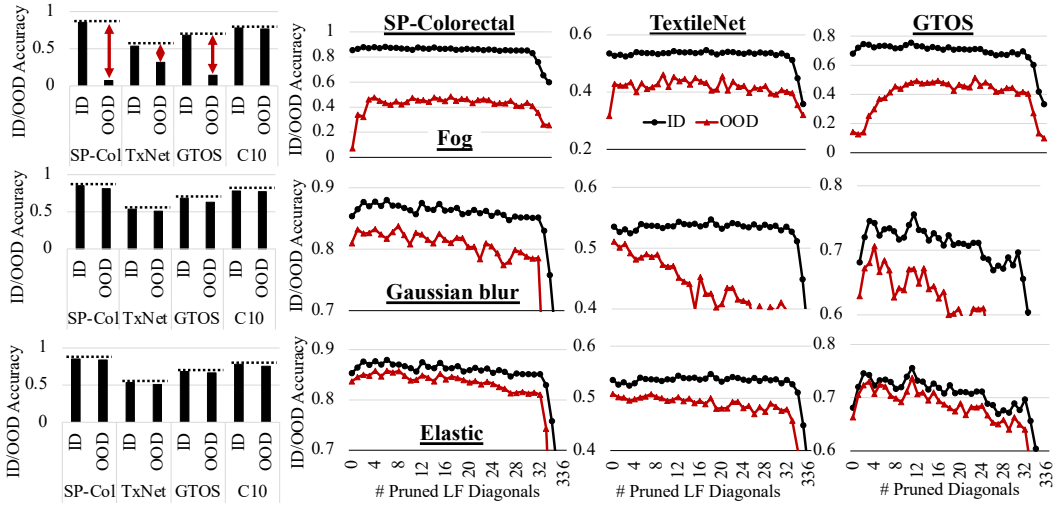


Figure 24: OOD Results for MobileNet-V3.

I Impact of Model Size and Architecture on OOD Results

Figure 23 to Figure 26 presents the OOD results for ResNet-50, MobileNet-V3, ViT-Small, and ViT-Tiny for the three corruption types we analyze, fog (low-frequency corruption), Gaussian blur (high-frequency corruption), and mixed low-and-high-frequeny corruption (elastic transform). As can be seen, trends are similar across different model sizes and architectures. For each model, we report the results for a single training with seed 42. Earlier in the appendix, we have shown that different seeds produce similar OOD results (see Appendix C).

J OOD Corruption Pipelines

Figure 27 presents the OOD results for the two corruption pipelines we tested: (i) corrupt-prune (CP), and (ii) prune-corrupt-prune (PCP). In Section 6, we covered the first pipeline. In this section, we present the results for both pipelines. The x-axis presents the number of pruned LFCs, and the y-axis presents the ID/OOD accuracy. While the red line shows the CP pipeline that we also present in Section 6, the blue line shows the PCP pipeline. Dark line presents the ID values for reference. As can be seen, blue and red lines are often close to each other and follow similar patterns. For the

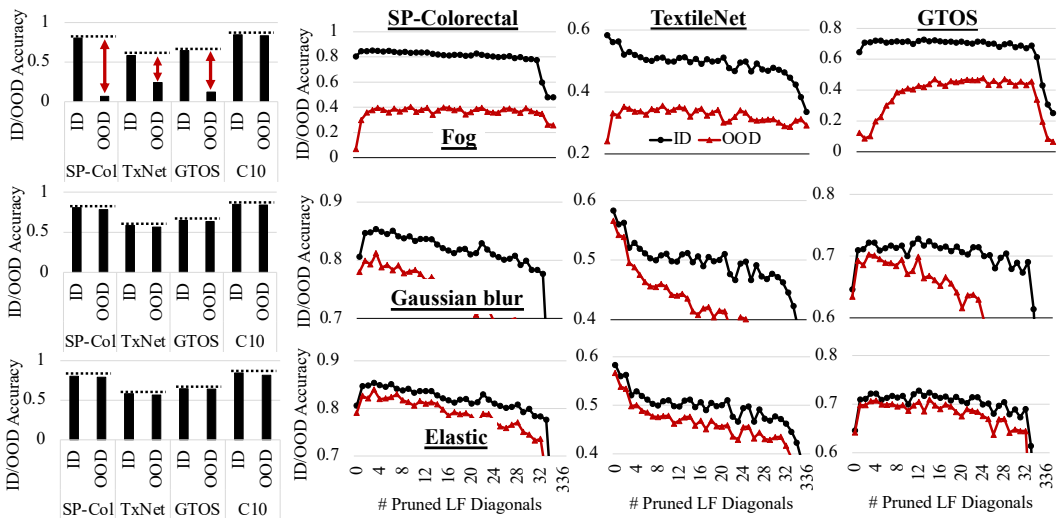


Figure 25: OOD Results for ViT-Small.

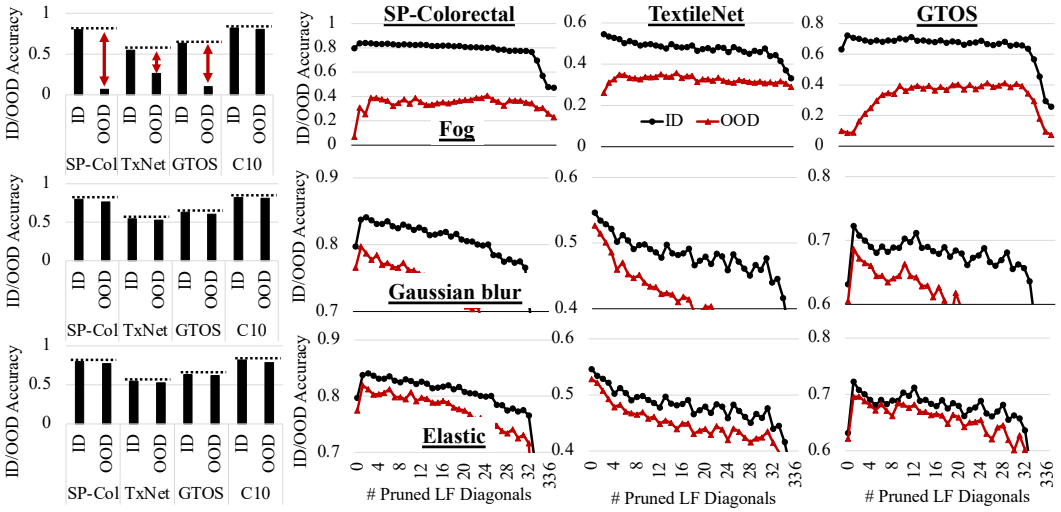


Figure 26: OOD Results for ViT-Tiny.

GTOS task and fog corruption, the PCP pipeline has a higher accuracy than the CP pipeline. In fact, the PCP pipeline largely recovers from the corruptions, closely approximating the ID accuracy. This is because applying corruption after pruning *reduces* the impact of corruption, as corruption uses some of the pruned components. As a result, the final OOD accuracy is higher for PCP than for CP.

For elastic corruption, however, CP achieves higher accuracy than PCP. This is because applying corruption after pruning *increases* its impact. Elastic transform spreads local pixels to their neighbors, which worsens its effect in the absence of LFCs. Which pipeline causes more harm depends on the exact formula the corruption uses and the dataset's characteristics.

Nevertheless, OOD accuracy follows similar trends compared to the ID accuracy for both pipelines. While LFC-pruning significantly helps recover from fog corruption, it introduces a trade-off for Gaussian blur and elastic transform corruptions. On the one hand, pruning LFCs improves representation, as HFCs better reflect the application semantics. On the other hand, the model's focus shifts to HFCs once LFCs are pruned. Hence, corruption at the HFCs is magnified. OOD accuracy increases/decreases/remains stable, depending on this trade-off.

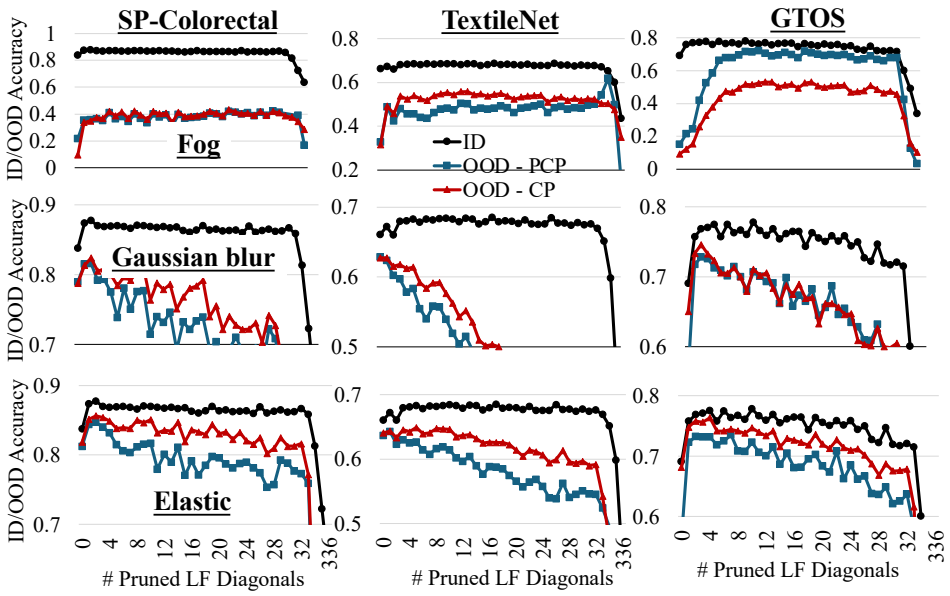


Figure 27: OOD corruption pipelines. PCP: prune-corrupt-prune. CP: corrupt-prune. **Top:** Low-frequency corruption (fog) results. **Middle:** High-frequency corruption (Gaussian blur) results. **Bottom:** Mixed-frequency corruption (elastic transform) results. PCP and CP follow similar trends across the tasks and corruption types.

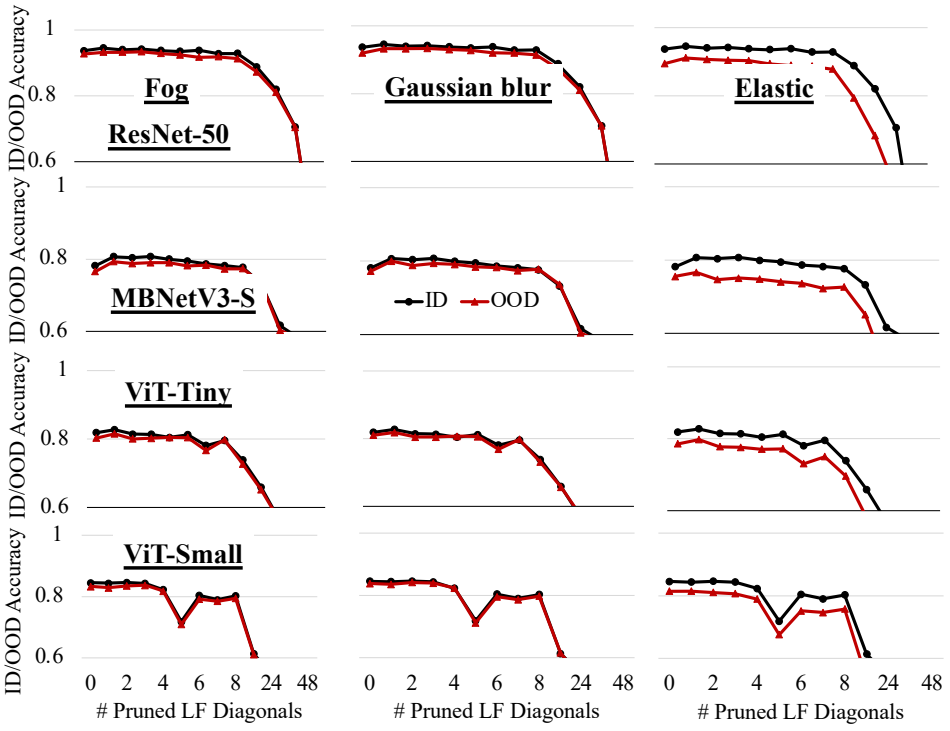


Figure 28: CIFAR-10's OOD performance closely approximates its ID performance. It suffers an observable drop in elastic corruption due to the elastic transform's broad coverage of corruption across low and high frequencies.

K OOD RESULTS ON CIFAR-10

CIFAR-10's OOD values closely approximate its ID values, thanks to its balanced spectral behavior shown in Figure 5. Figure 28 presents ID and OOD values for CIFAR-10 for the three corruptions we analyze. As shown, OOD values (red line) closely approximate the ID values (dark line). For elastic corruption, OOD values are observed to be lower. This is because elastic modifies a wide range of frequency components, including both low and high frequencies, which reduces the models' tolerance capacity.

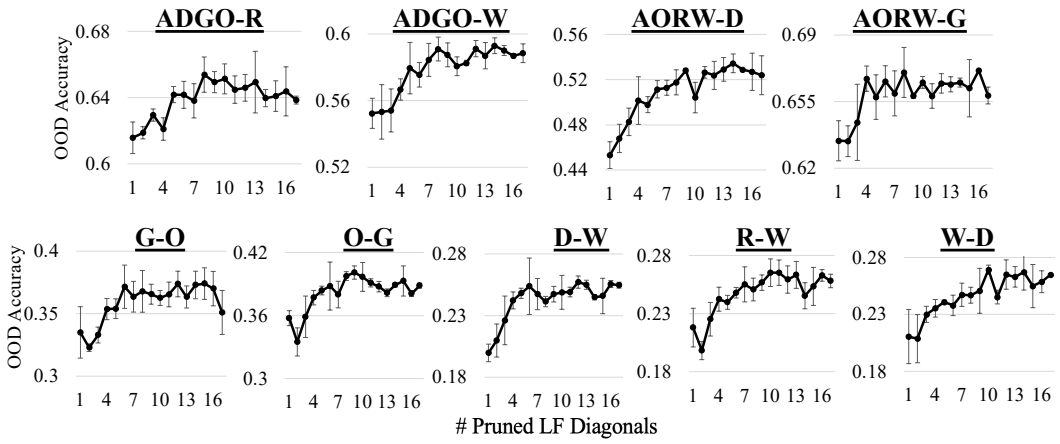


Figure 29: OOD results for real-world domain-generalization. Each graph represents results for a different subset of NICO++.

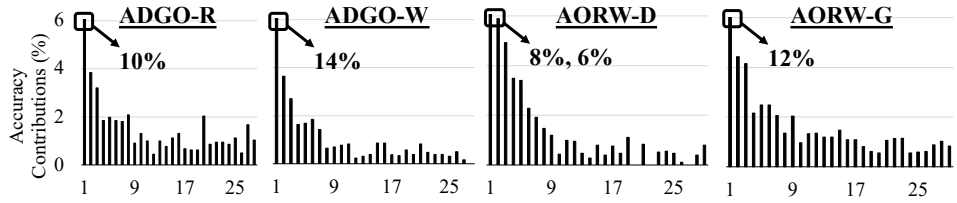


Figure 30: Spectral behavior for real-world domain-generalization. Each graph represents results for a different subset of NICO++.

L RESULTS ON REAL-WORLD DOMAIN-GENERALIZATION

Figure 29 presents OOD results for NICO++ for nine different combinations of training and validation sets using ResNet-50. As can be seen, pruning LFCs consistently improves accuracy. Figure 30 presents the spectral behavior for the four subsets, ADGO-R, ADGO-W, AORW-D, and AORW-D. As can be seen, spectral behavior is always skewed towards LFCs, even though being at different levels depending on the specific training/validation set combination. These results show that low-frequency shortcuts play an important role for domain-generalization problems. Neural nets are highly biased towards learning from background cues. Pruning LFCs forces the network to learn from more realistic higher-frequency features and hence achieve a better OOD generalization performance.

M TUNING HYPERPARAMETERS FOR LOW-FREQUENCY SHORTCUTS

We test GTOS and SP-Colorectal across different optimizers, SGD and Adam, using six different hyperparameters for ResNet-50 and ViT-Small. We list the hyperparameter search space for ResNet-50 in Table 2 and for ViT-Small in 3. Table 4 presents the ID results for ResNet-50 hyperparameter search space. As can be seen, pruning LFCs increases the ID accuracy for all the hyperparameters

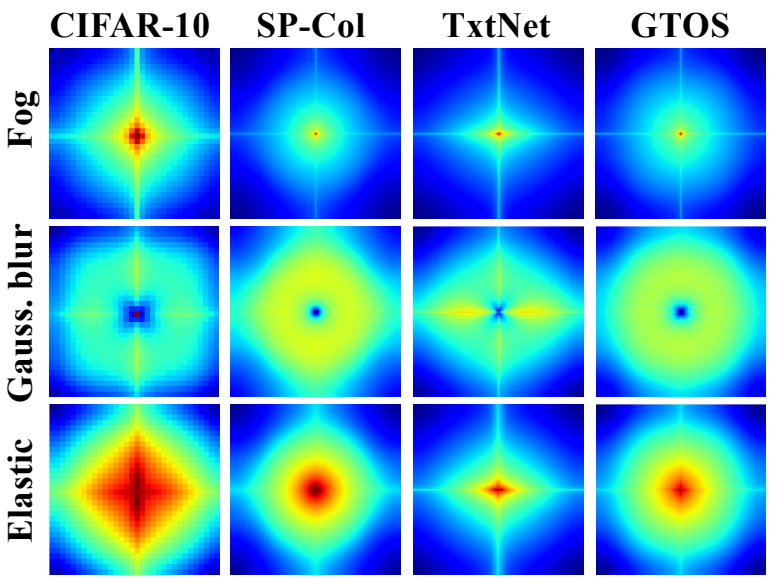


Figure 31: Frequency characteristics of the corruptions we use, across the four tasks we analyze. We use a log10-scale, unlike [106] in their Figure 2. This way, we can observe fog's high-frequency corruptions, which would otherwise go unnoticed. Each image represents the Fourier characteristics of the average difference between the original and corrupted image. We shift the lowest frequency coefficient to the center for compatibility with [106].

Table 2: Hyperparameter search space for ResNet-50.

HID	Optimizer	Learning rate	Momentum	Weight decay	Scheduler	Epochs
0	SGD	3×10^{-2}	0.9	5×10^{-4}	Cosine	200
1	SGD	1×10^{-1}	0.9	5×10^{-4}	Cosine	200
2	SGD	2×10^{-1}	0.9	5×10^{-4}	Cosine	200
3	SGD	1×10^{-1}	0.9	1×10^{-4}	Cosine	200
4	SGD	1×10^{-1}	0.9	1×10^{-3}	Cosine	200
5	SGD	1×10^{-1}	0.9	5×10^{-4}	Multistep	200
0	Adam	1×10^{-4}	–	1×10^{-4}	Cosine	200
1	Adam	3×10^{-4}	–	1×10^{-4}	Cosine	200
2	Adam	1×10^{-3}	–	1×10^{-4}	Cosine	200
3	Adam	3×10^{-4}	–	0	Cosine	200
4	Adam	3×10^{-4}	–	5×10^{-4}	Cosine	200
5	Adam	3×10^{-4}	–	1×10^{-4}	Multistep	200

for both optimizers. Table 5 presents the ID results for ViT-Small hyperparameters. Once again, we observe that pruning LFCs improves the ID accuracy for all the hyperparameters for both SP-Colorectal and GTOS datasets. This shows that low-frequency shortcuts are persistent across different hyperparameters.

N Additional Comparison to Related Work

Existing CIFAR-10 Studies. There are three main frequency characterization studies on CIFAR-10. [47] performs test set characterization on unpruned training images and concludes that high frequencies are important for CIFAR-10 classification. They prune all HFCs with an above radial frequency of 4.25 from test set images and observe up to 24% accuracy drop in test set accuracy[9]. Our results corroborate this, as the radial frequency of 4.25 corresponds to the top-left 8 LFCs in

[9] See Figure 8 of [47], a & b. First row with the first and second columns.

Table 3: Hyperparameter search space for ViT-Small. No scheduler or momentum used.

HID	Optimizer	Learning rate	Weight decay	Warmup epochs	Epochs
0	AdamW	1×10^{-3}	0.05	3.4	300
1	AdamW	3×10^{-3}	0.05	3.4	300
2	AdamW	5×10^{-3}	0.05	3.4	300
3	AdamW	3×10^{-3}	0.10	3.4	300
4	AdamW	3×10^{-3}	0.30	3.4	300
5	AdamW	3×10^{-3}	2.50	3.4	300

Table 4: GTOS ID performance with ResNet-50 using SGD and Adam across six hyperparameters each, as we prune increasing number of LFCs. As can be seen, pruning LFCs increases ID accuracy for both optimizers and all hyperparameters.

	GTOS w/ ResNet-50 & SGD						GTOS w/ ResNet-50 & Adam							
	hid=0	hid=1	hid=2	hid=3	hid=4	hid=5	hid=6	hid=7	hid=8	hid=9	hid=10	hid=11	avg. SGD	avg. Adam
0-LFC	0.706557	0.693443	0.698033	0.665574	0.699016	0.680328	0.674426	0.712787	0.680984	0.698689	0.705574	0.710820	0.690492	0.697213
1-LFC	0.775410	0.765574	0.760656	0.738033	0.759016	0.763934	0.746557	0.780000	0.762295	0.747213	0.755738	0.771475	0.760437	0.760546
2-LFC	0.761311	0.755738	0.746230	0.735410	0.761311	0.764262	0.745574	0.763934	0.758689	0.752787	0.767213	0.767213	0.754044	0.759235
3-LFC	0.775410	0.770164	0.765902	0.729180	0.778033	0.748852	0.752459	0.777377	0.760328	0.766885	0.761967	0.771475	0.761257	0.765082
4-LFC	0.789180	0.773115	0.765246	0.742295	0.767541	0.769836	0.748852	0.767869	0.761311	0.757705	0.777049	0.770492	0.767869	0.763880
5-LFC	0.780656	0.769508	0.768197	0.726557	0.760000	0.757049	0.744262	0.771475	0.748525	0.754426	0.760656	0.769508	0.760328	0.758142
6-LFC	0.766230	0.769180	0.774754	0.724590	0.767213	0.765574	0.743607	0.763934	0.744918	0.745902	0.773115	0.776393	0.761257	0.757978
7-LFC	0.761967	0.760984	0.760328	0.729836	0.783279	0.760000	0.738689	0.779016	0.749836	0.768197	0.770164	0.758033	0.759399	0.760656
8-LFC	0.774754	0.768525	0.762951	0.720328	0.768525	0.757705	0.740984	0.757377	0.751148	0.747213	0.769836	0.753115	0.758798	0.753279

our setup. Pruning everything beyond 8 LFCs can indeed cause a significant accuracy drop, as our accuracy contributions graph also shows in 5. [13] also performs test set characterization with a fixed unpruned training set of images, and show that pruning the highest end of the frequency spectrum causes the least accuracy loss[10], as we also have shown in our experiments. They further show that pruning higher frequencies in the lower half of the spectrum can cause a larger drop in accuracy than pruning lower frequencies. Our results corroborate this, as we have shown that the lowest 22 components in the lower half of the spectrum contribute roughly equally to test set accuracy. [89] performs training and test set characterization with a small number of configurations and shows that keeping LFCs contributes significantly more than HFCs to the test accuracy[11], which corroborates our results.

Existing ImageNet Studies. Our CIFAR-10 results on LFC and HFC pruning from training and test sets corroborate with numerous existing ImageNet [25] results. Task-specific image compression studies by [76] and [97] show that pruning HFCs is much more advantageous than LFCs for a subset and full ImageNet[12]. Similarly, [106] shows that accuracy drops much faster when training and test set images are high-pass filtered, as opposed to low-pass filtered[13]. Lastly, [13] shows a similar trend to ours for a subset of ImageNet, as they prune high frequencies[14]. These studies, however, did not conduct as fine-grained an analysis as we have. Their unit of analysis is more than a single diagonal. Hence, their results might be incomplete. Further analysis of ImageNet and similar datasets is in our future work.

Existing studies either group frequency coefficients in broader units than diagonals [13; 47], or perform an analysis one by one for each frequency coefficient separately [91]. Using broad units limits the findings of the existing studies. Performing an analysis for each coefficient is feasible for a test set characterization, but it is too costly for a training set characterization, as it requires too many training sessions. Diagonal-wise grouping of coefficients provides a middle ground between the two: it captures a fine-grained behavior along increasing frequencies, and is also feasible in terms of experimental time. Recent studies on task-specific image compression have shown that diagonal-wise pruning provides successful results [76; 77; 97; 96].

[10] See Figure 5 of [13], top row, (15-32) data point on the x-axis.

[11] See Table 1 of [89].

[12] See Figure 5 of [76], and Figure 5 of [97].

[13] See Figure 1 of [106].

[14] See Figure 5 of [13], third figure from the top.

Table 5: SP-Colorectal and GTOS ID performance with ViT-Small using Adam across six hyperparameters each, as we prune increasing number of LFCs. As can be seen, pruning LFCs increases ID accuracy for both datasets and all hyperparameters.

	SP-Colorectal w/ ViT-Small & Adam							GTOS w/ ViT-Small & Adam						
	hid=0	hid=1	hid=2	hid=3	hid=4	hid=5	avg.	hid=0	hid=1	hid=2	hid=3	hid=4	hid=5	avg.
0-LFC	0.791008	0.783392	0.786028	0.802871	0.803017	0.743117	0.784906	0.624918	0.345574	0.596721	0.602623	0.643607	0.634426	0.574645
1-LFC	0.826889	0.830258	0.819127	0.838459	0.842267	0.769479	0.821080	0.705574	0.367541	0.537705	0.651475	0.718033	0.659344	0.606612
2-LFC	0.825425	0.829379	0.827914	0.846514	0.842414	0.755419	0.821178	0.699344	0.423934	0.455410	0.661639	0.700984	0.670492	0.601967
3-LFC	0.830844	0.833333	0.835970	0.847832	0.839192	0.742824	0.821666	0.698361	0.474426	0.347213	0.681967	0.718361	0.635738	0.592678
4-LFC	0.828500	0.827182	0.829233	0.837141	0.845489	0.758055	0.820933	0.693115	0.400656	0.530820	0.675738	0.733115	0.631475	0.610820
5-LFC	0.836262	0.832162	0.839484	0.849151	0.842999	0.753661	0.825620	0.709180	0.363934	0.583934	0.685574	0.718033	0.617377	0.613005
6-LFC	0.832601	0.825864	0.833919	0.839777	0.848858	0.749268	0.821715	0.695410	0.413443	0.517049	0.682951	0.722951	0.623607	0.609235
7-LFC	0.839484	0.832748	0.842560	0.841681	0.846661	0.756444	0.826596	0.700000	0.393443	0.624918	0.684918	0.708525	0.614754	0.621093
8-LFC	0.832601	0.832748	0.838899	0.838606	0.846514	0.740920	0.821715	0.693443	0.368197	0.613443	0.689836	0.719672	0.616066	0.616776

Task-specific Image Compression. Furthermore, there has been a growing interest in task-specific image compression, where images are compressed for a specific computer vision task [76; 77; 30; 33; 52; 71; 97; 96]. This requires learning in the compressed domain. Studies have shown that training and test images can be significantly compressed without significant loss of accuracy. This allows reducing training storage, training time, and inference time, providing a resource-efficient computer vision system. Existing task-specific image compression studies are all based on a few standard benchmarks, similar to existing frequency characterization studies. Our characterization of the training and test sets sheds light on task-specific image compression across numerous application domains.

Simplicity Bias. Simplicity bias might occur due to visible [19; 20] or invisible superficial cues [91; 93]. While visible features are easy to detect via visual inspection, invisible features are deeply embedded in the data and require formal tools to analyze [61; 68; 18; 23; 108; 91; 93]. One such useful tool is Fourier analysis [91; 93]. Fourier analysis transforms data from its original domain, e.g., the spatial domain for images, into the frequency domain, enabling a principled analysis of shortcut behavior by leveraging the frequency structure embedded in the data. Analysis using other tools, such as geometric transformations or information-theoretic methods, is an interesting avenue for future work.

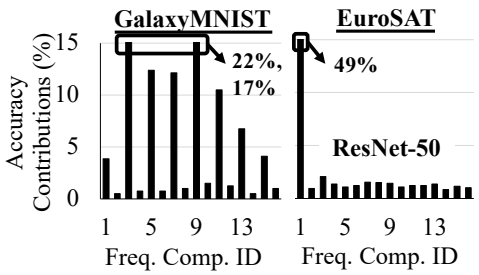


Figure 32: Skewed spectral behavior makes mixed-semantics tasks vulnerable to OOD corruptions.

O Spectral Behavior of Mixed Texture-Shape Domains

Figure 32 presents spectral behavior for GalaxyMNIST and EuroSAT datasets using ResNet-50. As can be seen, they are both heavily skewed towards LFCs. GalaxyMNIST has a high contribution of every other LFCs with decreasing levels of contributions. EuroSAT has a dramatically high reliance to the lowest-frequency component. These skewed behaviors make GalaxyMNIST and EuroSAT vulnerable to OOD corruptions. Pruning LFCs allow network to learn from a wider-range of higher-frequency features and be more robust to OOD corruptions.

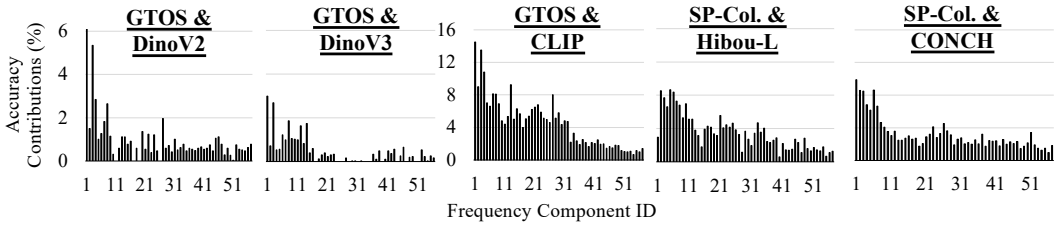


Figure 33: Spectral behavior of general-purpose (DinoV2, DinoV3, and CLIP) and domain-specific (Hibou-L and CONCH) pretrained foundation models.

P SPECTRAL BEHAVIOR OF PRETRAINED FOUNDATION MODELS

Figure 33 presents spectral behavior of the five VLM/VFMs we analyze. As can be seen, all models have a skewed behavior towards LFCs, except DinoV3. Figure 10 has shown that ID and OOD accuracies of DinoV2 increases as we prune LFCs. Hence, DinoV2 (86M) suffers from low-frequency shortcuts, despite being trained with large number of images and complex data augmentations. DinoV3 (840M) mitigates the shortcuts thanks to its 10x larger model size compared to DinoV2. This, however, comes at the cost of a high computational cost. CLIP (151M) has a heavy reliance on the LFCs. As a result, pruning LFCs cause a significant accuracy losses, which it cannot recover using higher-frequency features. Hibou-L (300M) also has a high reliance on LFCs. It can, however, recover from the lost information when LFCs are pruned and maintain similar level of accuracy (Figure 10). CONCH (307M), being the other VLM we analyze, is similar to CLIP. It heavily relies on LFCs, and it cannot recover from the lost information due to LFC-pruning.

CONCH's unpruned accuracy (0.82) is lower than that of the from-scratch trained ResNet-50 (0.84). Once LFCs are pruned, ResNet-50 achieves a significantly higher accuracy than CONCH (0.88), despite its smaller size (25M) and smaller training set. This shows that, even though foundation models can be helpful for shortcuts, they might also introduce confounding features, resulting in suboptimal accuracy.

Hibou-L's unpruned accuracy is 0.9. Once we prune LFCs, its accuracy remains stable, then slowly drops. This shows that domain-specific foundation models can indeed mitigate the shortcuts. Nevertheless, (i) they may introduce confounding features in their pre-training set, as in CONCH, and (ii) they are very expensive to pre-train, fine-tune, and use for inference, with a few percent advantage compared to our from-scratch trained models. Our LFC-pruned ResNet-50 (25M params) achieves 0.88 accuracy, despite being >10x smaller and trained on a much smaller set.

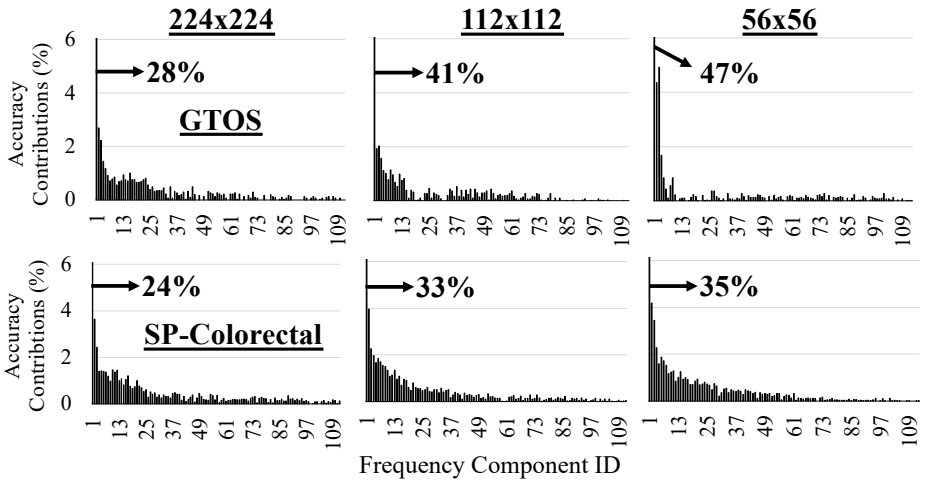


Figure 35: Lower-resolution images suffer from an increased the degree of low-frequency shortcuts, as they concentrate information on lower-frequency bands.

Q RESULTS ON IMAGE RESOLUTION

Figure 34 presents ID accuracy results as we prune LFCs, for GTOS and SP-Colorectal using ResNet-50 and decreasing image resolutions from 224x224, to 112x112 and 56x56. As can be seen, pruning LFCs improves ID accuracy for both cases for all the image resolutions. Figure 35 presents the spec-

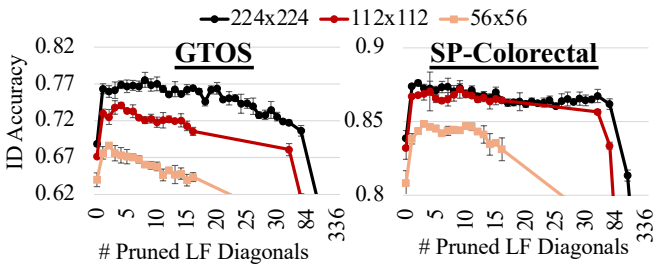


Figure 34: Low-frequency shortcuts persist across different image resolutions.

tral behavior for unpruned trained for decreasing resolutions for GTOS (top row) and SP-Colorectal (bottom row). As can be seen, spectral behaviors get more and more skewed towards LFCs, as image resolutions decreases. This shows that decreased resolution increases the degree of low-frequency shortcuts.

Chapter 5.1.3.1 of [54] studies image halving in the frequency domain. Reducing the resolution of an image is equal to (i) transforming the image into the frequency domain, (ii) taking its top-left NxM square of coefficients, where N and M are reduced spatial dimensions, and (iii) applying inverse frequency transformation using the reduced NxM basis functions (see Figure 5.5 of [54]). Therefore, reducing image resolution is equivalent to concentrating information in lower-frequency bands, which results in an increased degree of low-frequency shortcuts.

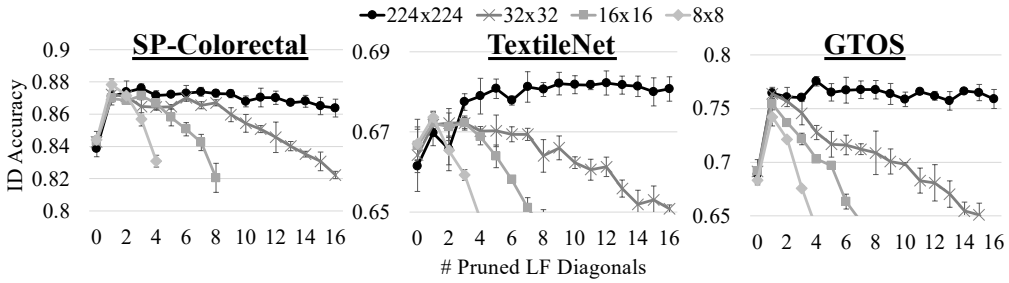


Figure 36: Large block sizes capture low-frequency shortcuts better than small block sizes.

R Results on Frequency-Transformation Block Size

Figure 36 presents ID accuracy results as we decrease block size from 224x224, to 32x32, 16x16, and 8x8. As can be seen, using larger block sizes generally produces higher accuracies than that of smaller block sizes. This is because low-frequency shortcuts are usually wide background cues that smaller block sizes only partially captures.

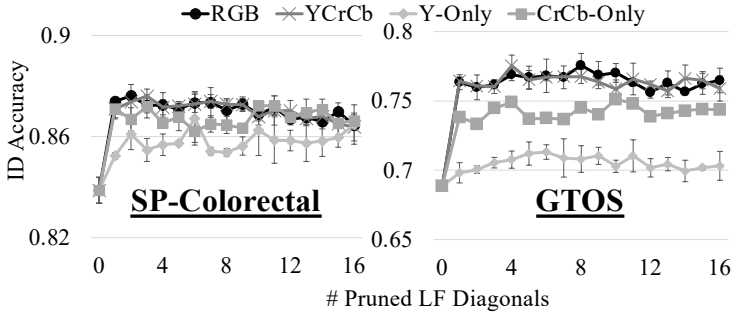


Figure 37: Low-frequency shortcuts persist across different color spaces. Pruning only grayscale or only color channels is not enough to completely eliminate the shortcuts.

S Results on Color Spaces

Figure 37 presents ID accuracy results when pruning LFCs over different color spaces and channels. RGB presents results for the color space that we use throughout the paper. YCrCb is a popular color space used by JPEG, which allows separating color information (Cr and Cr channels) from the grayscale information (Y channel). This is useful for many image compression, as human eye is more sensitive to the grayscale information that it is to the color information. Image compression algorithms more aggressively compress color channels than grayscale channel, providing a more efficient image storage [86].

As the figure shows, LFC-pruning over RGB or YCrCb color spaces have similar ID accuracy. On the other hand, pruning only the Y channel, or only the color channels have a lower accuracy than pruning all the YCrCb channels. This shows that low-frequency shortcuts is not a channel-specific phenomenon, but exits across all the channels. Hence, it needs to be mitigated across spectrum of all information channels.

T Frequency Characteristics of OOD Corruptions

Figure 31 presents frequency characteristics of the used corruptions: fog, Gaussian blur, and elastic transform. We follow the same methodology as [106] when obtaining their Figure 2, with one exception. We use a log10 scale instead of using raw numbers. We use `torch.log10()`. This way, we can see fog's corruption at higher frequencies more easily, which would otherwise be invisible in Figure 2 of [106]. Other than this difference, we use the exact same methodology. Each image in Figure 31 shows the Fourier spectrum of the average difference between the original and corrupted images in the validation sets, for each dataset. The lowest-frequency coefficient is shifted to the center of each graph for compatibility with [106]. We use `torch.fft.fft2()`, and `torch.fft.fftshift()`.

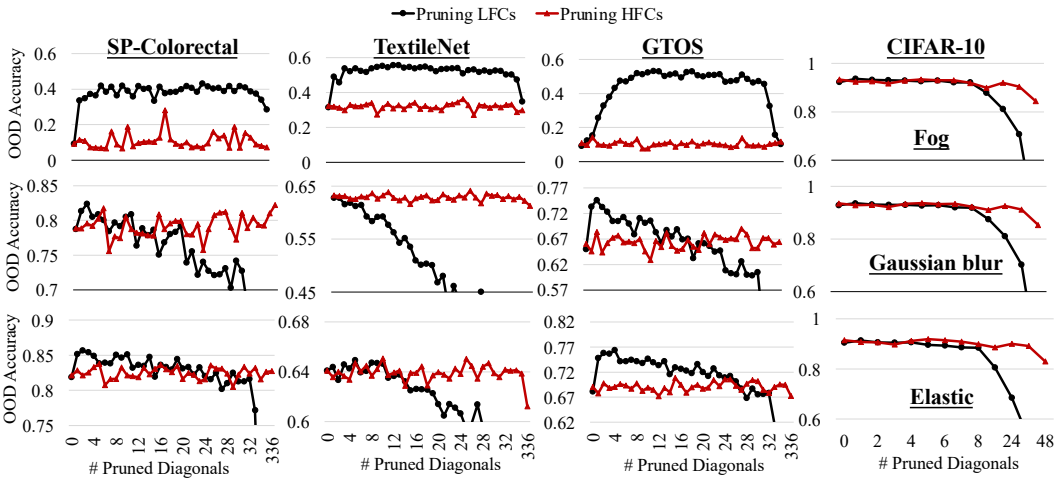


Figure 38: Impact of pruning HFCs on OOD performance (red line). We present pruning HFCs, along with the results for pruning LFCs, to provide a more complete picture. Similar to the ID performance, pruning HFCs has a minimal impact on OOD performance. For TextileNet and CIFAR-10, the impact is higher, as these two tasks depend more on higher frequencies at the lower end of the spectrum.

U Impact of HFCs on OOD Generalization

Figure 38 presents the impact of pruning LFCs (dark line) and HFCs (red line) on OOD generalization for the four tasks we analyze. As shown, pruning HFCs minimally impacts accuracy across all tasks and corruptions. This, once again, shows the efficiency of high-frequency compression. For CIFAR-10 and TextileNet, there is a slight drop at the end, similar to their ID accuracies. CIFAR-10 and TextileNet depend on higher frequencies, as their accuracy contribution graphs show in Figure 5 (top row). Hence, pruning more than 224/32 diagonals from the bottom-right corner to the top-left corner reduces their accuracy.