How Do Training Methods Influence the Utilization of Vision Models?

Paul Gavrikov[1,2] **Shashank Agnihotri**[2] **Margret Keuper**[2,3] **Janis Keuper**[1,2]

[1] IMLA, Offenburg University, Germany
[2] University of Mannheim, Germany
[3] Max-Planck-Institute for Informatics, Saarland Informatics Campus, Germany

Abstract

Not all learnable parameters (*e.g.*, weights) contribute equally to a neural network's decision function. In fact, entire layers' parameters can sometimes be reset to random values with little to no impact on the model's decisions. We revisit earlier studies that examined how architecture and task complexity influence this phenomenon and ask: *is this phenomenon also affected by how we train the model?*

We conducted experimental evaluations on a diverse set of ImageNet-1k classification models to explore this, keeping the architecture and training data constant but varying the training pipeline. Our findings reveal that the training method strongly influences which layers become critical to the decision function for a given task. For example, improved training regimes and self-supervised training increase the importance of early layers while significantly under-utilizing deeper layers. In contrast, methods such as adversarial training display an opposite trend. Our preliminary results extend previous findings, offering a more nuanced understanding of the inner mechanics of neural networks.

Code: `https://github.com/paulgavrikov/layer_criticality`

1 Introduction

A famous neurology myth often misattributed to Albert Einstein states that humans only use 10% of the neural connections in their brains (Radford, 1999). While modern research assumes that humans use all neural connections (Boyd, 2008; Herculano-Houzel, 2009) – the same cannot be said about artificial neural networks. Quite the contrary, it is well known that trained neural networks do not utilize their entire capacity. This becomes evident through the lens of *parameter pruning* (LeCun et al., 1989b; Hassibi et al., 1993), where (large numbers) of neurons can be removed after training without affecting performance, or, alternatively, the *distillation* of large into equivalent smaller networks (Hinton et al., 2015; Hoffmann et al., 2021). Alas, the learned decision function only occupies a fraction of the neural network and the remaining neurons seem to be wasted.

Zhang et al. (2022) showed that this seems to affect layers disproportionally. The learnable parameters[1] of some layers are *critical* to the decision function and replacing them with any other values than the learned ones (significantly) affects accuracy. In contrast, the performance is barely affected when the parameters in *auxiliary* layers are randomized. For instance, entire residual blocks of *ResNets* (He et al., 2015) trained on ImageNet (Russakovsky et al., 2015) can be randomized without hurting accuracy. Affected layers seem to be dictated by training data size or more generally the complexity of the training function in addition to the architecture. Chatterji et al. (2020) have even extended these

[1]Throughout the rest of the article, the term *parameters* specifically refers to learnable parameters.

Interpretable AI: Past, Present and Future Workshop at NeurIPS 2024

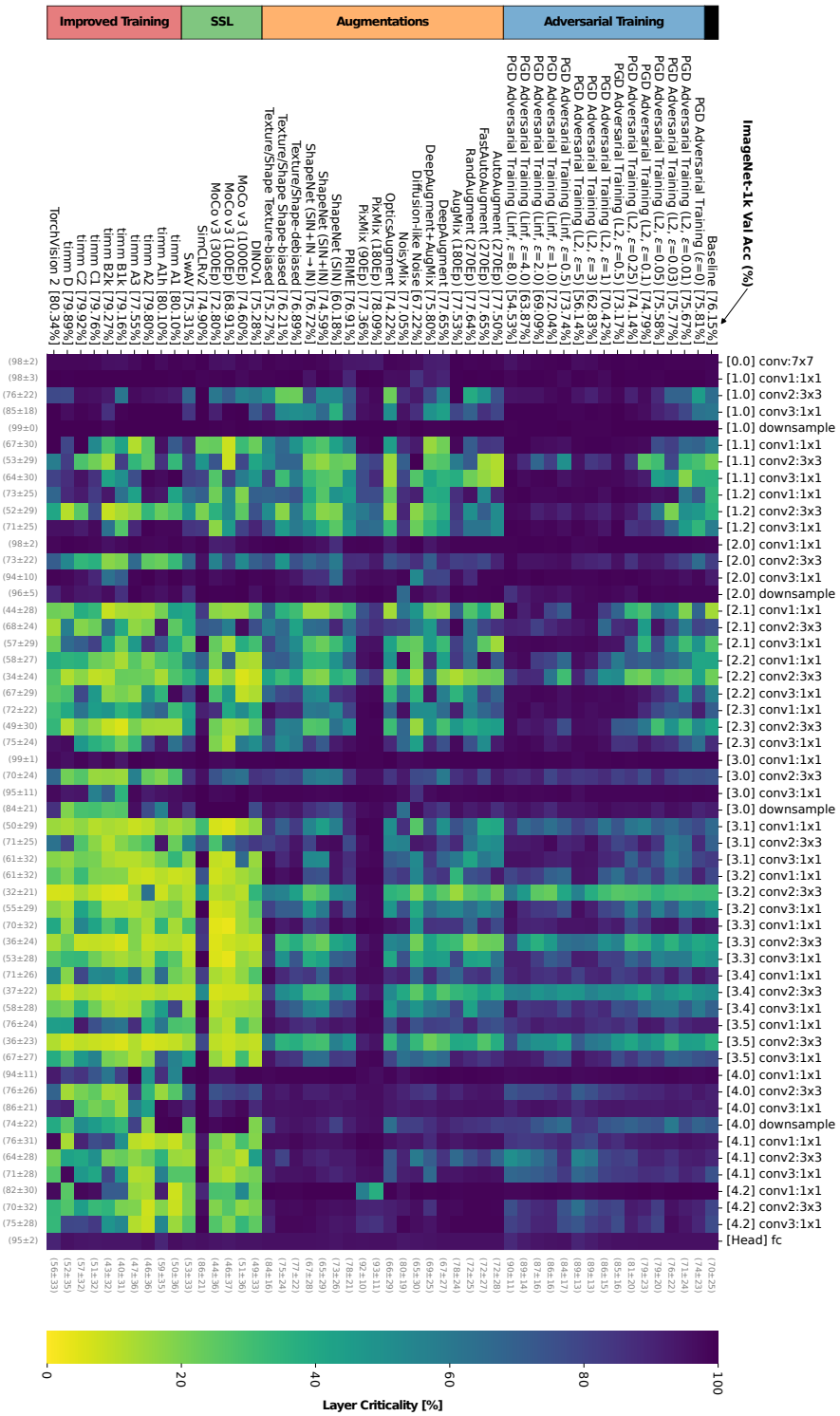

Figure 1: **Training methods determine what layers become critical.** We measure the criticality of fifty different ResNet-50-based models that all utilize the same exact network architecture and training data (ImageNet-1k) but differ in their training methods. Darker spots denote layers that are *critical*, *i.e.*, in significantly different predictions and decreased performance after reset. Brighter spots are *auxiliary*, *i.e.*, resetting these layers does not significantly affect the model. We denote the average (mean±std) layer criticality for both, a model across layers on the right, for a layer across model on the bottom.

findings to generalization, by showing that the average module criticality correlates with performance - *i.e.*, more complex networks seem to generalize better.

We revisit the prior findings of (Zhang et al., 2022; Chatterji et al., 2020) on image classification models, which were obtained under rather clean conditions, such as the absence of weight decay (Krogh & Hertz, 1991) or batch normalization layers (Ioffe & Szegedy, 2015) during training, and an overall simple training pipeline. These conditions do not reflect practical training pipelines well. Thus, we raise the question: *how does the training method affect layer criticality?* To this end, we study criticality on a large model zoo of image classification models where the training data and architecture are fixed but the training pipeline is modified.

Specifically, we use the model zoo of (Gavrikov & Keuper, 2024) with a few replacements and select *fifty* different ResNet-50 image classification models (He et al., 2015) that all utilize the exact same network architecture but were trained in different manners. Our changes to training belong to the following categories: a *baseline* model following the original training (He et al., 2015); various *augmentation* techniques (Hendrycks et al., 2020, 2022, 2021; Jaini et al., 2024; Müller et al., 2023; Modas et al., 2022; Li et al., 2021; Erichson et al., 2022; Cubuk et al., 2019, 2020; Lim et al., 2019; Geirhos et al., 2019); *adversarial training* against a projected gradient descent (PGD) adversary (Madry et al., 2018; Salman et al., 2020) – which technically, is also a form of augmentation, but we observed significantly different behavior of these models; various *self-supervised learning (SSL)* approaches (Chen et al., 2020, 2021; Caron et al., 2021, 2020) with supervised finetuning of the classification-head; and improved *training recipes* in *timm* (Wightman, 2019; Wightman et al., 2021) and *PyTorch* (Paszke et al., 2019; Vryniotis, 2023) combining multiple hyper-parameter optimizations into supervised training.

2 Methodology

Our study aims to assess the contribution of individual layers to a neural network's decision function. To gauge a layer's importance, we replace its parameters with random values (*randomization*). If the network's decisions remain largely unchanged after randomization, it suggests that the learned parameters contribute little beyond noise.

This methodology largely follows Zhang et al. (2022), who reset the parameters of individual layers to values drawn from the original initialization[2]. However, while Zhang et al. (2022) measured *criticality* of a layer by the change in accuracy due to the randomization, we measure the angle between the probability vectors resulting from the randomization. Specifically, we apply a Softmax function to the network logits to obtain (pseudo-)probabilities, measure the cosine distance between those before and after randomization, and aggregate the measurements into a single scalar by averaging over all samples. The effect of each layer randomization on this measured distance is what we define as the *criticality* of a layer.

This methodological change can be evaluated in an unsupervised manner and more importantly, is also sensitive to changes in the probability distribution including variations in errors (we refer the reader to (Geirhos et al., 2020) for a discussion on why this is important). As such, it provides a more holistic measurement of consistency in the decision before and after randomization well beyond correct predictions.

We call a layer *auxiliary* if the decision is insignificantly affected by the reset ($\approx 0\%$ criticality) and *critical* ($\approx 100\%$ criticality) if the distance between decisions changes significantly. Realistically, the criticality for most layers does not lie on the extremes of this spectrum, but anywhere in between. Due to a significant variance (standard error of up to 45% on a few layers in specific models; see Figure 5) in criticality on some layers, we repeat experiments with different random seeds and report the mean over three trials. For computational reasons, we evaluate layers on a subset of 10,000 random images from the ImageNet ILSVRC-2012 challenge validation set (Russakovsky et al., 2015).

Zhang et al. (2022) analyzed residual blocks as a whole – in contrast, we more meticulously randomize individual layers, which include different convolution and fully-connected layers. However, we do not re-initialize batch normalization layers (Ioffe & Szegedy, 2015) to avoid signal propagation issues.

[2]We match the initialization used by each training method.

3 Results

Due to the wide use and availability of pre-trained models, currently, all our results are obtained on ResNet-50 (He et al., 2015). Recall that this architecture consists of a stem (denoted by *[0.*]*), 4 stages (denoted by *[1-4.*]*), a pooling layer, and a fully-connected classification head (denoted by [Head]). Each stage consists of several *residual* bottle-neck blocks which include learnable 1×1 convolutions (*conv1*, *conv3*), 3×3 convolutions (*conv2*), as well as batch-normalization layers (Ioffe & Szegedy, 2015). The first residual block in each stage is special, as it downsamples by a strided convolution, thus, adding a learnable layer on the skip connection (*downsample*).

General Observations. The results in Figure 1 (analogously see Appendix A for more views) clearly show that the training method influences what layers become critical – despite that all models were trained on the same training set (some with more extreme forms of data augmentation utilizing a negligible amount of extra data).

In contrast to previous findings (Zhang et al., 2022; Chatterji et al., 2020), we observe that **no layer is always auxiliary** across training methods. For instance, we observe an average criticality of just 36% for a spatial convolution layer (*[3.5] conv2*). Yet, if we randomize the same layer in a *PixMix* (Hendrycks et al., 2022) model, we observe a strong criticality of 95%. On the opposite, we do find layers that are always critical. As expected, these include the initial stem convolution (*[0.0] conv*) and the classification head (*fc*). Beyond, we find that most first convolution layers in each stage (*[*.0] conv1*) are critical – yet the number of outliers increases with depth. Similarly, we find that the downsampling convolution (*[*.0] downsample*) in each stage is often critical. In stage 1 this layer is critical for all models but again the criticality of deeper downsampling convolutions depends on the training strategy used for the model. Lastly, akin to Gavrikov & Keuper (2023), we find that pointwise convolution layers tend to be more critical than spatial convolution layers (except for the stem). For all other layers, criticality depends on the training method. In the following paragraphs, we analyze specific categories of training methods.

Adversarial Training (AT). This training technique intends to increase the robustness of neural networks by training on adversarially perturbed training samples (Madry et al., 2018). To avoid perturbations that cause a shift in semantic meaning, perturbations are often constrained by an attack perturbation budget ϵ for some ℓ_p norm. We consider AT using a PGD attack (Madry et al., 2018) which optimizes the perturbations over several iterations (here: three). Please note, that reported ϵ values for ℓ_∞ norms are short for $\epsilon/255$ (but not for the ℓ_2 norm).

We find that AT increases the criticality proportional to the attack budget ϵ during training results. To make this more tangible, we average the criticality over all layers and show the results in Figure 2. We do not observe differences between training that utilizes ℓ_2 or ℓ_∞ norms for attacks. Our findings in Figure 1 suggest that neural networks utilize more of their capacity under increasing training attack strength. This augments previous findings that showed similar insights through accuracy improvements with larger networks (Madry et al., 2018) or richer representations in convolutions filters (Gavrikov & Keuper, 2022a,b). AT primarily increases the criticality in the layers of the first and second stages and slightly in the third stage. The criticality of layers in the fourth stage is barely affected but rather decreases, compared to the baseline (please refer to Figure 4).

Augmentations. Compared to AT, the influence of different augmentation strategies seems weaker. We do find that augmentations tend to increase the average criticality, *i.e.*, they do occupy more of the network capacity – but most changes are rather small. Affected layers seem to fluctuate by method, but we find that all augmentation methods consistently increase the criticality of some of the deepest layers (*[4.0] downsample, [4.1] conv2, [4.2] conv2/3*). Similar to the organization of the human brain (Hubel & Wiesel, 1979), deeper layers in neural networks are associated with activations of more complex features. For images, these tend to correspond to shapes as opposed to texture information that is captured by early layers (LeCun et al., 1989a; Yosinski et al., 2014). Indeed, prior work has observed that our tested augmentations increase in their shape responses (Gavrikov & Keuper, 2024). Thus, a reasonable hypothesis is that the increased criticality in deeper layers correlates with stronger shape representations.

Strong outliers to our observations are the *PixMix* models (Hendrycks et al., 2022). These models have the highest average criticality in our model zoo without a single auxiliary layer. The augmentation technique has been shown to improve multiple safety dimensions beyond test accuracy and combined

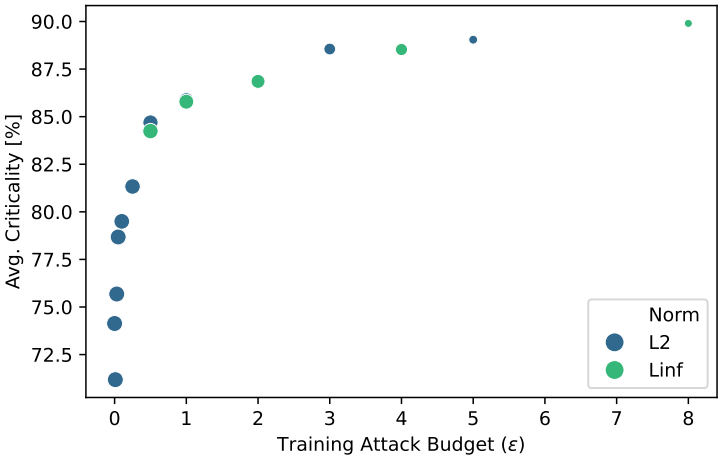

Figure 2: **Adversarial training increases the average criticality proportional to the training attack budget ϵ.** We ablate ℓ_∞ from ℓ_2-norm training but do not observe any significant differences in their trends. The marker size in the plot indicates the validation accuracy on ImageNet-1k (larger is better).

with the findings of Chatterji et al. (2020) it may indeed suggest that a higher degree of criticality correlates with "better" neural networks.

Self-Supervised Learning (SSL). SSL has been shown to produce rich representations for many downstream tasks (see Oquab et al. (2024) for a recent example) as the granularity and implicit biases of annotations do not confine it. Interestingly, we find that *DINO* (Caron et al., 2021), *MoCo v3* (Chen et al., 2021), and *SwAV* (Caron et al., 2020) severely differ in their criticality measurements from the supervised models we discussed before. These SSL models have a large presence of auxiliary layers in the last three stages but (slightly) increase in criticality of early layers. This suggests that SSL learns shorter decision functions and, thus, seeks to strengthen early operations. However, we again find an outlier: *SimCLRv2* (Chen et al., 2020) has no auxiliary layers and shows a somewhat similar distribution of criticality to *PixMix*.

Improved Training Recipes. The standard 90 epoch ImageNet training with simple augmentations was shown to be suboptimal for many models including ResNets (Wightman et al., 2021). Modern training recipes utilize a significantly more complex set of training tweaks – often in combination with longer training schedules (Wightman et al., 2021; Vryniotis, 2023). Similar to our findings on self-supervised learners, these improved training methods appear to shift model decisions to early operations while relaxing deeper operations. Most notably, this increases the criticality of the first and second residual blocks. These models show great improvements in generalization on many datasets but at the same time, they even further prioritize texture information (Gavrikov & Keuper, 2024). Ultimately, this may suggest that these techniques better fit ImageNet problems but may not provide improvements for representations beyond.

Lastly, we attempt to correlate average layer criticality with accuracy on ImageNet-1k, across different training strategies in Figure 3. Ignoring the category labels we observe a moderate Spearman's $R = -0.46$, but when we remove the adversarially trained models the correlation is faint (Spearman's $R = -0.17$). Thus, there is a low likelihood of a causal connection between ImageNet accuracy and layer criticality.

4 Conclusion, Limitations, and Future Work

Our ongoing study extends previous findings about the complexity of learned decision functions of image classification models. Instead of analyzing individual models as often done in mechanistic interpretability works (*e.g.*, (Olah et al., 2020; Goh et al., 2021)) we focus on the common impact of training methods on layer criticality. We have shown that some forms of training leave distinct patterns in the decision function.

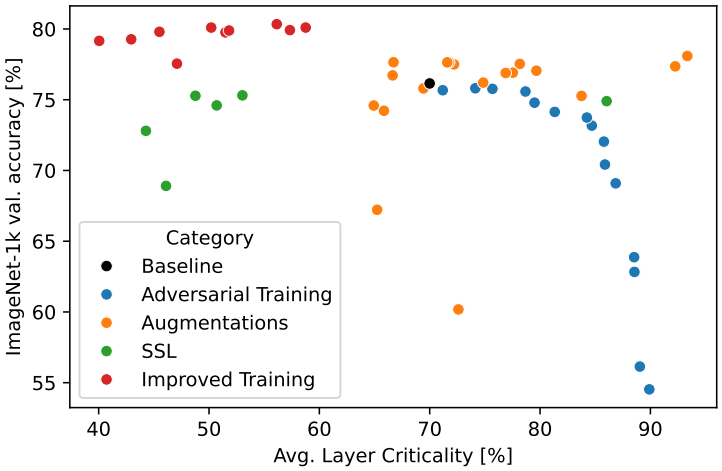

Figure 3: **Correlation between average network criticality and performance on ImageNet-1k.**

Discussion. Assessing the generalization of neural networks through benchmarks can be tricky, as models may specialize in specific settings at a cost to others. Capturing all of these nuances in static test datasets is often an unrealistic journey. For example, models excel in accuracy on the "clean" ImageNet data but drop in performance if the same samples are corrupted (Hendrycks & Dietterich, 2019). In that sense, the decision function complexity may offer a better (relative) assessment of generalization by focusing on the inner mechanics of models as opposed to benchmark results. Yet, it remains to be seen if the observed correlations of Chatterji et al. (2020) hold on larger model zoos and wider notions of generalization beyond clean accuracy on the ImageNet validation set.

However, even if criticality and generalization are orthogonal to each other, the complexity of the decision function may still be relevant and offer a better explanation of phenomena that were linked to (adversarial) robustness before. For instance, these include more human-likeness (Geirhos et al., 2021; Gavrikov et al., 2023), better calibration (Grabinski et al., 2022a), and transferability (Salman et al., 2020) under adversarial training. (Gavrikov & Keuper, 2022a,b) have shown that robust models contain more diverse feature representations and suggested that this may be linked to transferability.

Future Work. We aim to extend our study to understand if the distinct patterns in the decision function patterns may be an artifact of the scenario we studied. For instance, we wonder if layer criticality depends on the test data - *i.e.*, what happens if we replace ImageNet test data with corrupted images (Hendrycks & Dietterich, 2019), different renditions (Hendrycks et al., 2021), or adversarial perturbations (Goodfellow et al., 2015). Another dimension to explore is the architecture – we are curious if our findings scale to modern network classification architectures such as *ConvNeXt* (Liu et al., 2022), *vision transformers (ViTs)* (Dosovitskiy et al., 2021), zero-shot classification with join-embedding models like *CLIP* (Radford et al., 2021), (classification) prompting on large language models with vision capabilities such as *BLIP* (Li et al., 2022). How does criticality change under architectural interventions aimed at robustness improvements, *e.g.*, (Grabinski et al., 2022b; Lukasik et al., 2023; Agnihotri et al., 2023, 2024a,b) and their task-specific robustness tests (Agnihotri et al., 2024c)? Can we maybe even use this method to better understand the symbiotic connections of individual modalities in multi-modal models parallel to the approaches in (Goh et al., 2021; Gavrikov et al., 2024)?

Beyond mechanistic interpretability, we also hope that our results could guide practical applications such as model compression and transfer learning. We would expect, that auxiliary layers do not need to be distilled, or alternatively could be pruned without significantly affecting performance. As such, compression techniques may be affected by the training method. Additionally, it was shown that robust models perform better in transfer learning (Salman et al., 2020) but is this indeed due to the increased robustness? We have observed that robust models also have a high ratio of critical layers, perhaps this might be a better explanation.

Acknowledgements. S.A. and M.K. acknowledge support by the DFG Research Unit 5336 - Learning to Sense (L2S).

References

Agnihotri, S., Gandikota, K. V., Grabinski, J., Chandramouli, P., and Keuper, M. On the unreasonable vulnerability of transformers for image restoration-and an easy fix. In *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, 2023.

Agnihotri, S., Grabinski, J., Keuper, J., and Keuper, M. Beware of Aliases–Signal Preservation is Crucial for Robust Image Restoration. *arXiv preprint arXiv:2304.14736*, 2024a.

Agnihotri, S., Grabinski, J., and Keuper, M. Improving Feature Stability during Upsampling – Spectral Artifacts and the Importance of Spatial Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024b.

Agnihotri, S., Jung, S., and Keuper, M. CosPGD: an efficient white-box adversarial attack for pixel-wise prediction tasks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024c.

Boyd, R. Do People Only Use 10 Percent of Their Brains? *Scientific American*, 2008.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Chatterji, N., Neyshabur, B., and Sedghi, H. The intriguing role of module criticality in the generalization of deep networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big Self-Supervised Models are Strong Semi-Supervised Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Chen, X., Xie, S., and He, K. An Empirical Study of Training Self-Supervised Vision Transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. AutoAugment: Learning Augmentation Strategies From Data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Erichson, N. B., Lim, S. H., Xu, W., Utrera, F., Cao, Z., and Mahoney, M. W. NoisyMix: Boosting Model Robustness to Common Corruptions. *arXiv preprint arXiv:2202.01263*, 2022.

Gavrikov, P. and Keuper, J. CNN Filter DB: An Empirical Investigation of Trained Convolutional Filters. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

Gavrikov, P. and Keuper, J. Adversarial Robustness Through the Lens of Convolutional Filters. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022b.

Gavrikov, P. and Keuper, J. The Power of Linear Combinations: Learning with Random Convolutions. *arXiv preprint arXiv:2301.11360*, 2023.

Gavrikov, P. and Keuper, J. Can Biases in ImageNet Models Explain Generalization? In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Gavrikov, P., Keuper, J., and Keuper, M. An Extended Study of Human-Like Behavior Under Adversarial Training. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023.

Gavrikov, P., Lukasik, J., Jung, S., Geirhos, R., Lamm, B., Mirza, M. J., Keuper, M., and Keuper, J. Are Vision Language Models Texture or Shape Biased and Can We Steer Them? *arXiv preprint arXiv:2403.09193*, 2024.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, 2019.

Geirhos, R., Meding, K., and Wichmann, F. A. Beyond accuracy: quantifying trial-by-trial behaviour of CNNs and humans by measuring error consistency. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Geirhos, R., Narayanappa, K., Mitzkus, B., Thieringer, T., Bethge, M., Wichmann, F. A., and Brendel, W. Partial success in closing the gap between human and machine vision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A., and Olah, C. Multimodal Neurons in Artificial Neural Networks. *Distill*, 2021.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Grabinski, J., Gavrikov, P., Keuper, J., and Keuper, M. Robust Models are less Over-Confident. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.

Grabinski, J., Jung, S., Keuper, J., and Keuper, M. FrequencyLowCut Pooling–Plug & Play against Catastrophic Overfitting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022b.

Hassibi, B., Stork, D., and Wolff, G. Optimal Brain Surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, 1993.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Hendrycks, D. and Dietterich, T. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *International Conference on Learning Representations (ICLR)*, 2019.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. *International Conference on Learning Representations (ICLR)*, 2020.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., and Gilmer, J. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Hendrycks, D., Zou, A., Mazeika, M., Tang, L., Li, B., Song, D., and Steinhardt, J. PixMix: Dreamlike Pictures Comprehensively Improve Safety Measures. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Herculano-Houzel, S. The Human Brain in Numbers: A Linearly Scaled-up Primate Brain. *Frontiers in Human Neuroscience*, 2009.

Hinton, G., Vinyals, O., and Dean, J. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Hoffmann, J., Agnihotri, S., Saikia, T., and Brox, T. Towards improving robustness of compressed CNNs. In *ICML Workshop on Uncertainty and Robustness in Deep Learning (UDL)*, 2021.

Hubel, D. H. and Wiesel, T. N. Brain Mechanisms of Vision. *Scientific American*, 1979.

Ioffe, S. and Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

Jaini, P., Clark, K., and Geirhos, R. Intriguing Properties of Generative Classifiers. In *International Conference on Learning Representations (ICLR)*, 2024.

Krogh, A. and Hertz, J. A. A Simple Weight Decay Can Improve Generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1991.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1989a.

LeCun, Y., Denker, J., and Solla, S. Optimal Brain Damage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1989b.

Li, J., Li, D., Xiong, C., and Hoi, S. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

Li, Y., Yu, Q., Tan, M., Mei, J., Tang, P., Shen, W., Yuille, A., and Xie, C. Shape-Texture Debiased Neural Network Training. In *International Conference on Learning Representations (ICLR)*, 2021.

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. Fast AutoAugment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A ConvNet for the 2020s. *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Lukasik, J., Gavrikov, P., Keuper, J., and Keuper, M. Improving Native CNN Robustness with Filter Frequency Regularization. *Transactions on Machine Learning Research (TMLR)*, 2023.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Modas, A., Rade, R., Ortiz-Jiménez, G., Moosavi-Dezfooli, S.-M., and Frossard, P. PRIME: A Few Primitives Can Boost Robustness to Common Corruptions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

Müller, P., Braun, A., and Keuper, M. Classification Robustness to Common Optical Aberrations. In *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, 2023.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom In: An Introduction to Circuits. *Distill*, 2020.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Radford, B. The Ten-Percent Myth. *Skeptical Inquirer*, 1999.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., and Madry, A. Do Adversarially Robust ImageNet Models Transfer Better? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Vryniotis, V. How to Train State-Of-The-Art Models Using TorchVision's Latest Primitives, 2023. URL https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives. [Online; accessed 15. Nov. 2023].

Wightman, R. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models, 2019.

Wightman, R., Touvron, H., and Jegou, H. ResNet strikes back: An improved training procedure in timm. In *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*, 2021.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Zhang, C., Bengio, S., and Singer, Y. Are All Layers Created Equal? *Journal of Machine Learning Research (JMLR)*, 2022.

A Additional Analysis

We alternatively visualize the results from Figure 1 by plotting the difference in the criticality of each layer with respect to the baseline in Figure 4. Figure 5 displays the standard error of measurements over three independent runs with different seeds for the randomization.

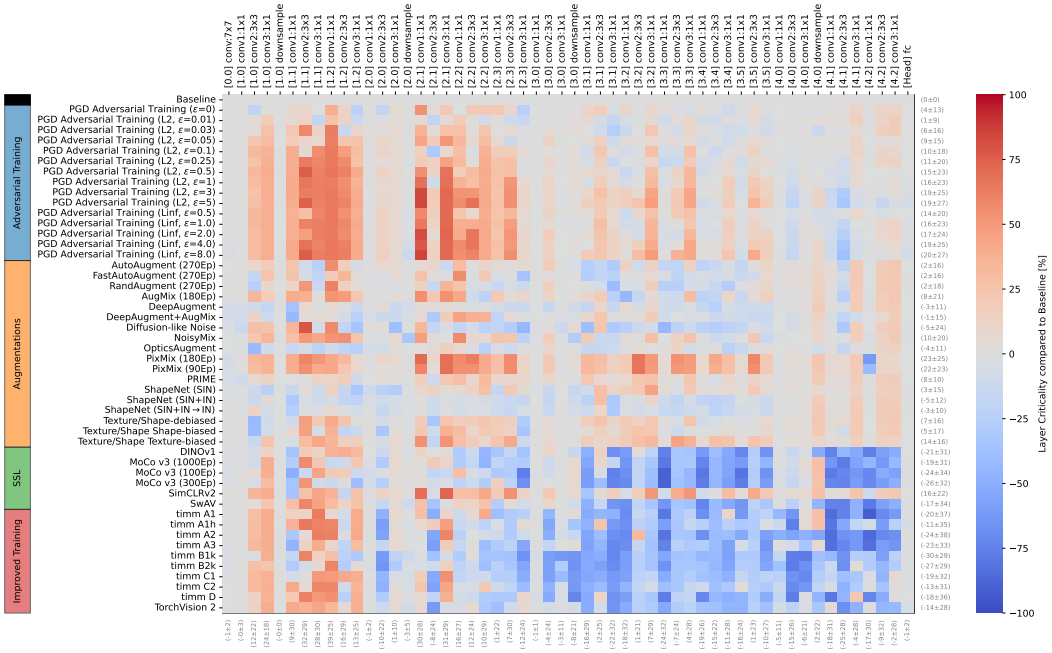

Figure 4: **Criticality difference to the baseline.** In addition to the plot in Figure 1, we here show the difference to the baseline model (He et al., 2015). Positive numbers indicate increases in criticality and negative numbers decrease.

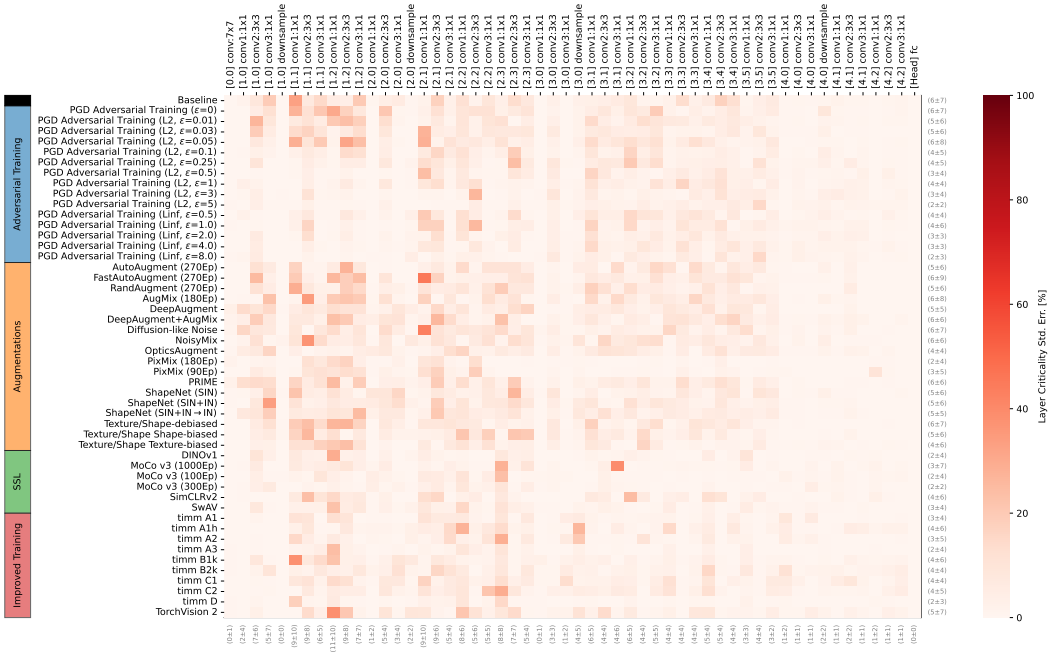

Figure 5: **Criticality standard error.** In addition to the plot in Figure 1, we here show the standard error in criticality measurements over 3 runs with different seeds for the randomization.

B Model Training Strategy Details

In Table 1, we provide a legend for the training strategies considered.

Table 1: An overview of the utilized models (training strategies) in our study.

	Model	**ImageNet accuracy** [%]
	Original Baseline (He et al., 2015)	76.15
Adversarial Training	PGD-AT (ϵ=0) (Salman et al., 2020; Madry et al., 2018)	75.81
	PGD-AT (ℓ_2, ϵ=0.01) (Salman et al., 2020; Madry et al., 2018)	75.67
	PGD-AT (ℓ_2, ϵ=0.03) (Salman et al., 2020; Madry et al., 2018)	75.77
	PGD-AT (ℓ_2, ϵ=0.05) (Salman et al., 2020; Madry et al., 2018)	75.58
	PGD-AT (ℓ_2, ϵ=0.1) (Salman et al., 2020; Madry et al., 2018)	74.79
	PGD-AT (ℓ_2, ϵ=0.25) (Salman et al., 2020; Madry et al., 2018)	74.14
	PGD-AT (ℓ_2, ϵ=0.5) (Salman et al., 2020; Madry et al., 2018)	73.17
	PGD-AT (ℓ_2, ϵ=1) (Salman et al., 2020; Madry et al., 2018)	70.42
	PGD-AT (ℓ_2, ϵ=3) (Salman et al., 2020; Madry et al., 2018)	62.83
	PGD-AT (ℓ_2, ϵ=5) (Salman et al., 2020; Madry et al., 2018)	56.14
	PGD-AT (ℓ_∞, ϵ=0.5) (Salman et al., 2020; Madry et al., 2018)	73.74
	PGD-AT (ℓ_∞, ϵ=1.0) (Salman et al., 2020; Madry et al., 2018)	72.04
	PGD-AT (ℓ_∞, ϵ=2.0) (Salman et al., 2020; Madry et al., 2018)	69.09
	PGD-AT (ℓ_∞, ϵ=4.0) (Salman et al., 2020; Madry et al., 2018)	63.87
	PGD-AT (ℓ_∞, ϵ=8.0) (Salman et al., 2020; Madry et al., 2018)	54.53
Augmentations	AutoAugment (270Ep) (Cubuk et al., 2019)	77.50
	FastAutoAugment (270Ep) (Lim et al., 2019)	77.65
	RandAugment (270Ep) (Cubuk et al., 2020)	77.64
	AugMix (180Ep) (Hendrycks et al., 2020)	77.53
	DeepAugment (Hendrycks et al., 2021)	76.65
	DeepAugment+AugMix (Hendrycks et al., 2021)	75.80
	Diffusion-like Noise (Jaini et al., 2024)	67.22
	NoisyMix (Erichson et al., 2022)	77.05
	OpticsAugment (Müller et al., 2023)	74.22
	PRIME (Modas et al., 2022)	76.91
	PixMix (180Ep) (Hendrycks et al., 2022)	78.09
	PixMix (90Ep) (Hendrycks et al., 2022)	77.36
	ShapeNet (SIN) (Geirhos et al., 2019)	60.18
	ShapeNet (SIN+IN) (Geirhos et al., 2019)	74.59
	ShapeNet (SIN+IN $\rightarrow$ IN) (Geirhos et al., 2019)	76.72
	Texture/Shape-debiased Augmentation (Li et al., 2021)	76.89
	Texture/Shape-Shape Bias Augmentation (Li et al., 2021)	76.21
	Texture/Shape-Texture Bias Augmentation (Li et al., 2021)	75.27
SSL	DINOv1 (Caron et al., 2021)	75.28
	MoCo v3 (1000Ep) (Chen et al., 2021)	74.60
	MoCo v3 (100Ep) (Chen et al., 2021)	68.91
	MoCo v3 (300Ep) (Chen et al., 2021)	72.80
	SimCLRv2 (Chen et al., 2020)	74.90
	SwAV (Caron et al., 2020)	75.31
Improved Training	timm A1 (Wightman, 2019; Wightman et al., 2021)	80.10
	timm A1h (Wightman, 2019; Wightman et al., 2021)	80.10
	timm A2 (Wightman, 2019; Wightman et al., 2021)	79.80
	timm A3 (Wightman, 2019; Wightman et al., 2021)	77.55
	timm B1k (Wightman, 2019; Wightman et al., 2021)	79.16
	timm B2k (Wightman, 2019; Wightman et al., 2021)	79.27
	timm C1 (Wightman, 2019; Wightman et al., 2021)	79.76
	timm C2 (Wightman, 2019; Wightman et al., 2021)	79.92
	timm D (Wightman, 2019; Wightman et al., 2021)	79.89
	TorchVision 2 (Vryniotis, 2023; Paszke et al., 2019)	80.34