Shortcut Learning Through the Lens of Training Dynamics

Nihal Murali [1] **Aahlad Manas Puli** [2] **Ke Yu** [1] **Rajesh Ranganath** [2] **Kayhan Batmanghelich** [3]

Abstract

This paper aims to better understand shortcut learning through the lens of the learning dynamics of the internal neurons during the training process. We make the following observations: (1) While previous works treat shortcuts as synonymous with spurious correlations, we emphasize that not all spurious correlations are shortcuts. We show that shortcuts are only those spurious features that are "easier" than the core features. (2) We build upon this premise and use *instance difficulty* methods (like Prediction Depth (Baldock et al., 2021)) to quantify "easy" and to identify this behavior during the training phase. (3) We empirically show that shortcut learning can be detected by observing the learning dynamics of the DNN's *early layers*. In other words, easy features learned by the initial layers of a DNN early during the training are potential shortcuts. We verify our claims on medical and vision datasets, both simulated and real, and justify the empirical success of our hypothesis by showing the theoretical connections between Prediction Depth and information-theoretic concepts like $\mathcal{V}$-usable information (Ethayarajh et al., 2021). Lastly, our experiments show the insufficiency of monitoring only accuracy plots during training (as is common in machine learning pipelines). We highlight the need for monitoring early training dynamics using example difficulty metrics.

1. Introduction

Geirhos et al. (2020) define shortcuts as spurious correlations that exist in standard benchmarks but fail to hold in real-world settings. The emphasis on shortcuts being synonymous with spurious correlations has led to the widespread adoption of viewing shortcut learning as a distribution shift problem (Bellamy et al., 2022; Wiles et al., 2021; Adnan et al., 2022; Kirichenko et al., 2022). While the distribution shift explains part of the story, we emphasize that what is equally important for shortcut learning is the difficulty of the spurious features themselves (see Fig-1).

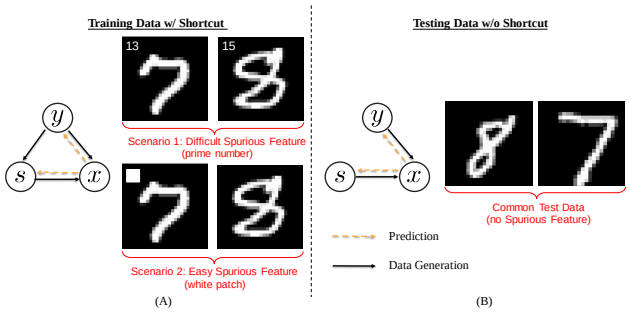

Figure 1. The causal view of shortcut learning is insufficient. If x can predict s, and y is not causally related to s on the test data, then s is viewed as a shortcut. (A) The figure shows two scenarios for even-odd classification. In scenario 1, all even numbers have a spurious composite number (located at the top-left), and odd numbers have a prime number. In scenario 2, all odd numbers have a spurious white patch. The spurious white patch (prime number) is an easy (hard) feature, so the model uses (ignores) it. This shows not all spurious correlations are shortcuts.

The premises that support our hypothesis are as follows: ***(P1)*** Shortcuts are only those spurious features that are *"easier"* to learn than the core features (see Fig-1). ***(P2)*** Initial layers of a DNN tend to learn easy features, whereas the later layers tend to learn the harder ones (Zeiler & Fergus, 2014; Baldock et al., 2021). ***(P3)*** Easy features are learned much earlier than the harder ones during training (Mangalam & Prabhu, 2019; Rahaman et al., 2019). Premises ***(P1-3)*** lead us to conjecture that: *"Easy features learned by the initial layers of a DNN early during the training are potential shortcuts."*

We make the following observations. First, when the spurious features are *known*, the paper sheds light on when and if we should be concerned about learning shortcuts. Second, when the spurious feature is *unknown* a priori, but a human user has an intuition about the difficulty of the task

[1]Intelligent Systems Program, School of Computing and Information, University of Pittsburgh [2]Department of Computer Science, New York University [3]Department of Electrical and Computer Engineering, Boston University. Correspondence to: Nihal Murali <nihal.murali@pitt.edu>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA.

in comparison to some reference tasks, the proposed metric can be used to detect shortcut learning patterns during training (figure-4). Third, we empirically show that our hypothesis works well on medical and vision datasets (sections-3.2,A.6), both simulated and real, regardless of the DNN architecture used. We justify this empirical success by theoretically connecting prediction depth with information-theoretic concepts like $\mathcal{V}$-usable information (Ethayarajh et al., 2021) (sections-2,A.1). Lastly, our experiments highlight that monitoring only accuracy during training is insufficient, and we need to monitor the learning dynamics of the model using instance difficulty metrics to detect shortcut learning (section-A.4). This could save time and computational costs and help develop reliable models that do not rely on spurious features.

2. Background and Methodology

Let P_{tr} and P_{te} be the training and test distributions defined over the random variables $\mathbf{X}$ (input), $\mathbf{y}$ (label), and $\mathbf{s}$ (*latent* spurious feature).

Definition-1 (Spurious Feature $\mathbf{s}$): A latent feature $\mathbf{s}$ is called spurious if it is correlated with label $\mathbf{y}$ in the training data but not in the test data. Specifically, the joint probability distributions P_{tr} and P_{te} can be factorized as follows.

$$P_{tr}(\mathbf{X},\mathbf{y},\mathbf{s}) = P_{tr}(\mathbf{X}|\mathbf{s},\mathbf{y})P_{tr}(\mathbf{s}|\mathbf{y})P_{tr}(\mathbf{y})$$

$$P_{te}(\mathbf{X},\mathbf{y},\mathbf{s}) = P_{tr}(\mathbf{X}|\mathbf{s},\mathbf{y})P_{te}(\mathbf{s})P_{tr}(\mathbf{y}).$$

The variable $\mathbf{s}$ appears to be causally related to $\mathbf{y}$ but is not. This is shown in Fig-1.

Definition-2 (Task Difficulty Ψ): Let $\Psi^{P}_{\mathcal{M}}(\mathbf{X}\rightarrow\mathbf{y})$ indicates the difficulty of predicting $\mathbf{X}\rightarrow\mathbf{y}$ for a model $\mathcal{M}$, where $\mathbf{X},\mathbf{y}\sim P$. Consider a joint distribution $(\mathbf{X},\mathbf{y},\mathbf{t})\sim P$ for two tasks, $\mathbf{t}$, and $\mathbf{y}$. Then, $\Psi^{P}_{\mathcal{M}}(\mathbf{X}\rightarrow\mathbf{y}) > \Psi^{P}_{\mathcal{M}}(\mathbf{X}\rightarrow\mathbf{t})$ indicates that the task $\mathbf{X}\rightarrow\mathbf{y}$ is harder than $\mathbf{X}\rightarrow\mathbf{t}$ for a given model $\mathcal{M}$.

Definition-3 (Shortcut): The spurious feature $\mathbf{s}$ is a potential shortcut for model $\mathcal{M}$ iff $\Psi^{P_{tr}}_{\mathcal{M}}(\mathbf{X}\rightarrow\mathbf{y}) > \Psi^{P_{tr}}_{\mathcal{M}}(\mathbf{X}\rightarrow\mathbf{s})$. In other words, given the input $\mathbf{X}$, predicting spurious feature $\mathbf{s}$ is easier for $\mathcal{M}$ than predicting the true label $\mathbf{y}$.

The definitions make it clear that "shortcut" is not just related to the dataset alone but is also closely tied to the model and the task. What is a shortcut for one model may not be so for another. We now explain two metrics (Prediction Depth and $\mathcal{V}$-Usable Information) to measure $\Psi^{P}_{\mathcal{M}}$. We use a binary classification setting to explain the concepts used in this section.

Notion of Prediction Depth: The PD is defined by building k-NN classifiers on the embedding layers of the model. The PD is simply the earliest layer after which all subsequent k-NN predictions remain the same (0 or 1) (Baldock et al., 2021). See Appendix-A.8 for more details. Figure-2 illustrates how to read the PD plots used in our experiment.

Notion of $\mathcal{V}$-Usable Information: The Mutual Information between input and output, $I(X;Y)$, is invariant with respect to lossless encryption of the input, i.e., $I(\tau(X);Y) = I(X;Y)$. Such a definition assumes unbounded computation and is counter-intuitive to define task difficulty as heavy encryption of X does not change the task difficulty. The notion of *"Usable Information"* introduced by Xu et al. (2020) assumes bounded computation based on the model family $\mathcal{V}$ under consideration. Usable information is measured under a framework called *predictive $\mathcal{V}$-information* (Xu et al., 2020). Ethayarajh et al. (2021) introduce *pointwise $\mathcal{V}$-information* (PVI) for measuring example difficulty.

$$\text{PVI}(x\rightarrow y) = -\log_2 g[\phi](y) + \log_2 g'[x](y), \quad \text{s.t.} \quad g, g' \in \mathcal{V} \tag{1}$$

The function g is trained on (ϕ, y) input-label pairs, where ϕ is a null input that provides no information about the label y. g' is trained on (x, y) pairs from the training data. Lower PVI instances are harder for $\mathcal{V}$ and vice-versa.

Proposition 1: (Informal) Consider two datasets: $D_s \sim P_{tr}(\mathbf{X},\mathbf{y})$ with spurious features and $D_i \sim P_{te}(\mathbf{X},\mathbf{y})$ without them. For some mild assumptions on PD (see Appendix-A.1), if the mean PD of D_s is less than the mean PD of D_i, then the $\mathcal{V}_{cnn}$-usable-information for D_s is larger than the $\mathcal{V}_{cnn}$-usable-information for D_i: $\mathcal{I}^{D_s}_{\mathcal{V}_{cnn}}(X\rightarrow Y) > \mathcal{I}^{D_i}_{\mathcal{V}_{cnn}}(X\rightarrow Y)$.

See proof in Appendix-A.1. The proposition intuitively implies that a sufficient gap between the mean PD of spurious and core features can cause the model to learn spurious features instead of core ones. This proposition justifies using the PD metric to detect shortcut learning, as demonstrated in the following experiments.

3. Experiments

3.1. Not all spurious correlations are shortcuts

Fig-3 shows the Dominoes binary dataset (Kirichenko et al., 2022) with images of size 64×32. We create three pairs of datasets, with both easy and hard spurious features relative to the shared core feature (see Table-1). The classes used are 0,1 for MNIST and SVHN, coat,dress for FMNIST, and airplane, automobile for CIFAR10. Additionally, we include two classes from Kuzushiji-MNIST (KMNIST) and introduce a modified dataset called KMNpatch, which incorporates a spurious white patch (5x5 in top-left corner)

Figure 2. Examples of PD plots (for DenseNet-121) at different stages of the training. The red bar indicates samples with undefined PD, and the dotted vertical line shows the mean PD. The undefined samples (shown in red) slowly accumulate in layer 88 as training progresses. This is because the model needs more time to learn harder samples that accumulate at higher prediction depth, i.e., later layers.

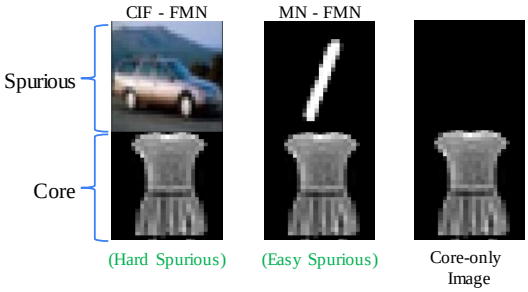

Figure 3: Dominoes Dataset

Table 1: Results for the Dominoes experiment averaged across 4-runs. Numbers in bracket show mean-PD (dataset difficulty). Core-only accuracy indicates the model's reliance on core features. Models achieve high core-only accuracy when spurious features are harder than core features.

Dataset (Spurious-Core)	Is spurious harder than core?	Test Accuracy	Core-only Accuracy
CIF(6.8) - FMN(3.9)	yes	99.12±0.27%	98.95±0.30%
MN(2.2) - FMN(3.9)	no	99.95±0.05%	50.75±2.96%
CIF(6.8) - KMN(5)	yes	98.91±0.16%	98.30±1.08%
MN(2.2) - KMN(5)	no	99.97±0.05%	50.48±2.64%
CIF(6.8) - MN(2.2)	yes	99.74±0.07%	99.5±0.66%
KMNpatch(1.1) - MN(2.2)	no	99.97±0.04%	68.78±20.03%

perfectly correlated with the target for one of the KMNIST classes. The ranking of dataset difficulty based on mean-PD is: KMNpatch(1.1) < MNIST(2.2) < FMNIST(3.9) < KMNIST(5) < SVHN(5.9) < CIFAR10(6.8). ResNet18 is used to measure test accuracy (sampled from same distribution) and core-only accuracy (by masking spurious feature). Higher core-only accuracy indicates lower reliance on spurious features.

Table 1 shows high test accuracy on all datasets. When spurious feature is harder to learn than core, the model heavily relies on core features (high core-only accuracy >98%). When spurious feature is easier than core, the model leverages them, causing the core-only accuracy to drop to random chance (50%). The KMNpatch-MN results show significantly higher core-only accuracy (69%) and standard deviation (explanation in Appendix-A.9). This shows that spurious features harder than core fail to act as a shortcuts.

3.2. Monitoring Initial Layers Can Reveal Suspicious Shortcut Learning Activity

Synthetic Shortcut on Toy Dataset: We demonstrate our method on the Kuzushiji-MNIST (KMNIST) (Clanuwat et al., 2018) dataset with ten classes and images of size 28×28. We insert a white patch (spurious feature) at a class-specific location and train VGG16 models on the KMNIST with a patch shortcut ($\mathcal{M}_{sh}$) and another on the original KMNIST without the patch ($\mathcal{M}_{orig}$).

Fig-4 shows that introducing the white patch makes KMNIST easier than even MNIST for $\mathcal{M}_{sh}$ (see Fig - 4A & 4D). The white patch is an easy feature, and hence the model only needs a single layer to detect it. The Grad-CAM maps for the layer-1 show that $\mathcal{M}_{sh}$ focuses mainly on the patch (see Fig-4D), and hence the test accuracy on the original KMNIST images is very low (∼8%). The PD plot for $\mathcal{M}_{orig}$ (see Fig-4E) is not as skewed toward lower depth as the plot for $\mathcal{M}_{sh}$. This is expected as $\mathcal{M}_{orig}$ is not looking at the spurious patch and therefore utilizes more layers to make the prediction. The mean PD for $\mathcal{M}_{orig}$ suggests that the original KMNIST is harder than Fashion-MNIST but easier than CIFAR10. $\mathcal{M}_{orig}$ also achieves a higher test accuracy (∼98%).

Real Shortcut on Medical Dataset: For this experiment, we use the NIH dataset (Wang et al., 2017a) which has the popular chest drain spurious feature (for Pneumothorax detection) (Oakden-Rayner et al., 2020). Chest drains are used to treat positive Pneumothorax cases and are therefore positively correlated with Pneumothorax and can be used by the deep learning model (Oakden-Rayner et al., 2020). We train a DenseNet121 model ($\mathcal{M}_{nih}$) for Pneumothorax

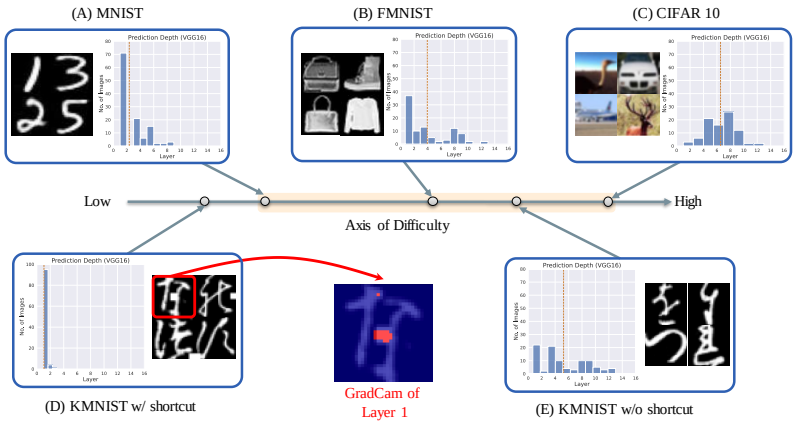

Figure 4. The datasets are ordered based on their difficulty (measured using mean PD shown by dotted vertical lines): KMN w/ sh.(1.1) < MNIST(2.2) < FMNIST(3.9) < KMN w/o sh.(5) < SVHN(5.9) < CIFAR10(6.8). The bottom row shows the effect of the shortcut on the KMNIST dataset. The yellow region on the axis indicates the expected difficulty of classifying KMNIST. While the original KMNIST lies in the yellow region, the shortcut significantly reduces the task difficulty. The Grad-CAM shows that the model focuses on the spurious patch.

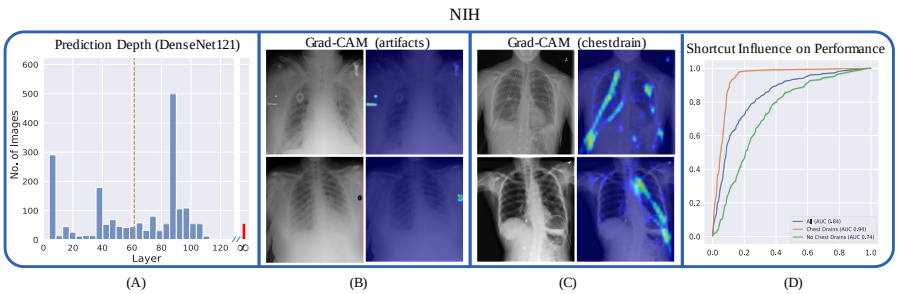

Figure 5. Shortcut learning on NIH dataset. (A) PD plot for DenseNet-121 trained on NIH shows prominent peaks in the initial layers. (B, C) Grad-CAM reveals that the initial layers use irrelevant artifacts and chest drains as shortcuts for classification. (D) The chest drain spurious feature affects the AUC performance of the model. The X-axis (Y-axis) shows the false positive (true positive) rate.

detection on NIH images of size 128×128. See Appendix-A.7 for more details.

Fig-5A shows the PD plot for $\mathcal{M}_{nih}$. We see suspicious peaks at the initial layers. Pneumothorax classification is challenging even for radiologists; hence, peaks at the initial layers raise suspicion. The Grad-CAM maps in Figs-5B & 5C reveal that the initial layers look at irrelevant artifacts and chest drains in the image. This provides evidence for shortcut learning happening in the initial layers. Fig-5D shows that the AUC performance is 0.94 when the diseased patients have a chest drain and 0.74 when they don't. In both cases, the set of healthy patients remains the same. This observation is consistent with the findings of Oakden-Rayner et al. (2020) and indicates that the model looks at chest drains to classify positive Pneumothorax cases.

The above experiments demonstrate how a peak located in the initial layers of the PD plot should raise suspicion, especially when the classification task is challenging. Visualization techniques like Grad-CAM can further help identify the shortcuts being learned by the model. This approach works well even for realistic and challenging spurious features (like chestdrains), as shown above. The appendix shows additional results on vision datasets (A.6), how shortcuts can be detected early during training (A.4), and how datasets with easy spurious features have more "usable information" (Ethayarajh et al., 2021) (A.5).

4. Conclusion

"*Potential shortcuts can be found by monitoring the easy features learned by the initial layers of a DNN early during the training.*" We validate this hypothesis on real medical and vision datasets. We also show that shortcuts are also learned quite early during the training. Further, we show a theoretical connection between PD and $\mathcal{V}$-information to support our empirical results. Datasets with spurious features have more $\mathcal{V}$-information causing the model to

learn the shortcut. Lastly, relying only on accuracy plots is insufficient, and we need to monitor instance difficulty metrics during training to detect shortcut learning patterns.

References

Adnan, M., Ioannou, Y., Tsai, C.-Y., Galloway, A., Tizhoosh, H., and Taylor, G. W. Monitoring shortcut learning using mutual information. *arXiv preprint arXiv:2206.13034*, 2022.

Baldock, R., Maennel, H., and Neyshabur, B. Deep learning through the lens of example difficulty. *Advances in Neural Information Processing Systems*, 34:10876–10889, 2021.

Bellamy, D., Hernán, M. A., and Beam, A. A structural characterization of shortcut features for prediction. *European Journal of Epidemiology*, 37(6):563–568, 2022.

Chaudhuri, K. and Dasgupta, S. Rates of convergence for nearest neighbor classification. *Advances in Neural Information Processing Systems*, 27, 2014.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

DeGrave, A. J., Janizek, J. D., and Lee, S.-I. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

Ethayarajh, K., Choi, Y., and Swayamdipta, S. Information-theoretic measures of dataset difficulty. *arXiv preprint arXiv:2110.08420*, 2021.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Gühring, I., Raslan, M., and Kutyniok, G. Expressivity of deep neural networks. *arXiv preprint arXiv:2007.04759*, 2020.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 590–597, 2019.

Jain, S., Agrawal, A., Saporta, A., Truong, S. Q., Duong, D. N., Bui, T., Chambon, P., Zhang, Y., Lungren, M. P., Ng, A. Y., et al. Radgraph: Extracting clinical entities and relations from radiology reports. *arXiv preprint arXiv:2106.14463*, 2021.

Johnson, A. E., Pollard, T. J., Greenbaum, N. R., Lungren, M. P., Deng, C.-y., Peng, Y., Lu, Z., Mark, R. G., Berkowitz, S. J., and Horng, S. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Mangalam, K. and Prabhu, V. U. Do deep neural networks learn shallow learnable examples first? 2019.

Oakden-Rayner, L., Dunnmon, J., Carneiro, G., and Ré, C. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM conference on health, inference, and learning*, pp. 151–159, 2020.

Puli, A. M., Zhang, L. H., Oermann, E. K., and Ranganath, R. Out-of-distribution generalization in the presence of nuisance-induced spurious correlations. In *International Conference on Learning Representations*, 2022.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pp. 5301–5310. PMLR, 2019.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017a.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017b.

Wiles, O., Gowal, S., Stimberg, F., Alvise-Rebuffi, S., Ktena, I., Cemgil, T., et al. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

Xu, Y., Zhao, S., Song, J., Stewart, R., and Ermon, S. A theory of usable information under computational constraints. *arXiv preprint arXiv:2002.10689*, 2020.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Zhang, X., Zhou, L., Xu, R., Cui, P., Shen, Z., and Liu, H. Nico++: Towards better benchmarking for domain generalization. *arXiv preprint arXiv:2204.08040*, 2022.

A. Appendix

A.1. Proof of Proposition-1

Proposition A.1. *Given two datasets, D_s with spurious features and D_i without them, we assume the following:*

1. *(Well-Trained Model Assumption) The part of the network from any representation to the label is one of the functions that compute $\mathcal{V}$-information.*
2. *(Function Class Complexity Assumption) Assume that there exists a $K \in \{1, N\}$ such that V_{cnn} of depth $N - K$ is deep enough to be a strictly larger function class than V_{knn} with a fixed neighbor size (29 in this paper). Assume that this V_{knn} is a larger function class than a linear function.*
3. *(Controlled Confidence Growth Assumption) For both datasets $D \in \{D_s, D_i\}$, assume that the for all $k \in \{1, \cdots, N\}$,*
$$\tau \leq \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_k) - \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_{k-1}) \leq \epsilon$$
4. *(Prediction Depth Separation Assumption) Let L be an integer such that, $L \leq K$ and $L < N - \psi \max_y (-\log p(Y = y))$. Note that $p(Y = y)$ is simply the prevalence of class y. Let there exist a gap in prediction depths of samples in D_s and D_i: $\psi \in (0, 0.5)$ such that $1 - \psi$ fraction of D_s has prediction depth $\leq L$ and $1 - \psi$ fraction of D_i has prediction depth $> K$.*

Then, for a model class of N-layer CNNs, we show that the $\mathcal{V}_{cnn}$-information for D_s is greater than $\mathcal{V}_{cnn}$-information for D_i:
$$\mathcal{I}^{D_s}_{\mathcal{V}_{cnn}}(X \to Y) \geq \mathcal{I}^{D_i}_{\mathcal{V}_{cnn}}(X \to Y)$$

Proof. Before proceeding to the proof, we attempt to justify and reason about the above assumptions.

Assumption-1 states a property of trained neural networks in the context of usable information. Let $f(X)$ be a trained neural network. Consider splitting the network into the representation $\phi_k(X)$ at the k^{th} layer and the rest of the network as a function applied to $\phi_k(X)$: i.e., $f(X) = f_k \circ \phi_k(X)$. Then we assume that $f_k(\cdot)$ is the function that achieves the $\mathcal{V}_{\text{cnn of size(n-k)}}$-information between $\phi_k(X)$ and Y (Ethayarajh et al., 2021; Xu et al., 2020). The function that computes $\mathcal{V}$-information must achieve a minimum cross-entropy (Ethayarajh et al., 2021). So if we train $f(X)$ by minimizing the cross-entropy loss, $f_k(.)$ must converge to a function that achieves the $\mathcal{V}_{\text{cnn of size(n-k)}}$-information between $\phi_k(X)$ and Y.

Assumption-2 implies that the CNN class in $\mathcal{V}_{cnn}$ is deep enough such that the network after the K^{th} layer can approximate a k-NN classifier with 29 neighbors; (K here is same as the K in assumption-4). This is also a reasonable assumption (Chaudhuri & Dasgupta, 2014; Gühring et al., 2020). Chaudhuri & Dasgupta (2014) lower bounds the error for k-NN classifiers for a fixed k, and Gühring et al. (2020) shows the depth expressivity of CNN classifiers. Assumption-3 states that the difference in $\mathcal{V}_{knn}$-information between intermediate layers does not explode indefinitely and thus can be bounded by some positive quantities τ and ϵ.

Assumption-4 is also easily satisfied. For example, if the smallest prevalence class in the dataset has a prevalence greater than $\frac{1}{1000}$, then assumption-4 boils down to saying $L < N - 0.5 * \max_y (-\log p(Y = y)) = N - 3.45$, where L is the low PD value caused by spurious features in D_s, and N is the total number of layers in the CNN. All our datasets satisfy the class prevalence $> \frac{1}{1000}$ constraint. Even diseases like pneumothorax which are rare, have a class prevalence of at least $\frac{1}{30}$ in both NIH and MIMIC-CXR. And $L < N - 3.45$ is easily satisfied in all our experiments. For e.g., see Fig-5 where 80% (or $1 - \psi = 0.8$) of the samples have PD ≤ 16. So $L = 16, N = 121$(for Densenet-121) easily satisfies $16 < 121 - 3.45$.

Now we elaborate on the proof of the proposition given the four assumptions. We proceed in two parts: first, we lower bound $\mathcal{V}_{cnn}$-information for D_s, and then we upper bound $\mathcal{V}_{cnn}$ for D_i.

Assumption 3 implies:
$$(B - A)\tau \leq \sum_{k=A}^{B} \tau \leq \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_k) - \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_{k-1}) \leq (B - A)\epsilon$$

Note: (A, B) are just placeholders for the min and max indices over which the summation is defined. They are replaced by $(L + 1, K)$ and $(K + 1, N)$ below while trying to lower bound $\mathcal{I}^{D_s}_{\mathcal{V}_{cnn}}$ and upper bound $\mathcal{I}^{D_i}_{\mathcal{V}_{cnn}}$ respectively.

PD - PVI connection. Note that by definition, when the prediction depth is k for a sample X, then $PVI_{knn}(\phi_k(X)) \geq \delta$ but $PVI_{knn}(\phi_{k-1}(X)) < \delta$. This follows from how we compute PD (see Section-2 in the main paper, and Appendix-A.8).

Lower bounding $\mathcal{I}^{D_s}_{\mathcal{V}_{cnn}}$

$$
\begin{aligned}
\mathcal{I}^{D_s}_{\mathcal{V}_{cnn}} &= \mathcal{I}^{D_s}_{\mathcal{V}_{cnn \text{ of depth } N-K}}(\phi_K) && \{\text{Assumption-1}\} \\
&\geq \mathcal{I}^{D_s}_{\mathcal{V}_{knn}}(\phi_K) && \{\text{Assumption-2}\} \\
&= \mathcal{I}^{D_s}_{\mathcal{V}_{knn}}(\phi_L) + \sum_{k=L+1}^{K} \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_k) - \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_{k-1}) && \{\text{Telescoping Sum}\} \\
&\geq \mathcal{I}^{D_s}_{\mathcal{V}_{knn}}(\phi_L) + (K-L)\tau && \{\text{Assumption-3}\} \\
&\geq \psi \min_{X,Y \in D_s, pd >= L} PVI_{knn}(X \to Y) \\
&\quad + (1-\psi) \min_{X,Y \in D_s, pd < L} PVI_{knn}(X \to Y) + (K-L)\tau && \{\text{Prediction Depth Separation}\} \\
&\geq 0 * \psi + \delta * (1-\psi) + (K-L)\tau && \{\text{Prediction Depth Separation}\}
\end{aligned}
$$

Upper bounding $\mathcal{I}^{D_i}_{\mathcal{V}_{cnn}}$

$$
\begin{aligned}
\mathcal{I}^{D_i}_{\mathcal{V}_{cnn}} &\leq \mathcal{I}^{D_i}_{\mathcal{V}_{knn}}(\phi_N) && \{\text{Assumption-2}\} \\
&= \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_K) + \sum_{k=K+1}^{N} \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_k) - \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_{k-1}) && \{\text{Telescoping Sum}\} \\
&\leq \mathcal{I}^{D}_{\mathcal{V}_{knn}}(\phi_K) + (N-K)\epsilon && \{\text{Assumption-3}\} \\
&\leq (N-K)\epsilon + \psi \max_{X,Y \in D_i, pd(X) \leq K} PVI^{D}_{\mathcal{V}_{knn}}(\phi_K(X) \to Y) \\
&\quad + (1-\psi) \max_{X,Y \in D_i, pd(X) > K} PVI^{D}_{\mathcal{V}_{knn}}(\phi_K(X) \to Y) && \{\text{Prediction Depth Separation}\} \\
&\leq (N-K)\epsilon + \psi \max_{y} (-\log p(Y=y)) \\
&\quad + (1-\psi) \max_{X,Y \in D_i, pd(X) > K} PVI^{D}_{\mathcal{V}_{knn}}(\phi_K(X) \to Y) && \{\text{PVI} \leq -\log p(Y=y)\} \\
&\leq (N-K)\epsilon + \psi \max_{y} (-\log p(Y=y)) + (1-\psi)\delta && \{\text{PD-PVI connection for pd} > K\}
\end{aligned}
$$

The proof follows by comparing the lower bound on $\mathcal{I}^{D_s}_{\mathcal{V}_{cnn}}$ and the upper bound on $\mathcal{I}^{D_i}_{\mathcal{V}_{cnn}}$. Intuitively what this means is that when there is a sufficiently large gap in the mean PD between D_s and D_i, then the $\mathcal{V}$-information of D_s exceeds the $\mathcal{V}$-information of D_i, which is why the model prefers learning the spurious feature and using them as shortcuts rather than using the core features for the task.

□

A.2. Grad-CAM Visualization

PD plots help us understand the model layers actively used for classifying different images. To further aid our intuition, we visualize the Grad-CAM outputs for the model's arbitrary layer k by attaching a soft-KNN head. Let g_{knn} denote the soft and differentiable version of k-NN. We compute g_{knn} as follows:

$$
g_{knn}(\phi^k_q; \phi^k_{i \in \{1,2,...m\}}) = \frac{\sum_{j \in \mathcal{N}(\phi^k_q, 1)} \exp^{-\|\phi^k_q - \phi^k_j\|/s}}{\sum_{j \in \mathcal{N}(\phi^k_q, :)} \exp^{-\|\phi^k_q - \phi^k_j\|/s}}
$$

This function makes the KNN differentiable and can be used to compute Grad-CAM (Selvaraju et al., 2017). We use the $\mathcal{L}_1$ norm for all distance computations. ϕ^k_q corresponds to feature at layer-k for query image x_q. Let $\phi^k_{i \in \{1,2,...m\}}$ be the training data for KNN. Let $\mathcal{N}$ denote the neighborhood function. $\mathcal{N}(\phi^k_q, :)$ returns the indices of K-nearest neighbors for ϕ^k_q. $\mathcal{N}(\phi^k_q, 1)$ returns indices of images with positive label ($y = 1$) from the set of K-nearest neighbors for ϕ^k_q. s is the median for the set of $\mathcal{L}_1$ norms $\{\|\phi^k_q - \phi^k_j\|\}$ for $j \in \mathcal{N}(\phi^k_q, :)$.

A.3. Semi-Synthetic Shortcut on Medical Datasets:

We follow the procedure by DeGrave et al. (2021) to create the ChestX-ray14/GitHub-COVID dataset. This dataset comprises Covid19 positive images from Github Covid repositories and negative images from ChestX-ray14 dataset (Wang et al., 2017b). In addition, we also create the Chex-MIMIC dataset following the procedure by Puli et al. (2022). This dataset comprises 90% images of Pneumonia from Chexpert (Irvin et al., 2019) and 90% healthy images from MIMIC-CXR (Johnson et al., 2019). We train two DenseNet121 models, $\mathcal{M}_{covid}$ on the ChestX-ray14/GitHub-COVID dataset, and $\mathcal{M}_{chex}$ on the Chex-MIMIC dataset. We use DenseNet121, a common and standard architecture for medical image analysis. Images are resized to 512×512.

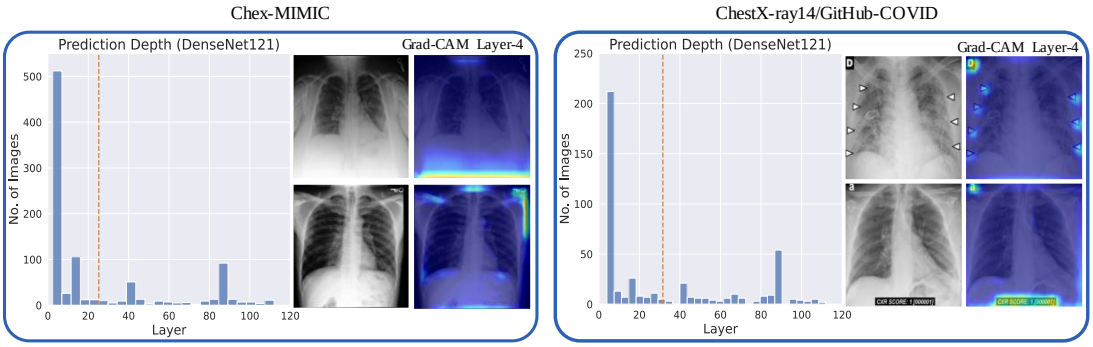

Figure 6. PD plots for two DenseNet-121 models trained on Chex-MIMIC and ChestX-ray14/GitHub-COVID datasets are shown in the figure, along with their corresponding Grad-CAM visualizations. Both PD plots exhibit a very high peak in the initial layers (1 to 4), indicating that the models use very easy features to make the predictions.

Fig-6 shows the PD plots for $\mathcal{M}_{chex}$ and $\mathcal{M}_{covid}$. Both the plots are highly skewed towards initial layers, similar to the KMNIST with patch shortcut in Fig-4D. This again indicates that the models are using very easy features to make the predictions, which is counterintuitive as the two tasks (pneumonia and covid19 detection) are hard tasks even for humans. Examining the Grad-CAM maps at layer-4 reveals that these models focus on irrelevant spurious features outside the lung region. This raises concern because both diseases are known to affect mainly the lungs. The reason for this suspicious behavior is that, in both these datasets, the healthy and diseased samples have been acquired from two different sources. This creates a spurious feature because source-specific attributes or tokens are predictive of the disease and can be easily learned, as pointed out by DeGrave et al. (2021). On the other hand, we don't observe this skewed distribution in the NIH with chestdrain experiment (Sec-3) because all the images come from a single dataset (NIH).

A.4. Detecting Shortcuts Early

In this experiment, we show how shortcuts can often be detected relatively early during training. This is because initial layers which learn the shortcuts converge very early during the training. We observe this by monitoring PD plots across training epochs. In all of our experiments, the shortcut is revealed by the PD plot within two epochs of training.

Fig-7 shows the evolution of the PD plot across epochs for $\mathcal{M}_{nih}$ (which is the model used in Fig-5). This visualization helps us observe the training dynamics of the various layers. The red bar in the PD plots shows the samples with undefined prediction depths.

These plots reveal several useful insights into the learning dynamics of the model. Firstly, we see three prominent peaks in epoch-1 at layers-4,40,88 (see Fig-7A). The magnitude of the initial peaks (like layers-4&40) remains nearly constant throughout the training. These peaks correspond to shortcuts, as discussed in the previous section. This indicates that easy shortcuts can often be identified early (epoch-1 in this case). Fig-8 shows the PD plots at epoch-2 for other datasets with shortcuts. It is clear from Fig-8 that the suspiciously high peak at the initial layer is visible in the *second epoch* itself. The Grad-CAM maps reveal that this layer looks at irrelevant artifacts in the dataset. This behavior is seen in all datasets shown in Fig-8.

Secondly, we also see that accuracy or AUC plots do not reveal shortcut learning patterns. We need to monitor the training dynamics using suitable metrics (like PD) to detect this behavior. Thirdly, the red peak (undefined samples) decreases in

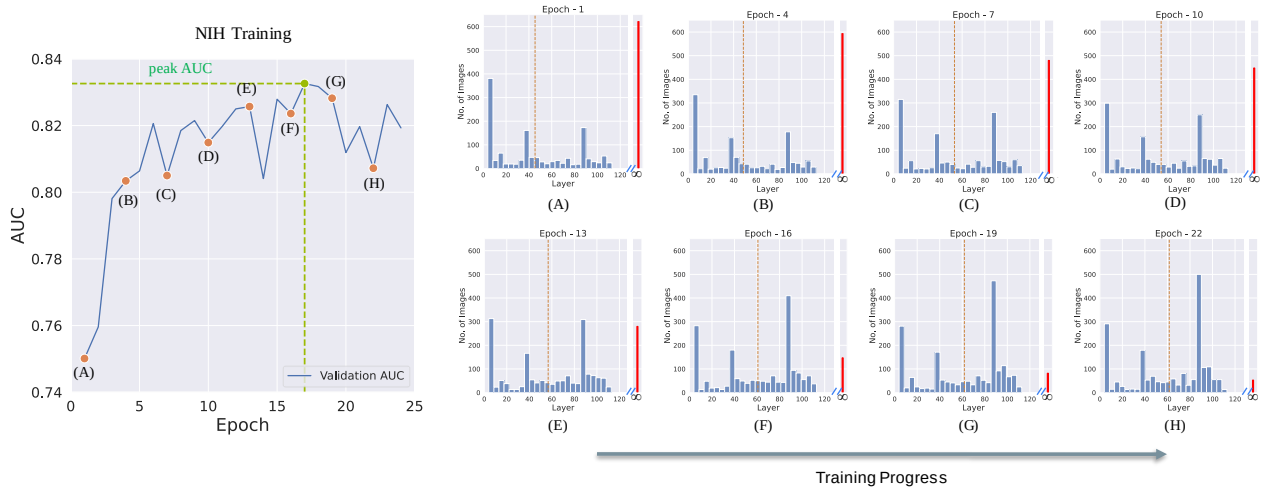

Figure 7. Evolution of PD plot across epochs shows the training dynamics of the DNN on the NIH dataset. The initial peaks (layers-4&40) are relatively stable throughout training, whereas the later peaks (layer-88) change with time. The initial layers learn the easy shortcuts, which can be detected early during the training. Samples with undefined PD (shown in red) take more time to converge and eventually accumulate in the later layers (layer 88 in this case).

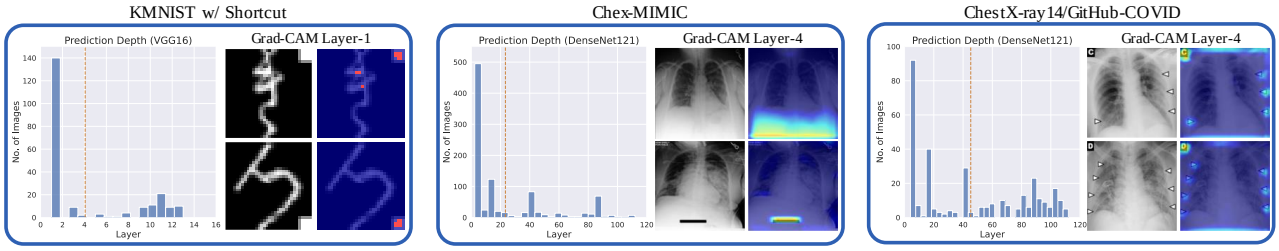

Figure 8. Epoch-2 PD plots for various datasets with shortcuts. The high spurious peak in the initial layer is visible in all the datasets indicating that shortcuts can be detected early during the training.

magnitude with time, and we see a proportional increase in the layer-88 peak. This corroborates well with the observation that later layers take more time to converge (Rahaman et al., 2019; Mangalam & Prabhu, 2019; Zeiler & Fergus, 2014). Therefore, samples with higher PD are initially undefined and do not appear in the PD plot. Nonetheless, samples with lower PD show up very early during the training, which helps us detect shortcuts early. Early detection can consequently help develop intervention schemes that fix the shortcut early.

A.5. Prediction Depth $\approx \mathcal{V}$-Usable Information

In this experiment, we show that datasets with easy spurious features have more "usable information" (Ethayarajh et al., 2021) compared to their counterparts without such features. Due to higher usable information, the model requires fewer layers to classify the images with spurious features. We use this experiment to empirically justify Proposition-1 outlined in Appendix-A.1.

Table-2 measures the influence of spurious features on NIH and KMNIST using PD and PVI metrics. All diseased patients in the "NIH w/ Spurious feat." dataset have a chest drain, whereas all diseased patients in the "NIH w/o Spurious feat." dataset have no chest drain. The set of healthy patients is common for the two datasets. The KMNIST datasets are the same as those used in Section-3.2. We use VGG16 for KMNIST and DenseNet121 for NIH. Other training details are the same as Section-3.2.

Table-2 shows that datasets with spurious features (D_s) have smaller mean PD values than their counterparts without such

Table 2. Effect of Spurious features on Prediction Depth and the negative conditional $\mathcal{V}$-entropy (-$\boldsymbol{H}_{\boldsymbol{\mathcal{V}_{cnn}}}(Y \mid X)$). The label marginal distributions are the same with or without the spurious feature, and thus the negative conditional $\mathcal{V}$-entropy is proportional to $\mathcal{V}$-information.

Dataset	mean PD	$-\boldsymbol{H}_{\boldsymbol{\mathcal{V}_{cnn}}}(Y \mid X)$
NIH w/ Spurious feat.	53.43	-0.1171
NIH w/o Spurious feat.	75.33	-0.2321
KMNIST w/ Spurious feat.	1.06	-0.0024
KMNIST w/o Spurious feat.	5.25	-0.0585

features (D_i). Proposition-1 (see Section-2, Appendix-A.1) shows that a sufficient gap between the mean PDs of D_s and D_i causes the $\mathcal{V}$-Information of D_s to be greater than D_i. Table-2 confirms this in a medical-imaging dataset with a real chest drain spurious feature, and we see that the mean "usable information" increases when there is a spurious feature. This implies that the model learns spurious features as they have more usable information than the core features. Ethayarajh et al. (2021) also show that $\mathcal{V}$-information is positively correlated with test accuracy. This explains the significant change in AUC observed in Fig-5D. Proposition 1 bridges the gap between the notions of PD and $\mathcal{V}$-usable information. This connection between $\mathcal{V}$-information and PD indicates that monitoring early training dynamics using PD not only helps detect shortcut learning but also bears insights into the dataset's difficulty (in information-theoretic terms) for a given model class.

We further investigate this relationship on four additional datasets: KMNIST, FMNIST, SVHN, and CIFAR10. We train a VGG16 model on these datasets for ten epochs using an Adam optimizer and a base learning rate of 0.01. We use a bar plot to show the correlation between PD and $\mathcal{V}$-entropy. We group PD into intervals of size four and compute the mean $\mathcal{V}$-entropy for samples lying in this PD interval.

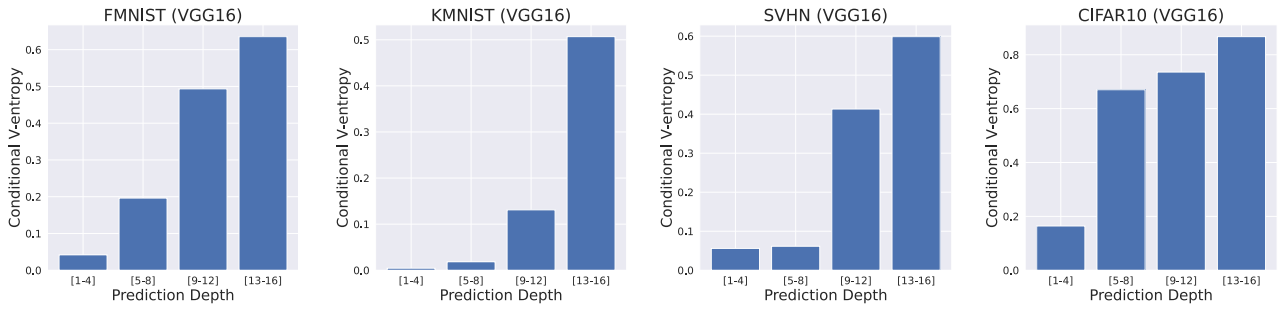

Figure 9. The bar plots show a positive correlation between PD and Conditional $\mathcal{V}$-entropy. Samples with higher PD also have a higher $\mathcal{V}$-entropy resulting in lower usable information for models like VGG16.

We again find that PD is positively correlated with $\mathcal{V}$-information. Instance difficulty increases with PD, and the usable information decreases with an increase in $\mathcal{V}$-entropy. It is, therefore, clear from Fig-9 that samples with a higher difficulty (PD value) have lower usable information, which is not only intuitive but also provides empirical support to Proposition-1 in Appendix-A.1.

A.6. Vision Experiments

We use the *NICO++* (Non-I.I.D. Image dataset with Contexts) dataset Zhang et al. (2022) to create multiple spurious datasets (Cow vs. Bird; Dog vs. Lizard) such that the context/background is spuriously correlated with the target. NICO++ is a Non-I.I.D image dataset that uses context to differentiate between the test and train distributions. This forms an ideal setup to investigate what spurious correlations the model learns during training. We follow the procedure outlined by (Puli et al., 2022) to create datasets with spurious correlations (90% prevalence) in the training data. The test data has the relationship between spurious attributes and the true labels flipped. This is similar to the Chex-MIMIC dataset illustrated in section-A.3. We test our hypothesis using ResNet-18 and VGG16. We train our models for 30 epochs using an Adam optimizer and a base learning rate of 0.01. We choose the best checkpoint using early stopping.

Figures-10,12 show PD plots and train/test accuracies for models that learn the spurious background feature present in the

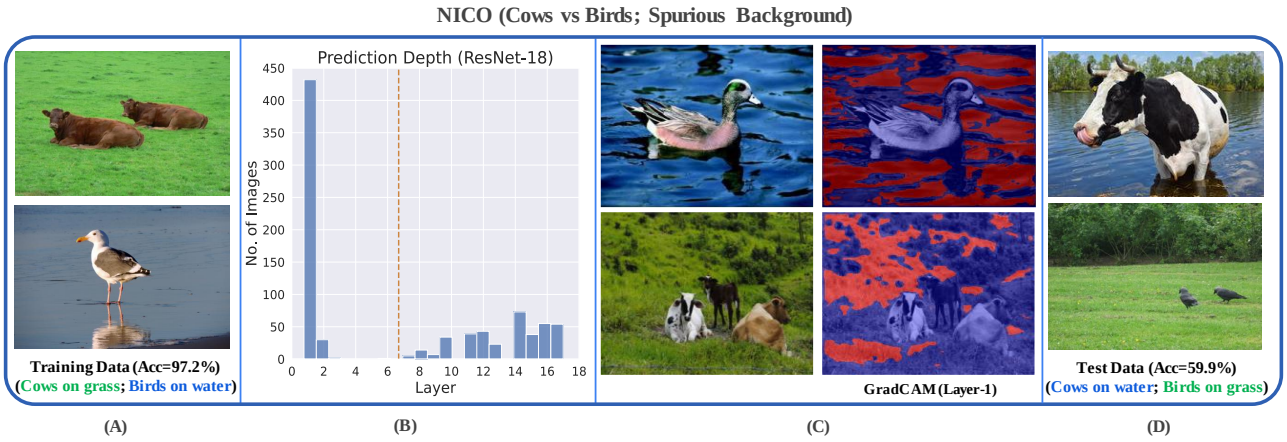

Figure 10. Cow vs. Birds classification on NICO++ dataset. (A) Training data contains images of cows on grass and birds on water (correlation strength=0.9). The model achieves 97.2% training accuracy. (B) PD plot for ResNet-18 reveals a spurious peak at layer-1, indicating the model's heavy reliance on very simple (potentially spurious) features. (C) GradCAM plots for layer 1 reveal that the model mainly relies on the spurious background to make its predictions. (D) Consequently, the model achieves a test accuracy of only 59.9% on test data where the spurious correlation is flipped (i.e., cows (birds) are found on water (grass)).

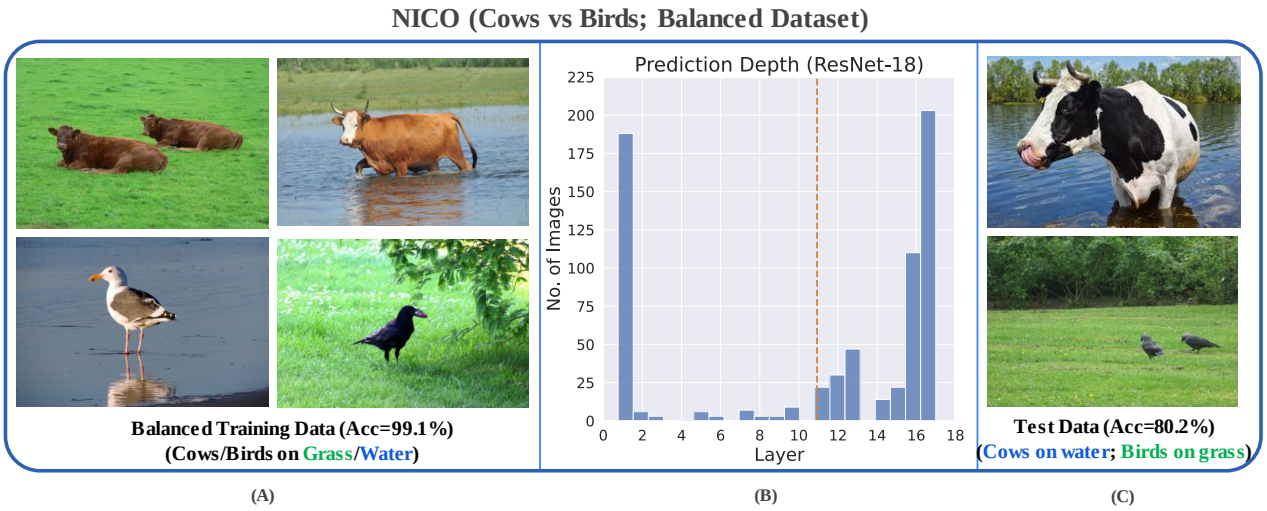

Figure 11. Balanced dataset for Cow vs. Birds classification task on NICO++ dataset. (A) The training dataset contains a balanced distribution of cows and birds found on water and grass (each group has an equal number of images). (B) The balanced dataset shifts the PD plot towards the later layers (compared to Fig-10B, indicating that the model relies lesser on spurious features. (C) This consequently results in an improved test accuracy of 80.2% (as compared to 59.9% in Fig-10D for the spurious dataset).

NICO++ dataset. While all models achieve $> 85\%$ training accuracy, they have poor accuracies (50%) on the test data where the spurious correlation is flipped. This can be seen simply by observing the PD plots for the model on the training data. The plots are skewed towards the initial layers indicating that the model relies heavily on very simple (potentially spurious) features for the task. GradCAM maps also confirm that the model often focuses on the background context rather than the foreground object of interest.

We further observe in Fig-11 that balancing the training data (to remove the spurious correlation) results in a model with improved test accuracy (80.2%) as expected. This is also reflected in the PD plot (Fig-11B), where we see that the distribution of the peaks, as well as the mean PD, shift proportionately towards the later layers, indicating that the model now relies lesser on the spurious features.

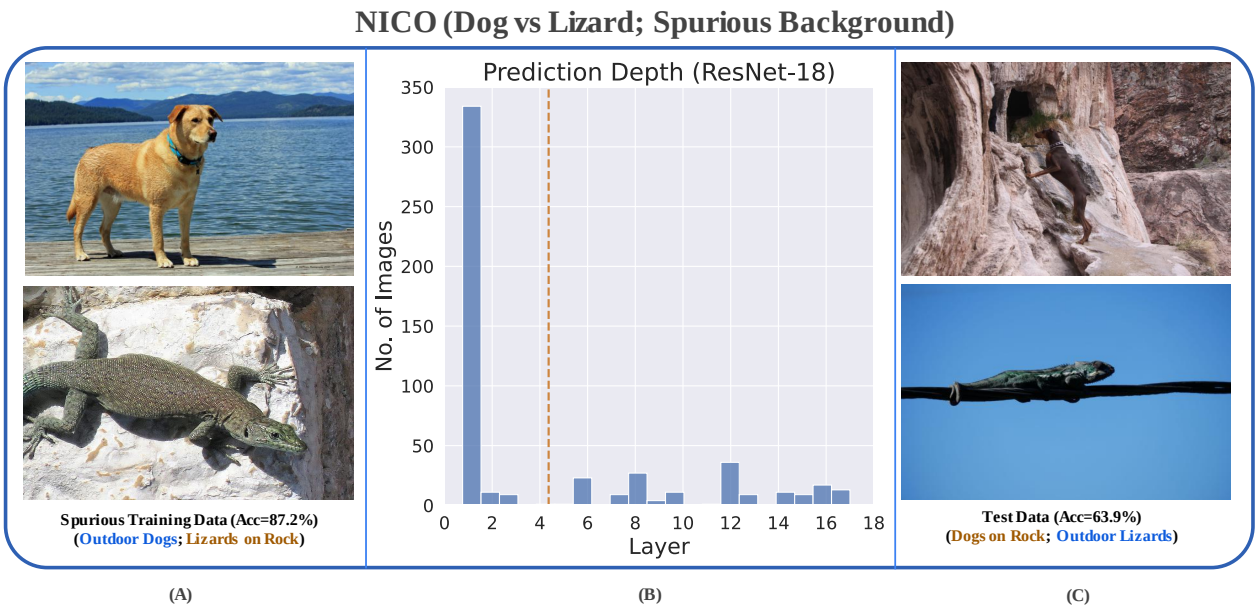

Figure 12. Dog vs. Lizard classification with a spurious background feature on NICO++ dataset. (A) Training data contains images of outdoor dogs and lizards on rock (correlation strength=0.9). The spurious background color/texture reveals the foreground object. The model achieves 87.2% training accuracy. (B) PD plot for ResNet-18 reveals a spurious peak at layer-1, indicating the model's reliance on simple (potentially spurious) features. (C) The low test accuracy confirms this (63.9%). The test data has the spurious correlation flipped (i.e., images contain dogs on rock and lizards found outdoors.)

By monitoring PD plots during training and using suitable visualization techniques, we show that one can obtain useful insights about the spurious correlations that the model may be learning. This can also help the user make an educated guess about the generalization behavior of the model during deployment.

A.7. Chest Drain Annotations for NIH Dataset

To reproduce the results by Oakden-Rayner et al. (2020), we need chest drain annotations for the NIH dataset (Wang et al., 2017a), which is not natively provided. To do this, we use the MIMIC-CXR dataset (Johnson et al., 2019), which has rich meta-data information in radiology reports. We collaborate with radiologists to identify terms related to Pneumothorax from the MIMIC-CXR reports. These include pigtail catheters, pleural tubes, chest tubes, thoracostomy tubes, etc. We collect chest drain annotations for MIMIC-CXR by parsing the reports for these terms using the RadGraph NLP pipeline (Jain et al., 2021). Using these annotations, we train a DenseNet121 model to detect chest drains relevant to Pneumothorax. Finally, we run this trained model on the NIH dataset to obtain the needed chest drain annotations. We use these annotations to get the results shown in Fig - 5D, which closely reproduces the results obtained by Oakden-Rayner et al. (2020).

A.8. Notion of Undefined Prediction Depth

The PD is simply the earliest layer, after which all subsequent k-NN predictions remain the same (0 or 1) Baldock et al. (2021).

$$\text{PD} = \min \arg\max_{n} \left[\prod_{i=n}^{N} f_{knn}(\phi^i) + \prod_{i=n}^{N} (1 - f_{knn}(\phi^i)) \right],$$

f_{knn} is a k-NN classifier that outputs 0 or 1 based on a given threshold, ϕ^i is the feature embedding for the given input at layer-i, and N is the index of the final layer of the model. The lower the PD of input, the easier it is to classify. We also use the notion of undefined PD to work with models that are not fully trained. We treat k-NN predictions close to 0.5 (for a binary classification setting) as invalid. If the k-NN predictions for the last three layers (for a given input to the model) are

invalid, we treat the PD of the input as undefined.

While fully trained models give valid PD values, our application requires working with arbitrary deep-learning models that are not necessarily fully trained. We, therefore, introduce the notion of undefined PD by treating k-NN predictions close to 0.5 (for a binary classification setting) as invalid. We define a δ such that $|f_{knn}(x) - 0.5| < \delta$ implies an invalid k-NN output. We use $\delta = 0.1$ and $k = 29$ in our experiments. If any k-NN predictions for the last three layers are invalid, we treat the PD of the input image to be undefined. To work with high-resolution images (like 512×512), we downsample the spatial resolution of all training embeddings to 8×8 before using the k-NN classifiers on the intermediate layers. We empirically see that our results are insensitive to k in the range $[5, 30]$.

A.9. A PD Perspective for Feature Learning

Table 1 shows that the core-only accuracy stays high (>98%) for datasets where the spurious feature is harder to learn than the core feature. When the spurious feature is easier than the core, the model learns to leverage them, and hence the core-only accuracy drops to nearly random chance (∼50%). Interestingly, the KMNpatch-MN results have a much higher core-only accuracy (∼69%) and a more significant standard deviation. This is because the choice of features that the model chooses to learn depends on the PD distributions of the core and spurious features. We provide three different perspectives on why KMNpatch-MN runs have better results.

PD Distribution Perspective: The KMNpatch-MN domino dataset has a smaller difference in the core-spurious mean PDs ($2.2 - 1.1 = 1$), as compared to other datasets (for e.g., MN-KMN has a difference of $5 - 2.2 = 2.8$ in their mean PDs). The closer the PD distributions of the core and spurious features are, the more the model treats them equivalently. Therefore, in the case of the KMNpatch-MN, we empirically observe that different initializations (random seeds) lead to different choices the model makes in terms of core or spurious features. This is why the standard deviation of KMNpatch-MN is high (20.03) compared to the other experiments.

Theoretical Perspective (Proposition-1): This is not surprising and, in fact, corroborates quite well with Proposition-1 in Appendix-A.1. The Prediction Depth Separation Assumption suggests that without a sufficient gap in the mean PDs of the core and spurious features, one cannot concretely assert anything about their ordinal relationship in terms of their usable information. In other words, spurious features will have higher usable information (for a given model) than the core features only if the spurious features have a sufficiently lower mean PD as compared to the core features. On the other hand, as the core and spurious features become comparable in terms of their difficulty, the model begins to treat them equivalently.

Loss Landscape Perspective: *(this is a conjecture; we do not have empirical evidence)* The loss landscape is a function of the model and the dataset. The solutions in the landscape that are reachable by the model depend on the optimizer and the training hyperparameters. Given a model and a set of training hyperparameters, we conjecture that the diversity (in terms of the features that the model learns during training) of the solutions in the landscape increase as the distance (difference in mean PD) between the core and spurious features decreases. This diversity manifests as the model's choice of using core vs. spurious features and could potentially result in a higher standard deviation of core-only accuracy across initializations.

A.10. Code Reproducibility

The code, along with the Python package list and environment files, will be made available upon acceptance.