MEASURING THE LAYER-WISE IMPACT OF IMAGE SHORTCUTS ON DEEP MODEL FEATURES

Nikita Tsoy & Nikola Konstantinov
INSAIT, Sofia University "St. Kliment Ohridski"
Sofia, Bulgaria
nikita.tsoy@insait.ai

ABSTRACT

Shortcuts, spurious patterns that perform well only on the training distribution, pose a major challenge to deep network reliability (Geirhos et al., 2020). In this work, we investigate the layer-wise impact of image shortcuts on learned features. First, we propose an experiment design that introduces artificial shortcut-inducing skews during training, enabling a counterfactual analysis of how different layers contribute to shortcut-related accuracy degradation. Next, we use our method to study the effects of a patch-like skew on CNNs trained on CIFAR-10 and CIFAR-100. Our analysis reveals that different types of skews affect networks layers differently: class-universal skews (affecting all instances of a target class) and class-specific skews (affecting only one class) impact deeper layers more than non-universal and non-specific skews, respectively. Additionally, we identify the forgetting of shortcut-free features as a key mechanism behind accuracy drop for our class of skews, indicating the potential role of simplicity bias (Shah et al., 2020) and excessive regularization (Sagawa et al., 2020) in shortcut learning.

1 INTRODUCTION

Shortcuts, spurious rules that only hold on the training distribution but do not generalize to real-world scenarios, present an important concern for the reliability of deep networks. Despite their prevalence, the mechanisms behind shortcut learning and their influence on learned representations are still unclear. While, from a statistical perspective, shortcuts represent a well-known statistical phenomenon of spurious correlations (Arjovsky et al., 2020), it remains unclear what correlations deep models capture during training and how these correlations shape learned feature representations (Hermann & Lampinen, 2020).

One of the overlooked aspects of deep learning in relation to shortcuts is the hierarchical nature of deep features. Since the different layers of a network correspond to different levels of abstraction (Simonyan et al., 2014), shortcuts likely affect layers in distinct ways. Consequently, layers may have different degrees of responsibility for shortcut learning. A quantitative understanding of this phenomenon could help design layer-specific strategies for mitigating distribution shifts (e.g. Lee et al., 2023). However, existing works do not precisely quantify this phenomenon and either only examine effects on the overall model accuracy without considering layer-wise effects (Scimeca et al., 2022) or only study the effects on feature representations and do not explicitly link these results with the final validation accuracy (Hermann & Lampinen, 2020; Islam et al., 2021).

Contributions To bridge this gap, we develop a method for systematically measuring the layer-wise effects of shortcuts on feature representations. In Section 3, we propose an experiment design that, given a fixed shortcut-inducing data skew, enables measuring *layer-wise resposibility* towards the loss in validation accuracy due to the shortcut. Our method is based on *counterfactual reasoning* about the behaviour of each layer had it been trained on unskewed data.

Next, we apply our method to evaluate layer-wise contributions to accuracy degradation due to *patch-like skews* on the CIFAR-10 and CIFAR-100 datasets (Krizhevsky, 2009) and study possible mechanisms behind this drop. Our experimental findings in Section 4 reveal that different types of patch-like skews affect network layers differently. Specifically, class-universal skews (affecting all

instances of a target class) and class-specific skews (affecting only one class) impact deeper layers more than non-universal and non-specific skews, respectively. This finding suggests that various aspects of data quality might have layer-specific effects. In Section 5, we identify the forgetting of shortcut-free features as a key mechanism behind accuracy degradation in our experiments, indicating a potential role of complexity-constraining mechanisms, such as simplicity bias (Shah et al., 2020) or regularization (Sagawa et al., 2020), in shortcut learning.

Finally, in Appendix A we evaluate the layer-wise effect of fine-tuning as a shortcut mitigation strategy and find that fine-tuning is somewhat less effective for shallow layers, supporting the application of layer-wise methods for adaptation to distribution shifts (e.g., Lee et al., 2023).

2 Related work

Impact of shortcuts on feature representations Several studies examine the impact of image shortcuts on learned representations (Hermann & Lampinen, 2020; Islam et al., 2021; Scimeca et al., 2022). They analyze how shortcuts are encoded in layers using metrics such as linear probing accuracy (Hermann & Lampinen, 2020), mutual information and read-out module accuracy (Islam et al., 2021), or validation accuracy on feature-labeled datasets (Scimeca et al., 2022). While these approaches provide insight into shortcut formation, they do not explicitly attribute accuracy degradation due to shortcuts to specific layers. In contrast, our work *quantifies layer-wise responsibility*, offering a more direct assessment of layer's role in shortcut learning.

Mechanisms of shortcut formation Our work contributes to the understanding of shortcut learning mechanisms (Shah et al., 2020; Sagawa et al., 2020; Nagarajan et al., 2021; Chaudhuri et al., 2023; Tsoy & Konstantinov, 2024). Our analysis of accuracy degradation suggests that feature forgetting plays a key role in shortcut learning, supporting prior hypotheses that simplicity bias (e.g., Shah et al., 2020; Tsoy & Konstantinov, 2024) or excessive regularization (Sagawa et al., 2020) might be an important factor in shortcut learning. In contrast to these works, we measure shortcut responsibilities for different layers, allowing for a more fine-grained quantitive understanding of shortcuts.

Layer-wise feature analysis Similarly to our work, Zhang et al. (2022); Maini et al. (2023); Huh et al. (2023) investigate feature representations in deep models and assess the importance of each layer for classification. Zhang et al. (2022) measure the importance of each particular layer of a deep network by injecting noise in the network weights. Maini et al. (2023) analyze the memorization behavior of different layers by introducing label noise. Huh et al. (2023) analyze learned feature representations of deep models and show how some of their properties help generalization. While these studies provide valuable insights into feature learning mechanisms, they do not attempt to quantify the responsibility of different layers for shortcut learning. Thus, these approaches are not directly comparable to ours.

Layer-wise fine-tuning analysis Our work also relates to the literature on layer-wise adaptation of deep models to distribution shifts (e.g., Kumar et al., 2022; Lee et al., 2023; Trivedi et al., 2023; Kirichenko et al., 2023). While we do not propose new adaptation methods, our findings can help practitioners identify potential bottlenecks and better understand the applicability of existing approaches.

3 Experiment design

This section outlines our experimental design for attributing *layer-wise responsibility in shortcut learning*. Our approach introduces *shortcut-inducing skews* into the training process in a *counterfactual* manner, allowing us to assess each layer's role directly. Specifically, we train multiple networks on the same task, exposing different layers of different networks to skewed data. Then, we evaluate these networks on a skew-free validation dataset and use these accuracies to quantify each layer's contribution to shortcut learning.

3.1 RESPONSIBILITY ATTRIBUTION

Formally, consider a feed-forward architecture

$$f(\theta, \cdot) = f^m(\theta^m, f^{m-1}(\theta^{m-1}, \dots, f^0(\theta^0, \cdot) \dots)).$$

Let $\theta^{i:j}$ represent the weights of layers i to j, and let $\mathrm{err}(\theta)$ denote the error rate of this architecture with weights θ on the shortcut-free validation dataset. Suppose that two networks of this architecture are trained for T rounds on clean data and data affected by a *shortcut-inducing skew* g respectively, resulting in final weights $\theta_{T,0}$ (trained on clean data) and $\theta_{T,m+1}$ (trained on skewed data). We measure the accuracy drop on clean validation data between these networks, interpreting it as a measure of shortcut learning. Our goal is to attribute *the total responsibility* for this drop

$$\mathrm{res}_a^{0:m} \coloneqq \mathrm{err}(\theta_{T,m+1}) - \mathrm{err}(\theta_{T,0})$$

to individual layers.

To this end, we want to analyze the features extracted by shallow skewed layers $\theta_{T,m+1}^{0:i-1}$ by replacing the corresponding skewed head $\theta_{T,m+1}^{i:m}$ with a head $\theta_{T,i}$ of the same architecture, counterfactually trained on clean data (see Section 3.2 for the counterfactual training procedure). We define *the absolute responsibility* of layers $0 : i-1$ for shortcut learning as the difference in validation accuracy (averaged over training seeds) between the clean network $\theta_{T,0}$ and a hybrid model composed of the skewed shallow layers $\theta_{T,m+1}^{0:i-1}$ and the clean head $\theta_{T,i}$,

$$\mathrm{res}_a^{0:i-1} \coloneqq \mathrm{err}((\theta_{T,m+1}^{0:i-1}, \theta_{T,i})) - \mathrm{err}(\theta_{T,0}).$$

We then define *the absolute responsibility* of layer i as the difference in the responsibilities of layers $0 : i$ and layers $0 : i-1$

$$\mathrm{res}_a^{i} \coloneqq \mathrm{res}_a^{0:i} - \mathrm{res}_a^{0:i-1}.$$

To compare results across experiments, we normalize this metric by the total responsibility, producing *the relative responsibility* of layer i

$$\mathrm{res}_r^{i} \coloneqq \mathrm{res}_a^{i} \,/\, \mathrm{res}_a^{0:m}.$$

3.2 COUNTERFACTUAL TRAINING ALGORITHM

A key challenge in our approach is designing a procedure for the *counterfactual* training of the heads. We solve this challenge using the simultaneous training procedure described in Algorithm 1. Here the loss of a network with weights θ on a data batch B is denoted by $L(\theta, B)$. This procedure trains the skewed network θ_{m+1}, the clean network θ_0, and the heads $\theta_1, \dots, \theta_m$ over T rounds. In each round, we sample a clean data batch B_t to update the heads and the clean network, where the head θ_i utilizes the shallow layers of the *skewed* network $\theta_{m+1}^{0:i-1}$ as a feature extractor. We then apply the *shortcut-inducing skew* g to generate a skewed batch $B'_t = g(B_t)$ and use it to update the skewed network.

Algorithm 1 Simultaneous training of networks

Initialize: $\theta_{0,m+1}$ — initial network weights.
$\forall s \in [m]\ \theta_{0,s} = \theta_{0,m+1}^{s:m}$ — initial heads weights.
for $t = 1$ **to** T **do**
  Sample batch B_t and skewed batch $B'_t = g(B_t)$
  for $s = 0$ **to** m **do**
    Update a head using skewed features:
    $\theta_{t,s} = \theta_{t-1,s} - \eta_t \nabla_{\theta^{s:m}} L((\theta_{t-1,m+1}^{0:s-1}, \theta_{t-1,s}), B_t)$
  end for
  Update skewed network:
  $\theta_{t,m+1} = \theta_{t-1,m+1} - \eta_t \nabla L(\theta_{t-1,m+1}, B'_t)$
end for

Rationale Throughout this process, each head progressively adapts to the intermediate skewed features to classify clean data. By design, all heads receive the same exposure to training data, with the only difference being the presence or the absence of a skew, enabling a meaningful comparison between them. Additionally, the training methodology remains consistent across all sub-networks, controlling for the optimizer's implicit biases.

To further justify our approach, we compare it with a simple post-training adaptation method, where the skewed network θ is first fully trained, and then clean heads are trained on top of the final skewed features. While both methods assess *the suitability of intermediate features for clean classification*, we argue that our procedure is better suited for *counterfactual comparison* for two key reasons. First, in our method, skewed and clean heads *progressively adapt to intermediate features*, an aspect shown to influence feature learning (Allen-Zhu & Li, 2019; Panigrahi et al., 2024; Abbe et al., 2022) compared to static post-training of heads. Second, our empirical obsevrations showed that our approach allows to use the *same hyperparameters for training all heads and networks to high accuracy*, whereas the post-training approach requires different hyperparameter tuning for each head to achieve high final accuracy. Since hyperparameters such as learning rate significantly impact feature quality (e.g., Li et al., 2019; Lewkowycz et al., 2020), our approach provides a more controlled basis for *attributing responsibility to specific layers*.

4 Responsibility estimation on vision data

We conducted experiments on CIFAR-10 and CIFAR-100 datasets (Krizhevsky, 2009) using five architectures: ResNet-10, ResNet-18, ResNet-34, ResNet-50 (He et al., 2016), and VGG-11 (Simonyan & Zisserman, 2015), all trained with stochastic gradient descent (SGD). For a layer-wise analysis, we divided each network into five layers. The first four layers roughly correspond to standard feature blocks in ResNet architectures (which are constructed according to the output size), while the final layer consists solely of the linear classification layer (see details in Appendix B.1).

Our experiments consider a patch-like skew that blends the upper-left corner of selected training images with a class-dependent solid color (see Figure 1). This skew simplifies the classification task by providing an easily learnable color feature, encouraging the network to rely on shortcut-based classification rather than original CIFAR features.

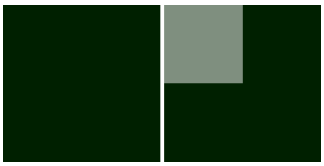

Figure 1: Example of the patch-like skew

The introduced skew has four hyperparameters: (1) skew frequency within a class, (2) blending strength, (3) affected region size, and (4) affected classes. We systematically varied these parameters across all architecture-dataset combinations. First, we define three skew categories: *One*, *Ten*, and *Combo3*. On CIFAR-10, the *One* category applies 1 color to 1 class, the *Ten* category assigns a different color to each of the 10 classes, and the *Combo3* category divides all classes into 3 groups of sizes 3, 3, and 4 and blends each group with a different color. For CIFAR-100, to ensure comparability, the *One* category skews 10 classes with one color, the *Ten* category skews all 100 classes with 10 colors, and the *Combo3* category uses 3 colors for 100 classes.

Within each category, we define three skew types: *Rare*, *Weak*, and *Small*. *Rare* skews affect the entire image and have a strong blending strength but only impact a fraction of the target class, leaving some clean data for training. *Weak* skews affect the entire image and all instances of a target class but with weaker blending strength. *Small* skews affect the whole target class and have strong blending but only affect a portion of the image.

For each type, we vary the presence of skew along frequency, blending strength, and size axis for *Rare*, *Weak*, and *Small* types, respectively, resulting in 27 experiments per architecture-dataset pair. We repeat each experiment 4 times to average out training noise.

Table 1: Average relative responsibility of individual layers on CIFAR-10 dataset

Skew class	Layer 0	Layer 1	Layer 2	Layer 3	Layer 4
All	4.0% (0.8%)	11.1% (1.2%)	16.9% (1.4%)	31.7% (1.7%)	33.4% (2.4%)
Low	2.4% (0.8%)	12.1% (1.6%)	16.5% (1.4%)	35.4% (1.8%)	33.5% (3.5%)
Medium	4.7% (1.3%)	11.1% (2.2%)	18.7% (2.0%)	32.1% (2.1%)	33.4% (3.7%)
High	4.8% (1.9%)	10.2% (2.6%)	15.4% (3.4%)	27.6% (4.1%)	33.2% (5.2%)
Ten	4.8% (1.3%)	17.3% (2.2%)	18.6% (2.4%)	29.9% (2.7%)	29.4% (4.1%)
Combo3	4.7% (1.7%)	14.1% (2.4%)	18.3% (2.6%)	32.7% (3.4%)	21.2% (4.5%)
One	2.4% (1.1%)	2.0% (0.5%)	13.7% (2.2%)	32.5% (2.5%)	49.4% (2.4%)
Rare	9.7% (2.0%)	16.5% (2.9%)	26.4% (2.6%)	27.5% (2.4%)	15.4% (4.0%)
Weak	1.1% (0.6%)	10.0% (1.7%)	17.5% (1.9%)	41.7% (2.6%)	29.7% (2.7%)
Small	1.1% (0.5%)	6.9% (1.3%)	6.6% (1.7%)	25.9% (3.1%)	55.0% (3.4%)

Average relative responsibility of individual layers (and the between experiments standard deviation of the average in parenthesis) over a certain skew class. Insignificant absolute average responsibilities are set to zero to reduce the influence of outliers.

Training results We depict the absolute responsibility of the sequences of layers for CIFAR-10–ResNet-50 pair in Appendix B.2. As expected, responsibility generally increases with the number of skewed layers. Furthermore, different layers contribute differently to validation accuracy degradation, and this effect varies with the skew parameters.

To systematically assess the role of layers in shortcut learning, we compute the average relative responsibility of each layer across experiments. Since responsibility values are transformed to the same scale, observations with small absolute responsibilities can introduce noise and outliers. To mitigate this, we set the average responsibility of a given layer sequence $0:i$ to zero if it is not significantly different from zero at a 5% significance level (using the t-statistic).

Table 1 presents the results for the CIFAR-10 dataset. On average, for our set of experiments, the last and penultimate layers are mostly responsible for shortcuts. *Rare* skews tend to affect shallow layers more than *Weak* or *Small* skews. Similarly, the *Combo3* and *Ten* categories of skews affect shallow layers more than the *One* category. At the same time, the *Small* type mostly affects the last layer. (See Appendix B.3 for the results for CIFAR-100 dataset.)

Discussion Our findings indicate that shortcut learning is primarily driven by the last two layers. The strong impact of the final layer (especially for *Small* skews) may explain why adaptation methods that modify only the last classification layer are often effective (e.g., Kirichenko et al., 2023). However, the broader involvement of earlier layers (particularly for *Rare* skews) explains why schemes that involve full-network fine-tuning often outperform simple linear probing in real-world situations (e.g., Kumar et al., 2022). More generally, our results highlight that different aspects of data quality might have different layer-specific effects. Additionally, our results for the *Rare* type suggest that dataset homogeneity may enhance feature learning, though further investigation is necessary to validate this observation.

5 Analysis of shortcut learning mechanisms

To analyze the mechanisms behind shortcut formation, we compare the last-layer features of different heads, evaluated on the validation dataset, against those of a clean network. We introduce three key metrics to quantify changes in feature representation due to shortcuts: *Skewness*, *Forgetting*, and *Inconsistency*. Using regression analysis, we assess how well these metrics explain shortcut learning in different layers, allowing us to pinpoint the primary mechanisms behind shortcut formation.

Metrics To define these metrics, assume that an experiment with seed $s \in S$ resulted in the models $\theta_{T,i,s}$ and consider a clean validation dataset D. For each head, we extract the last-layer features of corresponding to layers $0:i-1$ on the clean dataset

$$F_s^i \coloneqq \phi((\theta_{T,m+1,s}^{0:i-1}, \theta_{T,i,s}), D),$$

Table 2: Regression of absolute responsibility on the explanatory metrics

	CIFAR-10 absolute responsibility				CIFAR-100 absolute responsibility			
Forget.	0.639^*	0.6065^*			0.534^*	0.6520^*		
	(0.026)	(0.0122)			(0.036)	(0.0184)		
Skew.	-0.045		0.727^*		0.210^*		0.882^*	
	(0.034)		(0.027)		(0.057)		(0.046)	
Incons.	-0.175^*			-0.17	-0.026			-0.02
(corr.)	(0.035)			(0.23)	(0.072)			(0.25)
R^2	0.946	0.939	0.812	0.009	0.798	0.787	0.641	0.001
N	540	540	540	540	540	540	540	540

Each column presents coefficients (and their standard errors in parenthesis) of a specific regression. Coefficients with * are significantly different from zero on a 5% significance level.

where $\phi(\theta, D)$ are the last-layer features of model θ on dataset D.

Forgetting metric measures how many clean features are lost in the skewed feature extractor relative to the clean network. To compute it, we calculate the average R^2 statistic (Draper, 1998) of the regularized regressions of the clean features F_s^0 onto the skewed features F_s^i and compare it with the same statistic for the regression of clean features onto the clean features. Formally, we define $R_{n,k}^{i,j}$ as the R^2 statistic of the regression of F_k^j on F_n^j and *Forgetting* metric for layers $0:i-1$ as

$$\textit{Forgetting}^{0:i-1} \coloneqq \frac{1}{(|S|-1)|S|} \sum_{k,s\neq k} R_{k,s}^{0,0} - R_{k,s}^{i,0}.$$

Skewness metric measures how many shortcut-related features emerged in the skewed feature extractors compared to the clean network. To compute it, we calculate the average R^2 statistic of the regularized regressions of the skewed features onto the clean features and compare it with the same statistic for the regressions of the skewed features on the skewed features. Formally, we define *Skewness* metric for layers $0:i-1$ as

$$\textit{Skewness}^{0:i-1} \coloneqq \frac{1}{(|S|-1)|S|} \sum_{k,s\neq k} R_{s,k}^{i,i} - R_{s,k}^{0,i}.$$

Inconsistency metric measures the overall dissimilarity between the features of the skewed feature extractor and the clean network. To compute it, we calculate the average RV coefficient (Robert & Escoufier, 1976; Kornblith et al., 2019) between skewed and clean features and compare it with the same statistic for the pairs of clean features. Formally, define $C_{n,k}^{i,j}$ as the RV coefficient of features F_n^i and F_k^j. Then, we define *Inconsistency* metric for layers $0:i-1$ as

$$\textit{Inconsistency}^{0:i-1} \coloneqq \frac{1}{(|S|-1)|S|} \sum_{k,s\neq k} C_{k,s}^{0,0} - C_{k,s}^{i,0}.$$

Regression After calculating the desired metrics, we regress the absolute responsibility of layers $0:i$ on these three explanatory variables. Since *Inconsistency* metric also partially accounts the effects of the *Forgetting* and *Skewness* metrics, we correct it by regressing *Inconsistency* metric on *Forgetting* and *Skewness* metrics first and then use the residuals of this *Inconsistency* metric for the final regression.

Table 2 outlines our results. As we can see, only *Forgetting* and *Skewness* metrics have considerable explanatory power, suggesting that either feature forgetting or spurious feature learning is important in shortcut learning. Since *Skewness* and *Forgetting* metrics are highly correlated, their explanatory power is hard to compare. However, when *Skewness* metric remains the only explanatory variable in the regressions for the CIFAR-10 dataset, the regression coefficient changes sign, suggesting that the *Skewness* metric only tries to play a role of *Forgetting* metric and does not explain the results by itself. Thus, the forgetting of shortcut-free features appears to be a key factor in shortcut learning,

suggesting that complexity-constraining mechanisms, such as simplicity bias (Shah et al., 2020; Tsoy & Konstantinov, 2024) or excessive regularization (Sagawa et al., 2020), may play a significant role in this process. Additionally, we conduct a series of layer-specific regressions of the same type in Appendix C.1 and find that the results remain generally consistent across layers.

6 Conclusion

This work proposes a new counterfactual-based method for layer-wise analysis of deep features. This method allows researchers to investigate the layer-wise localization of shortcut features for a general shortcut-inducing skew on a shortcut-free dataset. Using this methodology, we investigated the emergence of shortcut features in CNNs for the specific type of patch-like skew. Our findings in this setting suggest that shortcut learning is generally not localized in a specific layer but occurs throughout the network. Moreover, even for the same type of shortcut, the responsibility of layers for shortcut learning might vary significantly. We hope these insights will inform the development of more robust models and improve fine-tuning strategies for mitigating shortcut dependencies.

Acknowledgments

This research was partially funded from the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure). This project was supported with computational resources provided by Google Cloud Platform (GCP).

References

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for SGD learning of sparse functions on two-layer neural networks. In Po-Ling Loh and Maxim Raginsky (eds.), *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 4782–4887. PMLR, 02–05 Jul 2022.

Zeyuan Allen-Zhu and Yuanzhi Li. What Can ResNet Learn Efficiently, Going Beyond Kernels? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization, 2020. arXiv:1907.02893 [stat.ML].

Kamalika Chaudhuri, Kartik Ahuja, Martin Arjovsky, and David Lopez-Paz. Why does Throwing Away Data Improve Worst-Group Error? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4144–4188. PMLR, 23–29 Jul 2023.

NR Draper. *Applied regression analysis*. McGraw-Hill. Inc, 1998.

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(11):665–673, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Katherine Hermann and Andrew Lampinen. What shapes feature representations? exploring datasets, architectures, and training. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9995–10006. Curran Associates, Inc., 2020.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The Low-Rank Simplicity Bias in Deep Networks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Md Amirul Islam, Matthew Kowal, Patrick Esser, Sen Jia, Björn Ommer, Konstantinos G. Derpanis, and Neil Bruce. Shape or Texture: Understanding Discriminative Features in CNNs. In *International Conference on Learning Representations*, 2021.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations. In *The Eleventh International Conference on Learning Representations*, 2023.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images, 2009.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *International Conference on Learning Representations*, 2022.

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical Fine-Tuning Improves Adaptation to Distribution Shifts. In *The Eleventh International Conference on Learning Representations*, 2023.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism, 2020. arXiv:2003.02218 [stat.ML].

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards Explaining the Regularization Effect of Initial Large Learning Rate in Training Neural Networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Pratyush Maini, Michael Curtis Mozer, Hanie Sedghi, Zachary Chase Lipton, J Zico Kolter, and Chiyuan Zhang. Can Neural Network Memorization Be Localized? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23536–23557. PMLR, 23–29 Jul 2023.

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*, 2021.

Abhishek Panigrahi, Bingbin Liu, Sadhika Malladi, Andrej Risteski, and Surbhi Goel. Progressive distillation improves feature learning via implicit curriculum. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024.

P. Robert and Y. Escoufier. A Unifying Tool for Linear Multivariate Statistical Methods: The RV-Coefficient. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 25(3):257–265, 12 1976. ISSN 0035-9254. doi: 10.2307/2347233.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An Investigation of Why Overparameterization Exacerbates Spurious Correlations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8346–8356. PMLR, 13–18 Jul 2020.

Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoo Yun. Which Shortcut Cues Will DNNs Choose? A Study from the Parameter-Space Perspective. In *International Conference on Learning Representations*, 2022.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The Pitfalls of Simplicity Bias in Neural Networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9573–9585. Curran Associates, Inc., 2020.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, 2014. arXiv:1312.6034 [cs.CV].

Puja Trivedi, Danai Koutra, and Jayaraman J. Thiagarajan. A Closer Look at Model Adaptation using Feature Distortion and Simplicity Bias. In *The Eleventh International Conference on Learning Representations*, 2023.

Nikita Tsoy and Nikola Konstantinov. Simplicity Bias of Two-Layer Networks beyond Linearly Separable Data. In *Forty-first International Conference on Machine Learning*, 2024.

Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are All Layers Created Equal? *Journal of Machine Learning Research*, 23(67):1–28, 2022. URL `http://jmlr.org/papers/v23/20-069.html`.

Appendices

A Fine-tuning adaptations **10**

B Additional results for Section 4 **12**

B.1 Details of training 12

B.2 Additional figures 13

B.3 Additional tables 13

C Additional results for Section 5 **14**

C.1 Layer-wise regression analysis 15

A FINE-TUNING ADAPTATIONS

Finally, we investigate the layer-wise effects of fine-tuning as a shortcut-mitigation strategy. To this end, we fine-tune a feature extractor with a corresponding clean head on an (around 6.7 times) smaller held-out clean dataset. Note that, in this experiment, we do not reinitialize the linear classification layer; hence, our setup becomes similar to the LP-FT setup of Kumar et al. (2022).

We use the standard SGD optimizer from PyTorch and linear learning scheduler with warm-up. The parameters of data and optimizer are listed below.

batch_size	512
lr	2^{-9}
momentum	0.9
nesterov	True
weight_decay	0.0005
Share of warm-up steps	12.5%
Number of epochs	32

Figure 2 compares the absolute responsibility of layers before and after fine-tuning for the ResNet-50–CIFAR-10 pair on the Ten category. As we can see, fine-tuning does not completely recover the network from shortcuts. Moreover, fine-tuning for this experiment seems somewhat relatively more effective for the deeper layers and less effective for shallow ones.

To systematically assess the effect of fine-tuning, we investigate the average relative improvement in absolute responsibility due to fine-tuning. Since the relative improvements are the ratios of two responsibilities over each other, the noise of the denominator could lead to big outliers. Thus, we transform relative improvements for each experiment in the following manner. If the average absolute responsibility before fine-tuning is not greater than zero on a 5% level according to the t-test or the difference in the responsibilities before and after fine-tuning is insignificant from each other, we set the relative improvement to 0%. If the absolute responsibility after fine-tuning is not greater than zero on a 5% level, we set the improvement to 100%. If both rules apply, we exclude the observation from the average (If we are left with only one observation in the category, we only report its average without its standard deviation).

Table 3 presents the results for the CIFAR-10 dataset. Fine-tuning does not seem to recover the network from shortcuts completely. Additionally, we notice that fine-tuning is more effective for the penultimate layer compared to shallower layers. Specifically, fine-tuning somewhat improves the distortion in the first layer only for the *Rare* type.

Table 4 presents the results for the CIFAR-100 dataset. Similarly to the previous case, fine-tuning does not seem to recover the network from shortcuts completely. However, it becomes somewhat more efficient for shallow layers and somewhat less efficient for deep layers (but the improvement

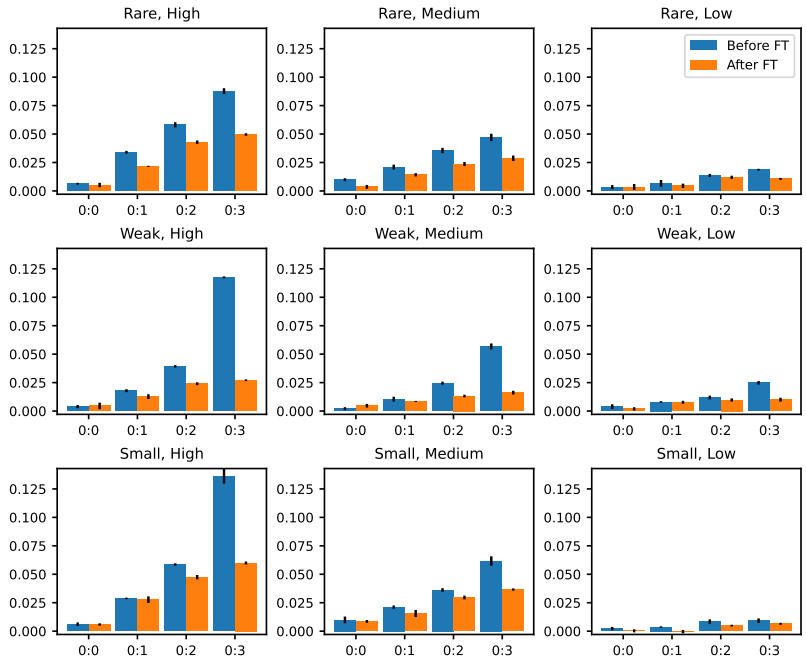

Figure 2: Absolute responsibility of the sequence of layers before (blue) and after after (orange) fine-tuning for ResNet-50–CIFAR-10 pair on the shortcuts from *Ten* category. Rows correspond to *Rare*, *Weak*, and *Small* shortcut types respectively. Columns correspond to *High*, *Medium*, and *Low* presence of the shortcut. Error bars depict the standard deviation of average responsibility over 4 training runs.

Table 3: Average relative improvement in absolute responsibility for CIFAR-10 dataset

Skew class or Model	Layers $0:0$	Layers $0:1$	Layers $0:2$	Layers $0:3$
All	7.4% (4.4%)	9.2% (2.2%)	13.5% (2.2%)	52.5% (2.1%)
High presence	9.1% (6.7%)	12.7% (3.4%)	11.4% (2.4%)	56.3% (1.6%)
Medium presence	7.7% (7.7%)	2.5% (1.8%)	11.3% (2.7%)	55.3% (2.3%)
Low presence	0.0% (0.0%)	13.5% (7.0%)	18.8% (6.1%)	44.7% (6.2%)
Ten category	0.0% (0.0%)	8.8% (3.3%)	20.9% (3.8%)	50.4% (3.4%)
Combo3 category	4.5% (4.5%)	14.1% (4.5%)	11.3% (3.5%)	44.3% (3.4%)
One category	40.0% (24.5%)	1.1% (1.1%)	7.6% (3.7%)	62.9% (3.6%)
Rare type	18.9% (10.6%)	9.1% (2.7%)	11.2% (3.2%)	45.0% (3.1%)
Weak type	0.0% (0.0%)	16.9% (5.6%)	21.1% (4.5%)	67.9% (2.1%)
Small type	0.0% (0.0%)	0.7% (0.5%)	7.3% (2.9%)	43.8% (4.3%)
ResNet-10	0.0% (0.0%)	2.5% (1.7%)	4.3% (2.0%)	39.8% (4.1%)
ResNet-18	0.0% (0.0%)	19.6% (8.3%)	22.6% (6.0%)	56.5% (4.9%)
ResNet-34	0.0% (0.0%)	8.7% (3.5%)	19.1% (5.5%)	58.5% (4.2%)
ResNet-50	16.1% (11.6%)	14.1% (5.8%)	13.3% (4.5%)	52.4% (4.0%)
VGG-11	50.0% (50.0%)	0.0% (0.0%)	8.7% (5.1%)	56.9% (5.5%)

Average relative improvement in responsibility for individual layers (and the between experiments standard deviation of the average in parenthesis) over a certain shortcut class. Some improvements are clipped to 0% or 100% to avoid outliers.

varies a lot between experiments for shallow layers). Additionally, we could see that some averages in the table are negative. These anomalies probably emerged due to noise in both the training and fine-tuning processes.

Table 4: Average relative improvement in absolute responsibility for CIFAR-100 dataset

Skew class or Model	Layers $0:0$	Layers $0:1$	Layers $0:2$	Layers $0:3$
All	22.2% (10.1%)	29.6% (6.3%)	10.1% (3.1%)	27.9% (2.8%)
High presence	40.0% (24.5%)	27.5% (9.1%)	5.6% (4.4%)	33.1% (4.2%)
Medium presence	12.5% (12.5%)	35.0% (10.9%)	6.7% (3.2%)	24.8% (4.1%)
Low presence	20.0% (20.0%)	20.5% (14.6%)	24.6% (10.1%)	25.5% (6.8%)
Ten category	14.3% (14.3%)	19.1% (7.2%)	7.4% (4.5%)	19.9% (3.6%)
Combo3 category	42.9% (20.2%)	43.7% (12.0%)	9.6% (4.3%)	19.4% (3.8%)
One category	0.0% (0.0%)	47.9% (28.9%)	20.3% (11.3%)	47.7% (6.3%)
Rare type	22.2% (14.7%)	43.3% (10.5%)	16.0% (5.4%)	27.3% (4.2%)
Weak type	25.0% (25.0%)	41.7% (14.9%)	8.6% (4.4%)	33.2% (4.9%)
Small type	20.0% (20.0%)	5.9% (5.9%)	4.7% (6.3%)	23.3% (5.6%)
ResNet-10	0.0% (0.0%)	0.0% (0.0%)	−4.9% (4.9%)	5.4% (2.6%)
ResNet-18	0.0% (−%)	0.0% (0.0%)	0.0% (0.0%)	12.1% (4.5%)
ResNet-34	0.0% (0.0%)	5.4% (5.4%)	−2.9% (2.9%)	17.9% (5.8%)
ResNet-50	50.0% (18.9%)	47.8% (14.6%)	6.4% (3.6%)	42.0% (6.5%)
VGG-11	0.0% (−%)	67.2% (12.8%)	39.2% (8.9%)	55.5% (4.3%)

Average relative improvement in responsibility for individual layers (and the between experiments standard deviation of the average in parenthesis) over a certain shortcut class. Some improvements are clipped to 0% or 100% to avoid outliers.

Discussion Our experiments suggest that the improvement in the absolute responsibility due to fine-tuning is generally disproportional across layers. It means that the "suitability" of the layer for classification is generally different from the "adaptability" of the layer under the fine-tuning procedure. This finding motivates the usefulness of approaches that adopt layers differently to the new target distribution (e.g., Lee et al., 2023).

B Additional results for Section 4

Here, we present additional details and results for the experiments in Section 4.

B.1 Details of training

Model details We define ResNet-10, as the network that consists of 1 convolutional layer and 4 BasicBlocks. For VGG-11, we use the version of architecture with batch normalization layers. For all architectures, we replace the initial 7×7 convolutional layer with 3×3 layer since this size is better suited for small CIFAR images.

As discussed in the main text, we divide all models to five layers. For ResNets, we base our layers on the usual layer blocks for ResNet models, but we assign the first convolutional layer to the first block. For VGG-11, we divide the networks by the max-pool layers (except for the last max-pool layer), it results in precisely four blocks of layers.

Training procedure We use the standard SGD optimizer from PyTorch and linear learning scheduler with warm-up. The parameters of data and optimizer are listed below.

batch_size	512
lr	0.25
momentum	0.9
nesterov	True
weight_decay	0.0005
Share of warm-up steps	12.5%
Number of epochs	256

Skew hyperparameters We use the following hyperparameters for the skew.

Type	Category	Frequency	Blending Strength	Size
Rare	*Ten* and *Combo3*	$1/2, 3/4, 7/8$	$1/8$	32
	One	$7/8, 31/32, 127/128$	$1/2$	32
Weak	*Ten* and *Combo3*	1	$1/48, 1/32, 1/24$	32
	One	1	$1/12, 1/8, 1/2$	32
Small	*Ten* and *Combo3*	1	$1/16$	$4, 5, 7$
	One	1	$1/3$	$4, 5, 8$

B.2 Additional figures

Figures 3, 4, and 5 compare the absolute responsibility of the sequence of layers for the CIFAR-10–ResNet-50 pair on *Ten*, *One*, and *Comobo3* categories.

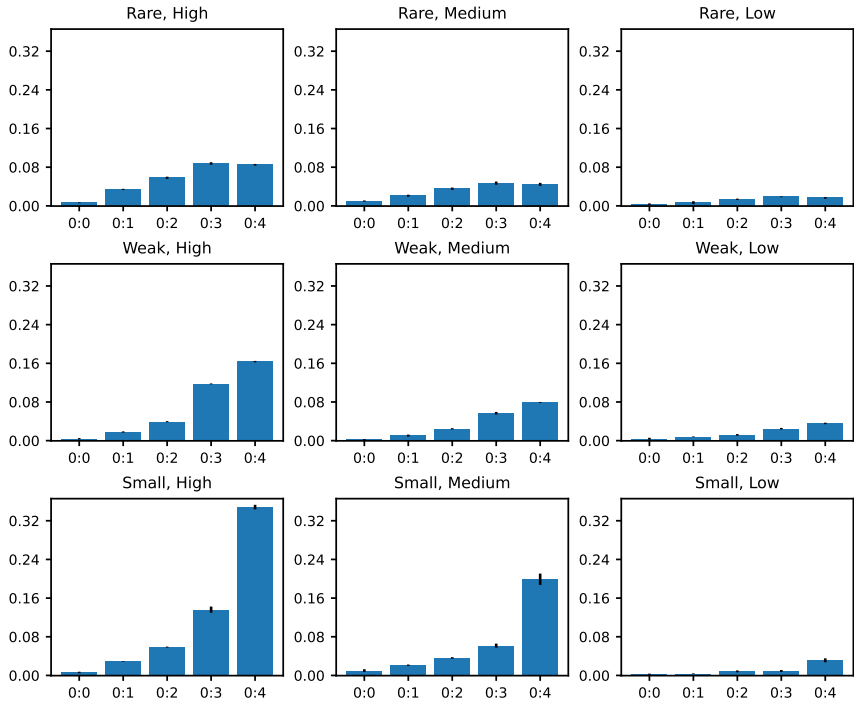

Figure 3: Absolute responsibility of the sequence of layers for ResNet-50–CIFAR-10 pair on the skews from *Ten* category. Rows correspond to *Rare*, *Weak*, and *Small* skew types respectively. Columns correspond to *High*, *Medium*, and *Low* presence of the skew. Error bars depict the standard deviation of average responsibility over 4 training runs.

B.3 Additional tables

Table 5 presents the results for the CIFAR-100 dataset. Similar to the previous case, the last and penultimate layers are mostly responsible for shortcuts; the *Rare* type and the *Combo3* and *Ten* categories tend to affect shallower layers of a network, while the *Small* type predominantly affects the last layer. However, compared to the CIFAR-10 dataset, the shortcuts start to influence deeper layers. Also, we notice that the average relative responsibility of layers 0 and 1 sometimes becomes negative. These anomalies are probably the effect of noise since the absolute responsibility for layers 0 and 1 is usually quite small.

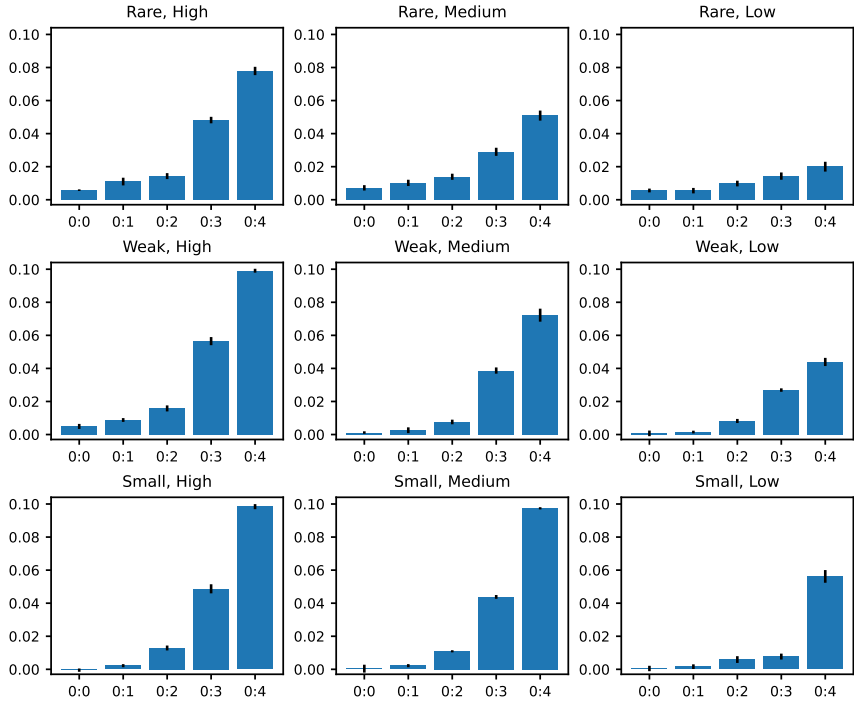

Figure 4: Absolute responsibility of the sequence of layers for ResNet-50–CIFAR-10 pair on the skews from *One* category. Rows correspond to *Rare*, *Weak*, and *Small* skew types respectively. Columns correspond to *High*, *Medium*, and *Low* presence of the skew. Error bars depict the standard deviation of average responsibility over 4 training runs.

Table 5: Average relative responsibility of individual layers on CIFAR-100 dataset

Skew class or Model	Layer 0	Layer 1	Layer 2	Layer 3	Layer 4
All	1.5% (0.5%)	3.2% (0.8%)	10.2% (1.2%)	31.5% (2.0%)	51.4% (2.4%)
Low	2.3% (0.8%)	3.9% (1.3%)	8.5% (1.4%)	36.0% (2.8%)	49.3% (3.0%)
Medium	1.5% (0.7%)	3.6% (1.3%)	11.7% (2.2%)	35.4% (3.3%)	47.8% (3.1%)
High	0.7% (0.9%)	2.1% (1.4%)	10.3% (2.8%)	23.2% (4.1%)	57.0% (5.6%)
Ten	1.0% (0.7%)	6.4% (1.2%)	12.3% (1.9%)	23.8% (2.5%)	54.3% (3.8%)
Combo3	2.2% (0.9%)	3.3% (1.5%)	15.7% (2.6%)	22.2% (2.5%)	52.1% (4.2%)
One	1.4% (0.7%)	−0.1% (1.2%)	2.5% (1.4%)	48.6% (3.9%)	47.6% (4.3%)
Rare	3.9% (1.1%)	6.3% (2.0%)	19.5% (2.8%)	36.5% (3.7%)	33.7% (3.9%)
Weak	0.2% (0.7%)	1.6% (0.8%)	5.2% (1.4%)	37.6% (3.1%)	55.4% (3.1%)
Small	0.4% (0.3%)	1.7% (0.6%)	5.8% (1.2%)	20.5% (3.2%)	64.9% (3.9%)
ResNet-10	1.0% (0.6%)	2.5% (1.1%)	12.2% (3.5%)	20.2% (3.4%)	60.4% (4.7%)
ResNet-18	−0.7% (0.7%)	2.5% (1.2%)	8.3% (2.6%)	31.5% (4.1%)	58.4% (4.3%)
ResNet-34	0.5% (0.7%)	3.4% (1.2%)	6.5% (2.3%)	25.6% (4.1%)	63.9% (4.8%)
ResNet-50	5.6% (1.6%)	0.4% (2.0%)	12.0% (2.5%)	29.9% (3.8%)	52.0% (4.5%)
VGG-11	1.1% (0.8%)	7.3% (2.6%)	11.8% (2.9%)	50.4% (5.1%)	22.0% (4.1%)

Average relative responsibility of individual layers (and the between experiments standard deviation of the average in parenthesis) over a certain skew class. Insignificant absolute average responsibilities are set to zero to reduce the influence of outliers.

C Additional results for Section 5

Here, we present additional results for Section 5.

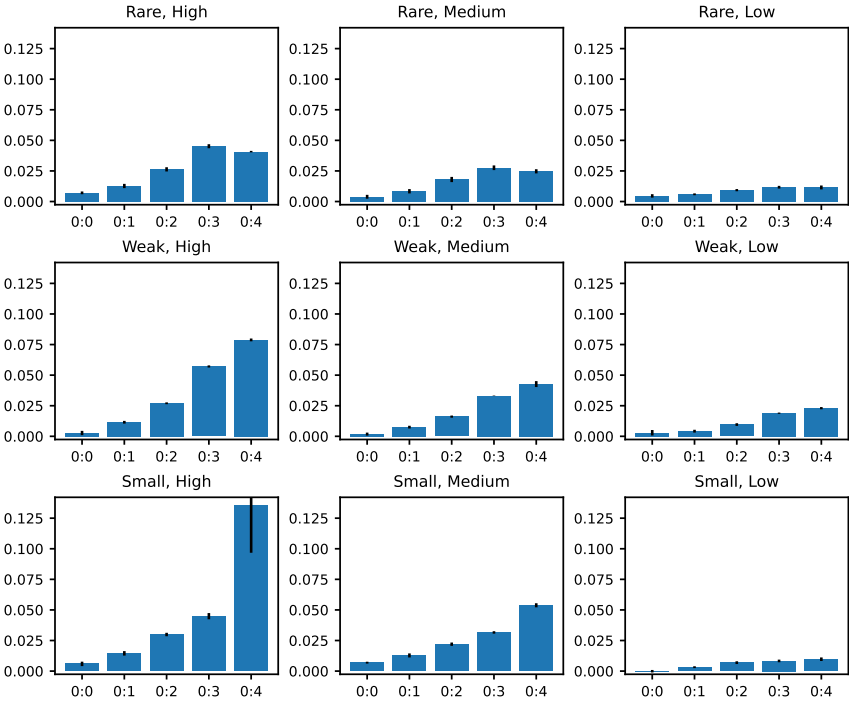

Figure 5: Absolute responsibility of the sequence of layers for ResNet-50–CIFAR-10 pair on the skews from *Combo3* category. Rows correspond to *Rare*, *Weak*, and *Small* skew types respectively. Columns correspond to *High*, *Medium*, and *Low* presence of the skew. Error bars depict the standard deviation of average responsibility over 4 training runs.

C.1 Layer-wise regression analysis

Tables 6–9 present the regression analysis results for individual network layers. Overall, these findings align with the main regression analysis in Section 5. However, the *Inconsistency* metric begins to show some explanatory power in the regressions for the CIFAR-100 dataset. Additionally, we observe a decline in the explanatory power of covariates for Layer 0. This drop is likely due to increased noise in the results for Layer 0, which inflates the denominator of the R^2 statistic.

Table 6: Regression of absolute responsibility on the explanatory metrics for Layer 0

	CIFAR-10 absolute responsibility				CIFAR-100 absolute responsibility			
Forget.	0.455*	0.528*			0.458*	0.586*		
	(0.085)	(0.073)			(0.139)	(0.119)		
Skew.	0.244		0.932*		0.274		0.623*	
	(0.177)		(0.159)		(0.154)		(0.133)	
Incons.	−0.11			−0.11	1.29*			1.29*
(corr.)	(0.25)			(0.35)	(0.23)			(0.27)
R^2	0.432	0.421	0.260	0.008	0.308	0.155	0.110	0.146
N	135	135	135	135	135	135	135	135

Table 7: Regression of absolute responsibility on the explanatory metrics for Layer 1

	CIFAR-10 absolute responsibility				CIFAR-100 absolute responsibility			
Forget.	0.269*	0.450*			0.548*	0.697*		
	(0.033)	(0.024)			(0.065)	(0.058)		
Skew.	0.487*		1.022*		0.447*		0.964*	
	(0.085)		(0.069)		(0.099)		(0.097)	
Incons.	−0.292			−0.29	0.720*			0.72*
(corr.)	(0.167)			(0.54)	(0.170)			(0.35)
R^2	0.850	0.801	0.772	0.010	0.595	0.497	0.338	0.056
N	135	135	135	135	135	135	135	135

Table 8: Regression of absolute responsibility on the explanatory metrics for Layer 2

	CIFAR-10 absolute responsibility				CIFAR-100 absolute responsibility			
Forget.	0.322*	0.5000*			0.447*	0.613*		
	(0.053)	(0.0188)			(0.039)	(0.053)		
Skew.	0.212*		0.564*		0.838*		1.139*	
	(0.063)		(0.024)		(0.052)		(0.082)	
Incons.	0.021			0.02	0.421*			0.42
(corr.)	(0.066)			(0.22)	(0.101)			(0.29)
R^2	0.906	0.895	0.873	0.007	0.818	0.538	0.547	0.034
N	135	135	135	135	135	135	135	135

Table 9: Regression of absolute responsibility on the explanatory metrics for Layer 3

	CIFAR-10 absolute responsibility				CIFAR-100 absolute responsibility			
Forget.	0.766*	0.6370*			0.528*	0.558*		
	(0.031)	(0.0154)			(0.050)	(0.029)		
Skew.	−0.165*		0.671*		0.052		0.618*	
	(0.038)		(0.043)		(0.076)		(0.062)	
Incons.	−0.118*			−0.11	−0.043			−0.04
(corr.)	(0.035)			(0.23)	(0.094)			(0.21)
R^2	0.960	0.948	0.751	0.012	0.699	0.697	0.465	0.007
N	135	135	135	135	135	135	135	135