Spurious Features Everywhere - Large-Scale Detection of Harmful Spurious Features in ImageNet

Yannic Neuhaus Maximilian Augustin Valentyn Boreiko Matthias Hein
Tübingen AI Center – University of Tübingen

Abstract

Benchmark performance of deep learning classifiers alone is not a reliable predictor for the performance of a deployed model. In particular, if the image classifier has picked up spurious features in the training data, its predictions can fail in unexpected ways. In this paper, we develop a framework that allows us to systematically identify spurious features in large datasets like ImageNet. It is based on our neural PCA components and their visualization. Previous work on spurious features often operates in toy settings or requires costly pixel-wise annotations. In contrast, we work with ImageNet and validate our results by showing that presence of the harmful spurious feature of a class alone is sufficient to trigger the prediction of that class. We introduce the novel dataset "Spurious ImageNet" which allows to measure the reliance of any ImageNet classifier on harmful spurious features. Moreover, we introduce SpuFix as a simple mitigation method to reduce the dependence of any ImageNet classifier on previously identified harmful spurious features without requiring additional labels or retraining of the model. We provide code and data at https://github.com/YanNeu/spurious_imagenet.

1. Introduction

Deep learning has led to tremendous progress in image classification [37, 50] and natural language processing [14]. Over the years, however, it has become apparent, that evaluating predictive performance on a fixed test set is not necessarily indicative of the performance when image classifiers are deployed in the wild. Several potential failure cases have been discovered. This starts with a lack of robustness due to image corruptions [31], adversarial perturbations [68], and arbitrary predictions on out-of-distribution inputs [45, 32, 30]. In this paper, we consider the problem of identifying and debugging image classifiers from spurious features [2]. Spurious features in image classification are features that co-occur with the actual class object and are picked up by the classifier. In the worst case, they lead to shortcut learning [25], where only the spurious but not the correct feature is associated with the class, e.g., [86] found that a pneumonia detector's bad generalization across hospitals was caused by the neural network learning to identify the hospital where the training data originated from. A weaker form of spurious feature (at least from a learning perspective) is the case when the classifier picks up the correct class features, e.g., of a hummingbird, but additionally associates a spurious feature, e.g., a bird feeder, with the class as they appear together on a subset of the training set. This becomes a harmful spurious feature if *only* the spurious feature *without* the class feature is sufficient to trigger the classification of that class, see Fig. 1 for an illustration of

Figure 1: **Top:** Examples of spurious features found via our neural PCA components but not in previous study [61]. **Bottom:** We validate our spurious features by mining images from the web showing **only the spurious feature but not the class**. They are classified by four ImageNet models as the corresponding class. Some of them even contain ImageNet classes (bees on feeder, grasshopper in leaves).

such spurious features found via our method. Harmful spurious features are difficult to detect and thus can easily go unnoticed, leading to unexpected behaviour of a deployed image classifier.

In this paper we make the following key contributions:

- we develop a pipeline for the detection of harmful spurious features with little human supervision based on our class-wise neural PCA (NPCA) components of an adversarially robust classifier together with their Neural PCA Feature Visualization (NPFV).
- unlike prior work, which used masking images or pixel-wise annotations, we validate our found spurious features by using our NPCA components to find real images containing only the spurious feature but not the class object.
- using these images we create the dataset "Spurious ImageNet" and propose a measure for dependence on spurious features. We do a large-scale evaluation of state-of-the-art (SOTA) ImageNet models. We show that the spurious features found for the robust model generalize to non-robust classifiers. Moreover, we analyze the influence of different training setups, e.g. pre-training on ImageNet21k or larger datasets like LAION.
- we develop SpuFix, a technique to mitigate the dependence on identified harmful spurious features without requiring new labels or retraining, and show how to transfer it to any ImageNet classifier. SpuFix consistently improves the dependence on harmful spurious features even for SOTA models with negligible impact on test accuracy.

2. Related work

When classifiers in safety-critical systems such as healthcare or autonomous driving are deployed in the wild [3], it is important to discover potential failure cases before release. Prior work has focused on corruption [31], adversarial robustness [11, 68, 43], and out-of-distribution detection [45, 32, 30]. There is less work on spurious features, although their potential harm might be higher.

Spurious features: It has been noted early on that classifiers show reliance on spurious features [15] e.g., a cow on the beach is not recognized [9] due to the missing spurious feature of grass. Other forms of spurious features have been reported in the classification of skin lesions [12], pneumonia [86], traffic signs [67], and object recognition [88]. Moreover, it has been shown that deep neural networks are biased towards texture [26] and background context [80], see [25] for an overview. [58] argues theoretically that spurious features are picked up due to a simplicity bias.

Detection of spurious features has been achieved using human label-intense pixel-wise annotations [48, 59, 60]. In [78], they use sparsity regularization to enforce a more interpretable model and use it for finding spurious features. [4] propose a complex pipeline to detect spurious features. While they scale to ImageNet, their analysis is limited to a few spurious features for a subset of 100 classes. [61, 44, 62] do a search on full ImageNet based on class-weighted "neural maps". The neural maps are used to add noise to "spurious" resp. "core" features but no significant difference in classification performance is observed. It remains unclear if their found spurious features are harmful, that is the feature alone triggers the decision for that class.

Interpretability methods: In recent years several interpretability methods have been proposed e.g., attribution methods such as GradCAM [57], Shapley values [42], Relevance Propagation [7], and LIME [51]. The use of these methods for the detection of spurious features has been analyzed in [2, 1] with mixed success and it has been argued that interpretability methods are not robust [19, 64, 27]. However, attribution methods work better for robust classifiers due to more interpretable gradients [22]. Another technique is counterfactual explanations [76, 75] which are difficult to generate for images due to the similarity to adversarial examples [68]. Thus visual counterfactual explanations are realized via manipulation of a latent space [56] or in image space [53, 6, 13] for an adversarially robust classifier. Visual counterfactuals for non-robust classifiers using diffusion models [35, 65, 33, 18] have been proposed in [5].

ImageNet: ImageNet [52] suffers from several shortcomings: apart from an inherent dataset bias [71], semantically overlapping or even identical class pairs were reported [34, 73, 10], e.g., two classes "maillot", "sunglass" vs "sunglasses", "notebook" vs "laptop" etc. We disregard such trivial cases of dataset contamination and focus on classes with harmful spurious features, in particular ones where only a small portion of the training set is contaminated.

3. Spurious features

A proper definition of spurious features is difficult. We describe two settings of harmful spurious features which appear in this paper. We denote by C_k the set of all images containing objects belonging to class k (assuming for simplicity that we have a deterministic problem and ignoring multi-labels). Let S be the set of all images containing a feature s (e.g. a bird feeder). It is a *correlated feature* for class k when $C_k \cap S$ and $S \setminus C_k$ are non-empty, i.e. the feature occurs frequently with the class object but there is no causal implication that appearance of s implies the appearance of the class object (a bird feeder in the image does not imply presence of a hummingbird). A *correlated feature* becomes a *spurious feature* when the classifier picks it up as feature of this class. Not every spurious feature is immediately harmful, even humans use context information [25] to get more confident in a decision. However, a spurious feature is *harmful* if the spurious feature alone is enough to trigger the decision for the corresponding class without the class object being present in the image. We consider two scenarios for a *harmful spurious feature* shown in Fig. 2.

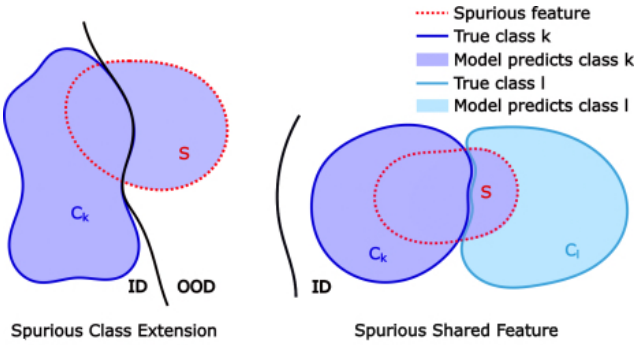

Figure 2: **Type of harmful Spurious Feature:** Left: The spurious feature s is taken up by the classifier which predicts class k on $C_k \cup S$ instead only on C_k. Right: The spurious feature s is shared between classes but appears more often in class k. The classifier associates S with class k and thus predicts class k also on $C_l \cap S$ instead of class l.

Spurious Class Extension: For this type of spurious feature (left in Fig. 2) the classifier picks up the spurious feature s for class k and predicts the class k even on $S \setminus C_k$ with high confidence (prediction of "hummingbird" for images showing a bird feeder but no hummingbird). The classifier predicts class k beyond its actual domain C_k and thus we call this a spurious class extension. While this spurious feature does not necessarily hurt in terms of test performance it can easily lead to completely unexpected behavior when the classifier is deployed in the wild.

Spurious Shared Feature: Here, two classes C_k and C_l share a spurious feature s (e.g., "water jet" for the classes "fireboat" and "fountain"). As there are more training images with feature s in C_k than in C_l the classifier associates S with class k and predicts class k for $S \cap C_l$.

The two types of harmful spurious features are not exclusive. A shared spurious feature s can at the same time lead to a spurious class extension, e.g., the object bird feeder leads to a spurious class extension of the class "hummingbird" (see Fig. 4) to images of bird feeders **without** hummingbirds. In the training set the hummingbird feeder appears only in images of class "hummingbird" but the hummingbird feeder has parts which mimic flowers and flowers are a shared spurious feature with bees. In Fig. 4 right top row, images of bees on a bird feeder are classified as "hummingbird" instead of "bee", so the spurious feature is strong enough to override the decision for the true class "bee" (spurious shared feature).

4. Finding spurious features via neural PCA and associated feature visualizations

First, we define our class-wise neural PCA (NPCA) which allows us to find diverse subpopulations in the training data, e.g., we checked that the bird feeder for "hummingbird" is visible in 15% of the training images (component 2), while another 15% contain a part of it (component 3), see Fig. 3 or, for more examples, App. F. Then we introduce our neural PCA feature visualization (which requires an adversarially robust model) and how we select NPCA components for human inspection. The identification of spurious features requires minimal human supervision, and our effective setup allows us to screen all ImageNet classes.

Adversarially robust model: Similar to [61], we use an adversarially robust model to find spurious features in ImageNet. The reason for this is that robust models have generative properties [74, 53, 6, 61, 13] in the sense that maximizing the predicted probability of a class in a neighborhood of an image leads to semantically meaningful changes. They also have more informative gradients [22] and thus attribution maps such as GradCAM [57] work better. We use the multiple-norm robust model of [13] as they claim it has the best generative properties. The generative properties of robust models are mainly used for the neural PCA feature visualization where we maximize the NPCA component of a class starting from a gray image. If a spurious feature appears without the class object, this is a strong indicator of a harmful spurious feature. A non-robust model would only produce semantically meaningless adversarial noise, hence we need a robust model for this part of our detection pipeline. Our detected spurious features are not specific to the robust model. We show that SOTA ImageNet models share the same spurious features (Fig. 4 and Sec. 7.2).

Class-wise neural PCA: Let $(x_i, y_i)_{i=1}^N$ be the training set, where $y_i \in \{1, \dots, K\}$ and K is the number of classes. We consider features of the penultimate layer $\phi(x) \in \mathbb{R}^D$ of a neural network for an input x. For a given class k and its associated weights $w_k \in \mathbb{R}^D$ in the final layer, we define

$$\psi_k(x) = w_k \odot \phi(x). \tag{1}$$

where $\odot$ is the componentwise product. Let $b \in \mathbb{R}^K$ be the bias vector of the final layer then the logit f_k of class k is:

$$f_k(x) = \sum_{j=1}^{D} w_{kj}\, \phi(x)_j + b_k = \langle \mathbf{1}, \psi_k(x) \rangle + b_k. \tag{2}$$

Let I_k be the index set of the training set of class k and $\bar{\psi}_k$ the class-wise mean, $\bar{\psi}_k = \frac{1}{|I_k|} \sum_{s \in I_k} \psi_k(x_s)$. The *class-wise neural PCA* allows us to identify variations in the set $\{\psi(x_r)\}_{r \in I_k}$ arising due to small subpopulations in the training set. In the class-wise neural PCA, we compute eigenvectors of the class-wise covariance matrix,

$$C = \sum_{s \in I_k} \left(\psi_k(x_s) - \bar{\psi}_k\right)\left(\psi_k(x_s) - \bar{\psi}_k\right)^T. \tag{3}$$

The eigenvectors, $v_1, \dots, v_D$ form an orthonormal basis of $\mathbb{R}^D$ and we write $\psi_k(x) - \bar{\psi}_k$ in this basis,

$$\psi_k(x) - \bar{\psi}_k = \sum_{l=1}^{D} v_l \left\langle \psi_k(x) - \bar{\psi}_k, v_l \right\rangle, \tag{4}$$

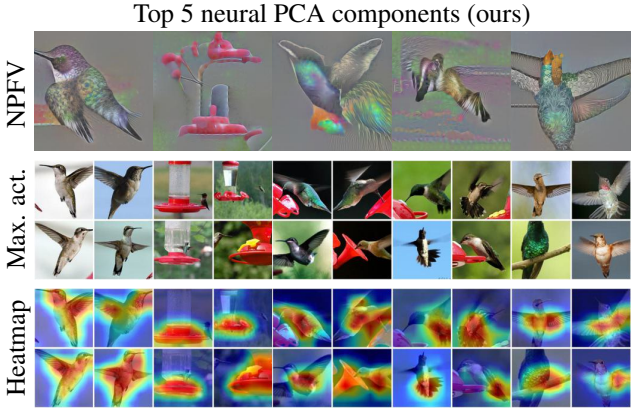

Figure 3: **Top 5 Neural PCA components for class hummingbird:** first row shows our neural PCA feature visualization (NPFV), second row shows four most activating training images of each NPCA component and the last row GradCAM for the NPCA component. Our NPCA components capture different subpopulations in the training data. Comp. 2 is identified as spurious feature "bird feeder", see NPFV and most activating training images (see also Fig. 4).

and define

$$\alpha_l^{(k)}(x) = \langle \mathbf{1}, v_l \rangle \left\langle \psi_k(x) - \bar{\psi}_k, v_l \right\rangle, \tag{5}$$

The logit $f_k(x)$ of the k-th class can then be written as

$$f_k(x) = \sum_{l=1}^{D} \alpha_l^{(k)}(x) + \left\langle \mathbf{1}, \bar{\psi}_k \right\rangle + b_k. \tag{6}$$

Thus for a given x, we can interpret $\alpha_l^{(k)}(x)$ as the contribution of the neural PCA component l of class k to the logit $f_k(x)$ of class k since the term $\left\langle \mathbf{1}, \bar{\psi}_k \right\rangle + b_k$ is constant for all inputs. Based on this we introduce a mitigation technique for spurious features without retraining in Sec. 5.

Neural PCA Feature Visualization: To identify semantic features corresponding to our neural PCA component l, we show the training images which attain the maximal values of $\alpha_l^{(k)}(x)$. Additionally, we generate an image $z_l^{(k)}$, which we call the **Neural PCA Feature Visualization (NPFV)** of feature l of class k, by maximizing $\alpha_l^{(k)}(x)$:

$$z_l^{(k)} = g + \text{argmax}_{\|\delta\|_2 \leq \epsilon} \, \alpha_l^{(k)}(g + \delta),$$

where g is a gray image (all channels equal to 0.5). Thus we maximize the feature $\alpha_l^{(k)}$ outgoing from a non-informative and unbiased initialization g. The optimization problem is solved using adaptive projected gradient descent (APGD) [16] with 200 steps. The budget for changes, $\epsilon = 30$, is small to avoid the overactivation of feature attacks maximizing the output of individual neurons [21, 63, 61], see Fig. 8. In Fig. 4 we show for each identified spurious feature, the corresponding NPFV, e.g., for "hummingbird" one can see the bird feeder but no hummingbird. The NPCA components together with the maximally activating training images are in principle sufficient to identify spurious features, but the NPFV is very useful to judge how *harmful* a spurious feature is. Therefore, we use an adversarially robust model, since a non-robust model would yield semantically meaningless adversarial noise as NPFV.

Selection of neural PCA components for human inspection: The penultimate layer of the robust ResNet50 we are using has 2048 neurons. Thus it is infeasible (and unnecessary) to investigate all neural PCA components. A strong criterion that one has found a harmful spurious feature is i) the NPFV shows mainly the spurious feature and not the class, and ii) the NPFV has high confidence. If ii) is not satisfied, then the NPFV is a spurious feature the classifier may have picked up, but it is not harmful in the sense that this feature alone causes the classifier to choose that class. Moreover, we noticed that the eigenvalues of the neural PCA, and the corresponding α values, decay quickly. Thus we compute the NPFV for the top 128 neural PCA components (having maximal variance) and then select the ten components which realize the highest confidence for their NPFV in the corresponding class. Note that we do not optimize the confidence when generating the NPFV but only $\alpha_l^{(k)}(x)$ which is part of the logit of the k-th class.

Identification of spurious neural PCA components via human supervision: For each ImageNet class k we show the human labeler the top 10 components. For each component l we show the NPFV $z_l^{(k)}$ and the 5 training images x_r of class k with the largest values of $\alpha_l^{(k)}(x_r)$. Moreover, we compute GradCAM [57] images for the NPFV and the five training images using the NPCA component $\alpha_l^{(k)}$ as score. The human marks a component as spurious if i) the NPFV shows dominantly an object not belonging to the class ii) the five training images show consistently this object, iii) the GradCAM activations are primarily not on the class object. The setup shown to the human labeler can be seen in App. B. The labeling of one class takes on average about 45 seconds, so the full labeling of all ImageNet classes took about 13 hours. The human labeler (researcher in machine learning) found in total 337 spurious components. Another human labeler checked all of them and removed spurious features in case of disagreement, resulting in 322 spurious features from 230 ImageNet classes.

5. SpuFix - Mitigation of spurious features

Once the spurious features are identified, the question is how one can mitigate that the classifier relies on them. One way is to identify the training images containing the spurious feature and then discard or downweight them during

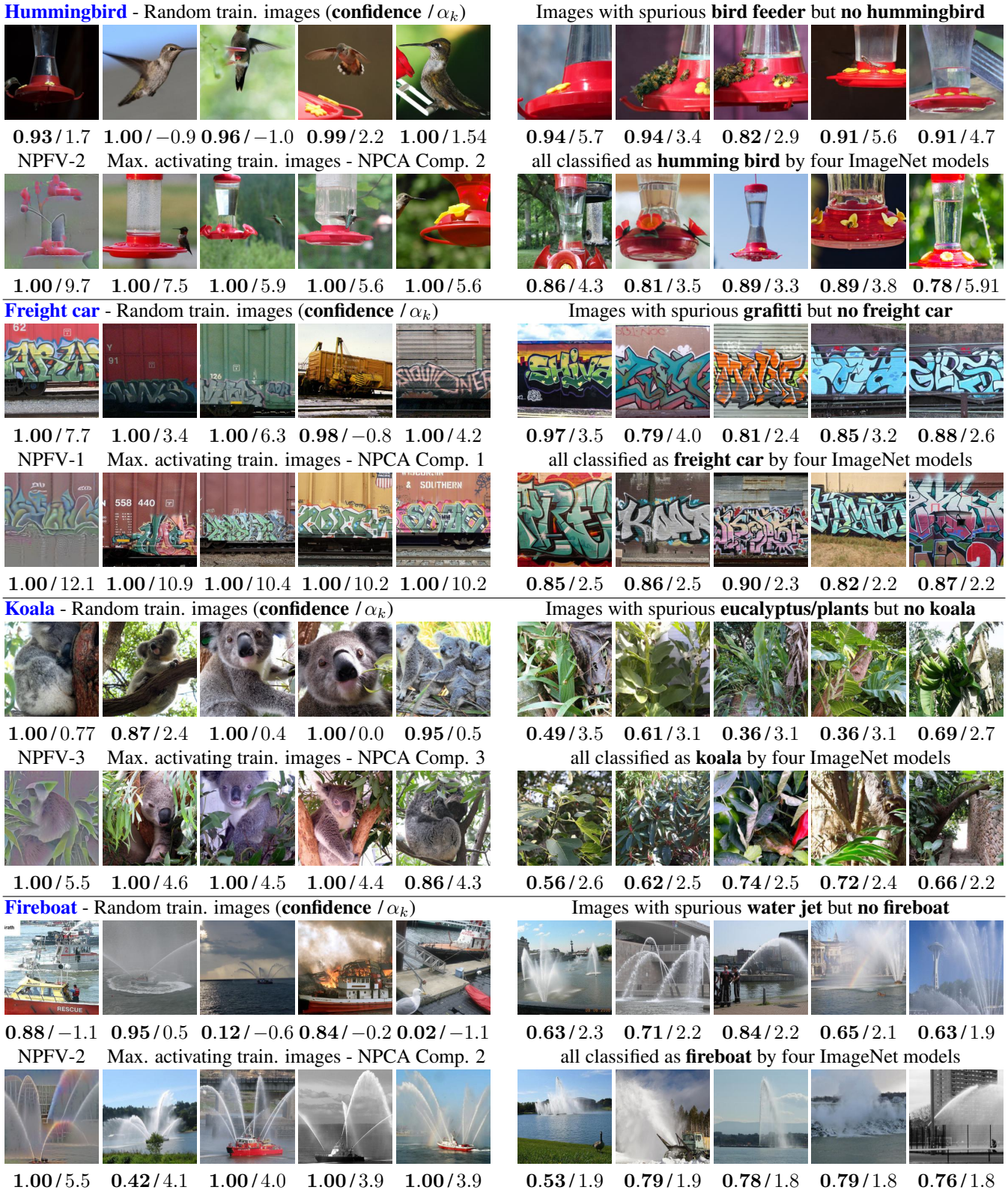

Figure 4: **Spurious features (ImageNet):** found by human labeling of our neural PCA components. For each class we show 5 random train. images (top left), the neural PCA Feature Visual. (NPFV) and 4 most activating train. images for the spurious feature component (bottom left). Right: four ImageNet models classify images **showing only the spurious feature but no class object** as this class.

training. However, this would require relabeling all spurious ImageNet classes which is not feasible. We could order the training set according to the value $\alpha_l^{(k)}$ of the corresponding neural PCA component which indicates how much of the spurious feature an image contains. While this would speed up the process significantly, it would still require a significant amount of manual relabeling. Can one do it also without any additional labeling? Yes, as described in Sec. 4 we can rewrite the logit of the k-th class as

$$f_k(x) = \sum_{l=1}^{D} \alpha_l^{(k)}(x) + \langle \mathbf{1}, \bar{\psi}_k \rangle + b_k. \tag{7}$$

For a spurious component l of class k, we use $\min\{\alpha_l^{(k)}, 0\}$ instead of $\alpha_l^{(k)}$ to remove its positive contribution from the logit (negative contributions are semantically different). After removal of the spurious features, the new logit becomes

$$f_k^{SpuFix}(x) = f_k(x) - \sum_{l \in \mathcal{S}_k} \max\{\alpha_l^{(k)}(x), 0\} \tag{8}$$

where $\mathcal{S}_k$ is the set of spurious NPCA components of class k. We denote this method as **SpuFix**. It significantly reduces dependence on spurious features, see Sec. 7.3.

Transfer of SpuFix to any ImageNet classifer: As described in Sec. 3, harmful spurious features are a result of subpopulations in the training data. While not every spurious correlation will be picked up by every model, most of our detected spurious features generalize to a wide range of classifiers (see Sec. 7). In the following, we show how SpuFix can be transferred to any given ImageNet classifier $\tilde{f}$ for which $\tilde{f}_k$ denotes the logit and $\tilde{\psi}_k$ the weighted penultimate layer of class k. The goal is to find a direction b in the weighted feature space $\tilde{\psi}_k$ of $\tilde{f}$ for every spurious NPCA component l corresponding to the eigenvector v_l of the original model, resp. $\alpha_l^{(k)}(x)$, and then truncate its positive component. To find this direction we maximize the covariance of the projection onto b and $\alpha_l^{(k)}$ over the training images of class k:

$$b_l^{(k)} = \underset{\|b\|_2=1}{\operatorname{argmax}} \sum_{s \in I_k} \left\langle b, \tilde{\psi}_k(x_s) - \bar{\tilde{\psi}}_k \right\rangle \alpha_l^{(k)}(x_s). \tag{9}$$

which has a closed form solution

$$b_l^{(k)} = \frac{\sum_{s \in I_k} \left(\tilde{\psi}_k(x_s) - \bar{\tilde{\psi}}_k\right) \alpha_l^{(k)}(x_s)}{\left\| \sum_{s \in I_k} \left(\tilde{\psi}_k(x_s) - \bar{\tilde{\psi}}_k\right) \alpha_l^{(k)}(x_s) \right\|_2}. \tag{10}$$

In contrast to the eigenvectors v_l, the matched vectors $b_l^{(k)}$ are not necessarily orthogonal. Thus before truncation the centered features $\tilde{\psi}_k(x) - \bar{\tilde{\psi}}_k$ need to be projected onto the subspace spanned by the $b_l^{(k)}$ and represented in the non-orthogonal basis $\{b_l^{(k)}\}_{l \in \mathcal{S}_k}$. We denote this representation by $P^{(k)}(x)$ (details in C.1). The logit of class k of the SpuFix version of $\tilde{f}$ is then:

$$\tilde{f}_k^{SpuFix}(x) = \tilde{f}_k(x) - \sum_{l \in \mathcal{S}_k} \max\left\{\langle \mathbf{1}, b_l^{(k)} \rangle P_l^{(k)}(x), 0\right\}. \tag{11}$$

In the case where $\tilde{f} = f$ (the robust model), we recover the original SpuFix truncation in (8) (see C.2). It turns out that SpuFix is even effective when the architecture is quite different from the ResNet50 we used for detection, e.g. ViT or VOLO, see Sec. 7.3 and Table 1 and 2.

6. Comparison to neural features of [61]

We compare our NPCA framework to the method of [61] to detect spurious features for ImageNet. As model, they use a ℓ_2-robust ResNet50. Let J_k be the set of training images classified as class k. [61] define the j-th component $m_j^{(k)}$ of the class-wise mean over predictions,

$$m^{(k)} = \frac{1}{|J_k|} \sum_{x_s \in J_k} \psi_k(x_s), \tag{12}$$

as the importance of the j-th neuron for class k.[1] Then, they order the neurons of the penultimate layer according to the score $m_j^{(k)}$ and consider the top-5 neurons of each class. The main difference to our approach is that they assume single neurons with maximal influence on the mean are capturing spurious features whereas our NPCA components are linear combinations of neurons that capture the variance *around* the mean. Given that the ResNet50 has only 2048 neurons for 1000 classes, some neurons are labeled as core **and** spurious feature for multiple classes simultaneously, even though the images are quite different. A major advantage of NPCA is that due to the orthogonality of the PCA components, we identify *diverse* subpopulations in the training data. As [61] use no constraints for the neurons, they often find very similar subpopulations. Hence, one may miss spurious subpopulations when only checking the top-5 components, see Fig. 5. Another difference is that they maximize the score $m_j^{(k)}$ for the training images x_r

$$w_j^{(k)} = x_r + \operatorname{argmax}_{\|\delta\|_2 \leq \epsilon} m_j^{(k)}(x_r + \delta),$$

whereas we maximize our NPCA component $\alpha_l^{(k)}(x)$ starting from a gray image to introduce no bias. Thus, we check if the classifier produces the spurious feature and not only enhances it on an image showing it already.

The third difference is that [61] want to identify *any* spurious feature while our goal is to find *harmful* ones. Thus, they use weaker criteria for deciding if a neuron shows a spurious feature: main criterion is if the neural activation

[1] The top-5 neurons do not change when using I_k instead of J_k.

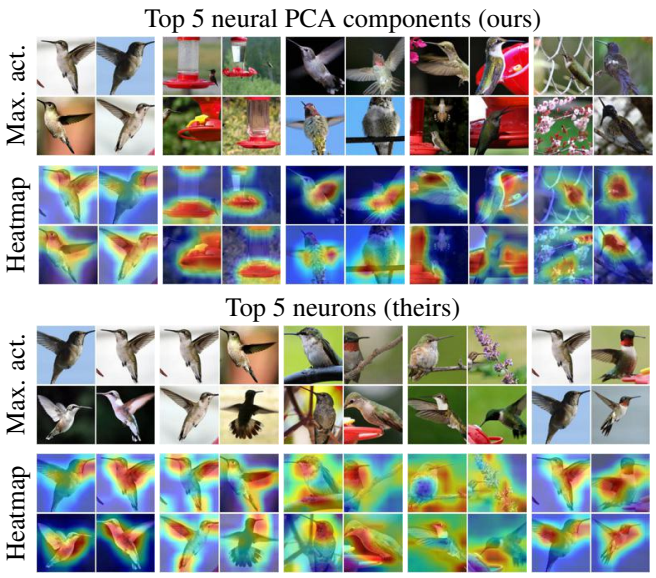

Figure 5: **Comparison of top-5 NPCA and top-5 neurons for class hummingbird for the robust model of [61]:** Our NPCA components identify diverse subpopulations in the training set whereas the top neurons show similar ones and the spurious bird feeder is not detected, see also App. A.

map based on $m_j^{(k)}$ is off the class object according to a majority vote of 5 human labelers. The visualizations $w_j^{(k)}$ are only used if the activation maps are inconclusive. In contrast, we require i) our NFPV to show the spurious feature, ii) the GradCAM based on $\alpha_l^{(k)}$ highlights mainly the spurious feature, and iii) our human labelers have to *agree* that the NPCA component is a harmful spurious feature.

As our criteria are more strict (in particular, that the NFPV shows the spurious feature is a strong criterion), it is not surprising that [61] find more spurious features (630 in 357 classes) than we do (322 in 230 classes). Moreover, the employed models are different and we examine top-10 NPCA components whereas they check top-5 neurons. Thus, the comparison is difficult and our novel dataset "SpuriousImageNet" could be biased towards spurious features which only we found. Hence, we compute our top-5 NPCA components for their robust ResNet50 for a direct comparison to their top-5 neurons and their found spurious features. In Fig. 5 we compare them for the class hummingbird where they do not find the spurious feature "bird feeder". In general, we observe that our found subpopulations are more diverse and thus we find more spurious features than they do when using their weaker criteria. In App. E we do an extensive comparison for all classes.

7. Experiments

In this section, we provide a qualitative and quantitative evaluation of our 322 detected spurious features in ImageNet, see Sec. 4. For the quantitative evaluation, we create the dataset "Spurious ImageNet", which allows checking the reliance of a given ImageNet classifier on spurious features. We also evaluate our mitigation strategy "SpuFix" which does not require additional labels or retraining of the classifier and can be transferred to other image classifiers.

7.1. Qualitative evaluation

For the qualitative evaluation, we visualize some of our found 322 spurious features, see Fig. 4. For each class, we show five random training images, the NPFV, and the four most activating training images of the neural PCA component labeled to be spurious. Additionally, we always show ten images which **only** show the spurious feature but **not** the actual class e.g., only the bird feeder (spurious) but no hummingbird (class). All ten images are classified as the corresponding class for the robust classifier we have used to compute the NPCA components and three non-robust ImageNet classifiers (ResNext101, Eff.Net B5, ConvNext-B, see also Tab. 2). This shows that our spurious features generalize from the robust classifier to SOTA ImageNet classifiers, indicating that the found spurious features are mainly due to the design of the training set, rather than failures in model training. Our novel validation by collecting real images with the spurious feature but without the class object which are consistently classified as this class directly shows the impact of harmful spurious features and has the advantage that it does not introduce artifacts via masking nor requires expensive pixel-wise segmentations.

Image Collection: The images showing only the spurious feature were obtained by sorting the 9 million images of OpenImages [38] by the value $\alpha_l^{(k)}(x)$ of the neural PCA component. We check the top 625 retrieved images classified by the robust classifier as the corresponding class if they are all classified as the same class by the additional non-robust classifiers *and* do not show the corresponding class. This is a quite strict criterion as spurious features can be shared across classes, e.g., twigs for birds, and thus agreement of classifiers is not granted and the images can show the spurious feature *and* the true class. Nevertheless, this procedure yields between 77 ("hummingbird") and 179 ("freight car") images of which we show a selection. For "hummingbird", a lot of these images show red flowers (without a hummingbird) which makes sense as the NPFV displays features of red flowers and due to the bias that OpenImages does not contain many images of hummingbird feeders. In these cases, we additionally retrieve Flickr images with appropriate text queries e.g., "hummingbird feeder" and filter them.

Spurious Class Extension: For "hummingbird", "freight car", and "koala" the spurious features significantly extend the predictions beyond the actual class (see Fig. 2). Bird feeders are classified as hummingbirds, graf-

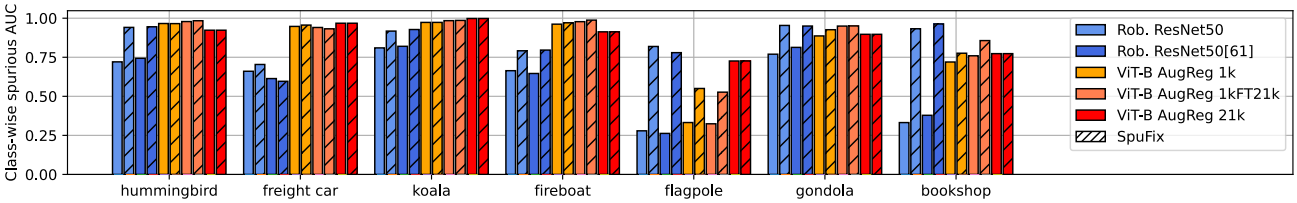

Figure 6: **Spurious Score:** we plot the AUC of different models for 7 out of 100 classes in "Spurious ImageNet" (see Fig 13-15): hummingbird (bird feeder/red flowers), freight car (graffiti), koala (plants/trees), fireboat (water jet), flagpole (US flag), gondola (house/river), and bookshop (storefront). The spurious features for hummingbird, freight car and koala which were not detected in [61, 62] are also spurious for their robust ResNet50 [61]. Training on ImageNet 21k or Fine-tuning from 21k (1kFT21k) decreases dependence on harmful spurious features but classes like flagpole and bookshop remain strongly affected. Our cheap mitigation strategy SpuFix improves the AUCs significantly for both robust ResNet50 but also improves the ViT-B variants trained/fine-tuned on ImageNet1k, especially for the difficult classes flagpole and bookshop.

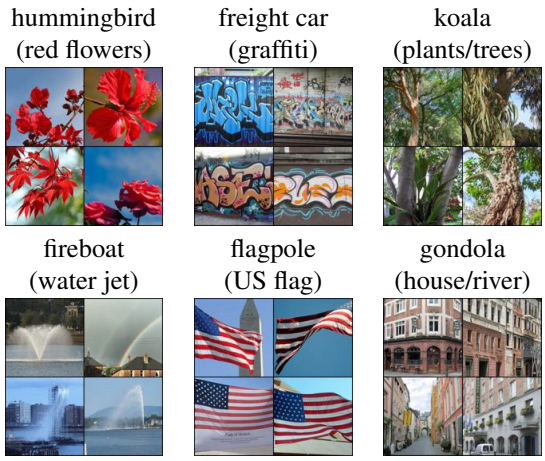

Figure 7: **Spurious ImageNet:** sample images from the dataset for 6 out of 100 classes showing the spurious feature but not the class object, see also App. G.

fiti as freight cars, and (eucalyptus) plants as koalas. This class extension cannot be detected by monitoring test performance and thus is likely to be noticed only after deployment. For "hummingbird", we see in Fig. 4 two images with bees on the bird feeder where "bee" is an ImageNet class (also a grasshopper for "koala"). Nevertheless, the spurious "bird feeder" feature of "hummingbird" overrules "bee" even though no hummingbird is present.

Spurious Shared Feature: The spurious feature "water jet" is shared among the classes "fireboat" and "fountain". It appears more frequently for "fireboat" (see Fig. 2) which leads to an in-distribution shift where now a large number of images of the "fountain"-class with a water jet are wrongly classified as "fireboat". The spurious feature "water jet" for "fireboat" has been found also in the Salient ImageNet dataset [61, 62]. However, they did not find spurious features for freight car and koala (in App. A we do a comparison). More examples are in App. F.

7.2. The Spurious ImageNet dataset

A key contribution of this paper is our novel evaluation of spurious features for image classifiers without requiring pixel-wise annotations [48, 59] or having to rely on the validity of neural heatmaps [61]. Instead, we use images from OpenImages to show that images only containing the spurious feature but not the class object are classified as this class. This has the advantage that we consider real images and thus provide a realistic impression of the performance of ImageNet classifiers in the wild. Adding noise [61] or masking [48, 44] image regions requires pixel-wise accurate annotations which are labor-expensive, masking only the object still contains shape information, and using masks avoiding this, e.g., a bounding box around the object, can hide a significant portion of the image which is unrealistic.

To allow for a quantitative analysis of the influence of spurious features on ImageNet classifiers, we collected images similar to the ones shown to illustrate the spurious features in Fig. 4. The images are chosen such that they show the spurious feature but not the class object. The only difference is that we relax the classification condition and only require two of the four classifiers (robust ResNet50, ResNext101, EfficientNet-B5, ConvNext-B) to predict the corresponding class. We select 100 of our spurious features and for each collect 75 images from the top-ranked images in OpenImages according to the value of $\alpha_l^{(k)}$ for which two human labelers agree that they contain the spurious feature but not class k and two out of four classifiers predict class k. We call the dataset **Spurious ImageNet** as it allows to check the dependence on spurious features with real images for ImageNet classifiers, see Fig. 7 and App. G for samples.

Spurious Score: A classifier f not relying on the spurious feature should predict a low probability for class k for the Spurious ImageNet samples, especially compared to ImageNet test set images of class k. Thus, for each class, we measure the AUC (area under the curve) for the separation

of images with the spurious features but not showing class k versus test set images of class k according to the predicted probability for class k. A classifier not depending on the spurious feature should attain a perfect AUC of 1, whereas a value significantly below 1 shows strong reliance. We report the mean AUC (mAUC) over all 100 classes in Tab. 1. All ImageNet models trained only on ImageNet1k are heavily influenced by spurious features. Thus, spurious features are mainly a problem of the training set rather than the classifier, and spurious features found with an adversarially robust model transfer to other ImageNet classifiers.

Pre-training on larger datasets: Some spurious features such as flag (flag pole), bird feeder (hummingbird), and eucalyptus (koala) are classes in ImageNet21k. Therefore, they should no longer be spurious for the other classes. Thus, we test if ImageNet1k-classifiers fine-tuned from an ImageNet21k model are less reliant on spurious features. The results in Tab. 1 and Fig. 6 suggest that the influence of spurious features is damped but they are far from being free of them. To check how much is lost due to fine-tuning we evaluate a ViT-B trained on ImageNet21k which has a mean AUC of 0.931 whereas the fine-tuned model has 0.917. This shows that fine-tuning does not hurt much. While finetuning from ImageNet21k improves the mean AUC, for several classes the dependence on spurious features is still significant, see also Fig. 16 how one has to be careful in the interpretation of higher AUC values. In addition to ImageNet21k, we also evaluate models trained on other large image datasets (JFT-300M[28], YFFC-100M, 1B Instagram[84], MIM[24], LAION-2B and LAION-400M[54]) using self-supervised learning or which are based on CLIP [49]. However, these models also do not achieve better spurious scores (Tab. 1 and Tab. 2). We evaluate a large number of SOTA models in App. D.

7.3. Evaluation of mitigation technique SpuFix

Fixing spurious features is a non-trivial task and can require a substantial labeling effort. We evaluate our SpuFix from Sec. 5 that does not require retraining or additional labels. The positive effect of this fix of spurious features (SpuFix) can be seen in Tab. 1 and Fig. 6. Compared to the original robust ResNet50 with a spurious mAUC of 0.630, the SpuFix version has a significantly better spurious mAUC of 0.763. Test set accuracy reduces by 0.6% but this is a rather positive effect, as several of the additional errors arise since the robust ResNet50 uses spurious features for its decision, e.g., for classes like "balance beam" or "puck" the class object is often not visible in the cropped test set images. In Table 2 we provide a large scale evaluation of the transfer of SpuFix to SOTA ImageNet models. We observe a consistently better mAUC on Spurious ImageNet, even for very large models fine-tuned from 21k or trained on other large datasets, e.g. SpuFix improves the mAUC of VOLO-D5 (87.1% acc.) trained only on 1k by 2.5%, or by 0.6% for EVA-G\14 CLIP 560 trained on MIM (91.9% acc.), as well as BeiT-L\16 fine-tuned from 21k (88.6% acc.) by 0.6%. Thus even SOTA models profit from our SpuFix with negligible difference in accuracy ($\leq 0.1\%$) and thus the use of SpuFix is recommended for any ImageNet model.

Model	Original		SpuFix	
	INet Acc. ↑	SpurIN mAUC ↑	INet Acc. ↑	SpurIN mAUC ↑
ImageNet1k				
Rob. ResNet50	57.4%	0.630	56.8%	**0.763**
Rob. ResNet50[61]	57.9%	0.651	57.2%	**0.764**
ConvNeXt-L[41]	84.8%	0.803	84.8%	**0.819**
ViT-B AugReg[66]	81.1%	0.850	81.1%	**0.859**
VOLO-D5 512[85]	87.1%	0.882	87.1%	**0.907**
ImageNet21kFT1k				
EfficientNetv2-L[70]	86.8%	0.893	86.8%	**0.898**
ConvNeXt-L[41]	87.0%	0.910	87.0%	**0.913**
ViT-B AugReg[66]	86.0%	0.917	85.9%	**0.925**
BEiT-L\16[8]	88.6%	0.921	88.6%	**0.927**
LAION-2B				
CNeXt-L CLIP 384[49]	87.8%	0.879	87.9%	**0.884**
ViT-L\14 CLIP[49]	88.2%	0.912	88.2%	**0.914**
MIM				
EVA-G\14 CLIP 560[24]	89.8%	0.919	89.8%	**0.925**
ImageNet21k				
ConvNeXt-L[41]	-	0.943	-	0.943
ViT-B AugReg[66]	-	0.931	-	0.931

Table 1: **Quantitative Evaluation on Spurious ImageNet:** ImageNet classifiers of different training modalities depend on spurious features in varying strength. The mAUC is the mean of AUCs for the separation of images containing the spurious feature but not class k versus test images of class k with the predicted probability of class k as score.

8. Conclusion

We have shown that large-scale identification of spurious features is feasible with our neural PCA components and neural PCA feature visualizations. With "Spurious ImageNet" we introduced a novel dataset to evaluate the dependence of ImageNet classifiers on spurious features based on real images. We demonstrated that our SpuFix method mitigates the dependence on harmful spurious features for any ImageNet classifier without costly labeling or re-training.

Acknowledgements We acknowledge support by the DFG, Project number 390727645, the Carl Zeiss Foundation, project "Certification and Foundations of Safe Machine Learning Systems in Healthcare". The authors thank the IMPRS-IS for supporting YN.

References

[1] Julius Adebayo, Michael Muelly, Hal Abelson, and Been Kim. Post hoc explanations may be ineffective for detecting unknown spurious correlation. In *ICLR*, 2022.

[2] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. In *NeurIPS*, 2020.

[3] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mane. Concrete problems in AI safety. arXiv:1606.06565v2, 2016.

[4] Christopher J. Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Finding and removing clever hans: Using explanation methods to debug and improve deep models. *Information Fusion*, 77:261–295, 2022.

[5] Maximilian Augustin, Valentyn Boreiko, Francesco Croce, and Matthias Hein. Diffusion visual counterfactual explanations. In *NeurIPS*, 2022.

[6] Maximilian Augustin, Alexander Meinke, and Matthias Hein. Adversarial robustness on in- and out-distribution improves explainability. In *ECCV*, 2020.

[7] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *JMLR*, 2010.

[8] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022.

[9] Sara Beery, Grant van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018.

[10] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with ImageNet? *arXiv:2006.07159*, 2020.

[11] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *ECML/PKKD*, 2013.

[12] Alceu Bissoto, Eduardo Valle, and Sandra Avila. Debiasing skin lesion datasets and models? not so fast. *arXiv:2004.11457*, 2020.

[13] Valentyn Boreiko, Maximilian Augustin, Francesco Croce, Philipp Berens, and Matthias Hein. Sparse visual counterfactual explanations in image space. In *GCPR*, 2022.

[14] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[15] Myung Jin Choi, Antonio Torralba, and Alan S. Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33(7):853–862, 2012.

[16] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[17] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR workshop*, pages 702–703, 2020.

[18] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

[19] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *NeurIPS*, 2019.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[21] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations, 2019.

[22] Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. On the connection between adversarial robustness and saliency map interpretability. In *ICML*, 2019.

[23] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. In *ICLR*, 2022.

[24] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *arXiv preprint arXiv:2211.07636*, 2022.

[25] 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.

[26] R. Geirhos, P. Rubisch, C.Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

[27] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *AAAI*, 2019.

[28] A Gupta, C Sun, A Shrivastava, and S Singh. Revisiting the unreasonable effectiveness of data. *arXiv preprint arXiv:1707.02968*, 2017.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[30] M. Hein, M. Andriushchenko, and J. Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *CVPR*, 2019.

[31] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[32] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

[33] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021.

[34] Sara Hooker, Yann Dauphin, Aaron Courville, and Andrea Frome. Selective brain damage: Measuring the disparate impact of model pruning. In *ICLR*, 2020.

[35] Pieter Abbeel Jonathan Ho, Ajay Jain. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[36] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

[37] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *ICLR Workshop*, 2017.

[38] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[39] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022.

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[41] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

[42] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, 2017.

[43] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[44] Mazda Moayeri, Sahil Singla, and Soheil Feizi. Hard ImageNet: Segmentations for objects with strong spurious cues. In *NeurIPS Datasets and Benchmarks Track*, 2022.

[45] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

[46] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

[47] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. *Proc. CVPR*, 2022.

[48] Gregory Plumb, Marco Tulio Ribeiro, and Ameet Talwalkar. Finding and fixing spurious patterns with explanations. *Transactions on Machine Learning Research (TMLR)*, 2022.

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[50] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *ICML*, 2019.

[51] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *KDD*, 2016.

[52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. License: No license specified.

[53] Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *NeurIPS*, 2019.

[54] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

[55] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[56] Kathryn Schutte, Olivier Moindrot, Paul Hérent, Jean-Baptiste Schiratti, and Simon Jégou. Using stylegan for visual interpretability of deep learning models on medical images. In *NeurIPS Workshop*, 2020.

[57] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[58] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *NeurIPS*, 2020.

[59] Rakshith Shetty, Bernt Schiele, and Mario Fritz. Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In *CVPR*, 2019.

[60] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: learning to overcome contextual bias. In *CVPR*, 2020.

[61] Sahil Singla and Soheil Feizi. Salient imagenet: How to discover spurious features in deep learning? In *ICLR*, 2022.

[62] Sahil Singla, Mazda Moayeri, and Soheil Feizi. Core risk minimization using salient imagenet. *arXiv:2203.15566*, 2022.

[63] Sahil Singla, Besmira Nushi, Shital Shah, Ece Kamar, and Eric Horvitz. Understanding failures of deep networks via robust feature extraction. In *CVPR*, 2021.

[64] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[65] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[66] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. In *TMLR*, 2022.

[67] Pierre Stock and Moustapha Cisse. Convnets and ImageNet beyond accuracy: Understanding mistakes and uncovering biases. In *ECCV*, 2018.

[68] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[69] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[70] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *ICML*, 2021.

[71] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011.

[72] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*, 2022.

[73] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From ImageNet to image classification: Contextualizing progress on benchmarks. In *ICML*, 2020.

[74] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

[75] Sahil Verma, John P. Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv:2010.10596*, 2020.

[76] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 2018.

[77] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[78] Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In *ICML*, 2021.

[79] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023.

[80] Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *ICLR*, 2021.

[81] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.

[82] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[83] Cong Xu and Min Yang. Adversarial momentum-contrastive pre-training. *arXiv preprint, arXiv:2012.13154v2*, 2020.

[84] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

[85] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *PAMI*, 2022.

[86] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLOS Medicine*, 15(11):1–17, 2018.

[87] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[88] Zhuotun Zhu, Lingxi Xie, and Alan L. Yuille. Object recognition with and without objects. In *IJCAI*, 2017.