DiagViB-6: A Diagnostic Benchmark Suite for Vision Models in the Presence of Shortcut and Generalization Opportunities

Elias Eulig[1,2,*] Piyapat Saranrittichai[1,3] Chaithanya Kumar Mummadi[1,3]
Kilian Rambach[1] William Beluch[1] Xiahan Shi[1] Volker Fischer[1,*]
[1]Bosch Center for AI (BCAI) [2]Heidelberg University [3]University of Freiburg

Abstract

Common deep neural networks (DNNs) for image classification have been shown to rely on shortcut opportunities *(SO) in the form of predictive and easy-to-represent visual factors. This is known as shortcut learning and leads to impaired generalization. In this work, we show that common DNNs also suffer from shortcut learning when predicting only basic visual object factors of variation (FoV) such as shape, color, or texture. We argue that besides shortcut opportunities,* generalization opportunities *(GO) are also an inherent part of real-world vision data and arise from partial independence between predicted classes and FoVs. We also argue that it is* necessary *for DNNs to exploit GO to overcome shortcut learning. Our core contribution is to introduce the Diagnostic Vision Benchmark suite* DiagViB-6, *which includes datasets and metrics to study a network's shortcut vulnerability and generalization capability for six independent FoV. In particular,* DiagViB-6 *allows controlling the type and degree of SO and GO in a dataset. We benchmark a wide range of popular vision architectures and show that they can exploit GO only to a limited extent.*

1. Introduction

Despite their state-of-the-art performance on object classification tasks, deep neural networks (DNN) are highly prone to shortcut learning [8, 33, 11]. Instead of learning holistic representations and decision rules that can generalize beyond the training data, DNNs overly rely on so-called *shortcut opportunities* (SO), which occur when the target class is highly correlated to one or very few easy-to-represent input factors [12]. This leads to poor generalization on many out-of-distribution (OOD) settings, e.g. ImageNet trained DNNs are biased towards texture and fail to generalize under texture-shape cue conflict evaluation [8].

While humans are also prone to shortcut learning in certain cases, such as object classification under context-based cue conflict settings [29], the biological model remains largely unaffected by shortcuts when it comes to predicting *basic* object factors of variation (FoV) such as `shape`, `hue` or `texture`. This "shortcut-immunity" w.r.t. basic FoV is just as necessary for intelligent systems; thus, efforts towards improving model generalization are of utmost importance.

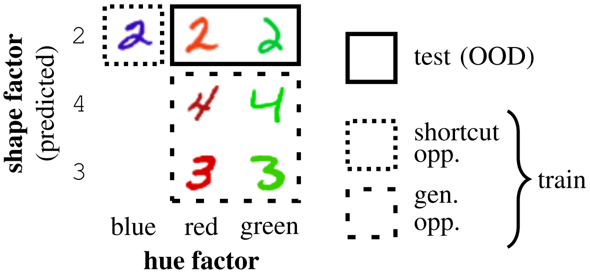

Figure 1: Exemplar study in our proposed benchmark. The network is trained to predict factor classes `2`, `4`, `3` for the `shape` factor with varying `hue`. All five depicted training combinations are uniformly shown during training. The shape `2` co-occurs solely with the `blue` class of the `hue` factor, which poses a shortcut opportunity. The shapes `4` and `3` occur uniformly with hue `red` and `green`; these combinations pose a generalization opportunity, since they reduce the predictiveness of the `hue` factor for the `shape` factor. Test accuracy is computed on examples from OOD factor combinations to evaluate a model's shortcut vulnerability in the context of the given generalization opportunity.

Existing literature studies shortcut behavior in DNNs mainly in the context of object classification. In this work, we address a more fundamental variant of shortcut learning, focusing specifically on the prediction of basic FoV themselves, similar to [12]. In the context of FoV prediction, we refer to different manifestations of a factor as *factor classes*. For example, "red" and "green" are factor classes for the factor `hue`, whereas "circle" and "elephant shape" are fac-

[*] Corresponding authors. Contact via `elias@eeulig.com` or `volker.fischer@de.bosch.com`.

tor classes for the factor `shape`. Object classes such as "elephant" or "car" are characterized by the co-occurrence of certain factor classes, e.g. "gray", "elephant shape", and "elephant texture" characterize an elephant. SO arise from this co-occurrence of different factor classes.

As stated in [7], SO are a property of real-world vision data. In this work, we propose to additionally consider *generalization opportunities* (GO), which are a relaxation of the strict correlation (or co-occurrence) between a target class and an input FoV at training time. For instance, consider the object "car" with factors `shape` and `color`; cars appearing in different colors during training would induce a GO compared to cars appearing only in one color. We refer to such cases as *compositional-based* GO. Correlations can also be violated in the form of outliers consisting of rare combinations of factor classes, e.g. a "white elephant". We refer to these cases as *frequency-based* GO.

A straightforward approach to introduce GO in the training data is data augmentation. For example, [21] applies random color transformations to the training images, removing a potential correlation between the target class and the factor color. However, data augmentation results in models that are invariant with respect to the augmented factor. Such models lose important information that may be needed to properly identify and reason about OOD samples. In contrast to being invariant, we argue that a good vision model needs to have an explicit representation of these FoV. Our work aims at analyzing a model's capability to exploit the GO already present in the data, as opposed to adding more GO to a dataset, as done in data augmentation. While several synthetic benchmark datasets for compositional generalization and cue-conflict settings already exist [12, 2, 28], none of them enables sufficient and systematic control over the SO and GO present in the dataset for a broad set of different visual object FoV.

Inspired by prior work on shortcut learning and compositional generalization [7, 12], we present a synthetic but diagnostic benchmark suite DiagViB-6[1]that includes different studies to evaluate a model's shortcut vulnerability under varying degrees of GO. Figure 1 illustrates an exemplar study in our benchmark. The benchmark suite contains an image-generating function that allows direct and independent control over the six basic, visual object FoV: `position`, `hue`, `lightness`, `scale`, `shape`, and `texture` (Fig. 2). Additionally, our framework provides a dataset-generating function that enables a user to control the nature of SO and GO appearing in a dataset. This is achieved by introducing different degrees of correlation between factors, and inducing co-occurrences of certain factor class combinations. Furthermore, the benchmark suite provides metrics to evaluate a model's shortcut vulnerability under different GO for each factor.

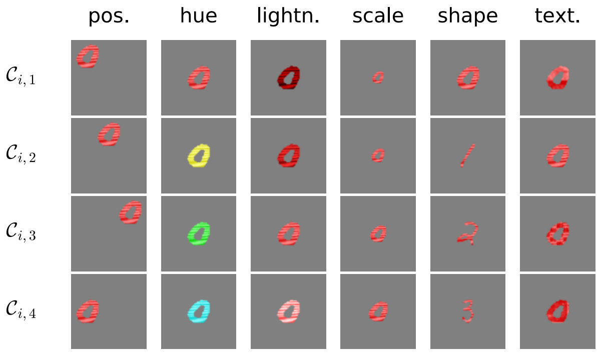

Figure 2: Image space traversal across all six FoV and four corresponding class labels used in this work. Along each column only the corresponding factor is varied, while all others are fixed. Note that some factors have more than the four classes shown here (refer Tab. 1).

We evaluate a wide range of common deep learning vision models on our benchmark and perform an exhaustive investigation of their shortcut vulnerability w.r.t. the six stated FoV. We show that while they exploit frequency GO, they exploit the more relevant compositional GO only to a limited extent. This holds true also for approaches specifically designed to counteract shortcut learning.

We admit that this benchmark suite does not *sufficiently* and directly prove a vision model's ability to generalize on real-world data (e.g. object classification on ImageNet). However, it serves as a critical diagnosis that is *necessary* in order to study a model's shortcut vulnerability and generalization ability under various controlled tasks and data setups. Ultimately, the design of our benchmark suite allows a user to control different degrees of SO and GO in a dataset (not commonly available in real-world data), in order to assess a model's behavior under different conditions.

Our contributions in this work can be summarized as follows: We propose a benchmark suite to create datasets that enable the user to independently combine six visual FoV, allowing explicit control over which SO and GO are present in the resulting data. We establish suitable metrics to evaluate both a model's shortcut vulnerability, and its capability to exploit GO in the data. Lastly, we provide empirical evidence that common vision architectures exploit GO only to a limited extent, especially compositional-based GO.

2. Related work

Compositional generalization in visual attribute prediction Our work is formulated under the general framework of visual attribute prediction. In contrast to classical object recognition, the task is to learn semantic attributes of a given object [6, 32]. By learning such class-agnostic visual object attributes, models can make useful predictions about object classes that have not been seen at training

[1] https://github.com/boschresearch/diagvib-6

time. We are interested in specific scenarios, called *compositional generalization*, in which only a limited number of attribute combinations are provided during training. The model needs to generalize during training in order to handle inputs with unseen attribute combinations during testing [27, 28, 35, 24, 1, 2, 36]. While most of these works use a multi-task setup requiring object and attribute annotations for every image during training, we consider a single-task prediction, in which only one factor is predicted. In addition, we consider predicting more fundamental factors (e.g. lightness, scale, hue) that are independent of each other and can be applied to all types of objects, as opposed to high-level visual attributes (e.g. glossy, furry, smooth).

Shortcut learning In recent years, multiple works have shown that DNNs are vulnerable to shortcut learning on several real-world and synthetic datasets [3, 8, 11, 7, 12]. As a result, many studies investigate the reasons for shortcut learning, and propose methods to alleviate such vulnerabilities. It is suggested in [11] that differences between humans and DNNs may arise from differences in the data that they see. [12] investigated how a model's representations are shaped by inductive biases in the presence of SO. A common attempt to overcome shortcut learning is to augment the training data with handcrafted transformations in order to reduce the importance of each individual factor (e.g. shape or texture) [8, 23]. This approach has further been extended to generative model-based augmentations [31, 34, 33]. Recently, [26] showed that for self-supervised learning, certain shortcut features can be removed automatically, under the assumption that such features are most vulnerable to adversarial attacks. Most of the aforementioned works on shortcut learning are benchmarked on black-box datasets (e.g. Stylized ImageNet [8]), leading to a limited, implicit knowledge of the SO and GO introduced in both the training and test data. In contrast, our DiagViB-6 enables explicit image generation, allowing shortcuts learned from individual FoV to be evaluated.

Benchmarks Numerous works exist on related areas, such as compositional generalization or disentanglement of FoV, that introduce datasets which allow to control image factors to some extent [12, 16, 2, 17, 20, 3, 13, 22, 15, 36]. However, these all contain shortcomings that we try to address in our work.

In contrast to [12, 2, 17, 20, 3, 13], our dataset contains a richer intra-factor class variation. E.g. for the `shape` factor, each digit class of MNIST specifies a factor class. During image generation we use different instances of each individual factor class. Similarly, for *red*, we use different tones of red. Other datasets only provide for example a single cylinder shape or red tone. For an overview of the variation provided for each of our six factors see Sec. A.1.

Unlike the datasets in [12, 20, 3, 22], which use only 2-3 FoV, DiagViB-6 includes six fundamental FoV that all good vision models should be shortcut-robust towards.

$\mathcal{F}_i$	$\mathcal{S}_i$	$\mathcal{N}_i$	$\mathcal{C}_{i,1}$	$\mathcal{S}_{i,1}$
position	$[0,1]^2$	9	top-left	$[0.1, 0.3] \times [0.1, 0.3]$
hue	$[0, 2\pi)$	6	red	$[345°, 15°]$
lightness	$[0,1]^2$	4	dark	$[0, 0.1] \times [0.4, 0.5]$
scale	$[0.69, 1.45]$	5	small	$[0.69, 0.74]$
shape	MNIST	10	'0'	digits '0'
texture	textures	5	tiles	tiles texture crops

Table 1: Overview of factors $\mathcal{F}_i$, respective factor spaces $\mathcal{S}_i$, and number of classes $\mathcal{N}_i$. $\mathcal{C}_{i,1}$ & $\mathcal{S}_{i,1}$ are exemplary factor classes and factor space regions of class label 1.

Some works, e.g. [12], investigate the internal feature representations of DNNs at certain layers, whereas our benchmark depends only on a model's predictions and is therefore architecture agnostic.

In contrast to multi-attribute prediction [2, 15, 36], we evaluate a model's shortcut behavior w.r.t. a single factor under different correlations to other FoV. This allows for a more structured and comprehensive analysis of shortcut behavior, where the interaction between individual factors can be investigated.

3. Diagnostic vision benchmark suite

This section describes our benchmark suite DiagViB-6, used to examine the shortcut vulnerability and generalization capability of DNNs over six different independent FoV. The suite consists of different image datasets tailored for different diagnostic studies, as well as suitable metrics for measuring shortcut vulnerability. We begin with an overview of the image generation process in our benchmark studies (Sec. 3.1), describe the benchmark setup in Sec. 3.2, introduce different studies in Sec. 3.3, and lastly establish the metrics to evaluate DNNs on our studies in Sec. 3.4.

3.1. Prerequisite

Fundamentals Datasets in our benchmark suite consist of images with a single object described by a fixed, predefined set of six independent factors. Each factor $\mathcal{F}_i$, $i \in \{1, \ldots, 6\}$ corresponds to a semantically meaningful image attribute: `shape`, `texture`, `hue`, `lightness`, `position` and `scale`. Every factor $\mathcal{F}_i$ is associated with a certain *factor space* $\mathcal{S}_i$ from which factor realizations $f_i \widehat{=} f_{\mathcal{F}_i} \in \mathcal{S}_i$ that describe the objects are sampled. For example, lightness of an object $f_{\text{lightness}}$ can be realized as a scalar sampled from $\mathcal{S}_{\text{lightness}} = [0, 1]$. We assign $\mathcal{N}_i$ discrete factor class labels for each factor, denoted as $\mathcal{C}_{i,j}$, $j \in \{1, \ldots, \mathcal{N}_i\}$, which correspond to regions $\mathcal{S}_{i,j} \subset \mathcal{S}_i \; \forall j$ and $\mathcal{S}_{i,j} \cap \mathcal{S}_{i,k} = \emptyset, j \neq k$. Similar to factors, each factor class $\mathcal{C}_{i,j}$ corresponds to a semantically meaningful attribute class (e.g. $\mathcal{C}_{\text{lightness},1}$ refers to "dark" and $\mathcal{C}_{\text{hue},3}$ refers to "green"). Note that our choice of classes for each factor is arbitrary and based on human intuition (similar to [12]) and thus our work does not evaluate a factor's general ability to act as a shortcut over another.

Table 1 provides an overview of the list of factors, their corresponding factor spaces, and the number of factor classes used throughout this work. As examples, the factor spaces $\mathcal{S}_{i,1}$ and the factor classes $\mathcal{C}_{i,1}$ are also provided. A comprehensive overview of factor spaces for all factor classes is provided in Sec. A.1.

Image generation An image is generated by defining a combination of factor classes $\mathcal{C} = (\mathcal{C}_{1,j}, \mathcal{C}_{2,k}, \ldots, \mathcal{C}_{6,l})$, one from each factor $\mathcal{F}_i$. This is done by first sampling factor realizations f_i from the corresponding regions $(\mathcal{S}_{1,j}, \mathcal{S}_{2,k}, ..., \mathcal{S}_{6,l})$ and then generating unique images from those sampled f_i using an image-generating function $I : (f_1, f_2..., f_6) \rightarrow \mathcal{I}$ with $\mathcal{I} \in [0,1]^{H\times W\times 3}$, where $H = W = 128$. The six FoV are incorporated into the image as follows: f_{shape} is always an instance of a digit in the MNIST dataset, which is then thresholded to yield a binary segmentation. From the two lightness values $f_{\text{lightness}} = (f^{(1)}_{\text{lightness}}, f^{(2)}_{\text{lightness}})$ two colors with equal hue f_{hue} are generated. The f_{texture}'s are crops of histogram-normalized grey-scale *texture* images and their pixel values serve as the coefficients of a convex combination between the two aforementioned colors, resulting in the final color. The object is then either up or down-sampled depending on the chosen *scale* realization f_{scale}, and placed at a specific *position* f_{position} on a greyscale background.

Figure 2 illustrates the variation over the six different FoV. In particular, the first column is generated with factor class labels $\mathcal{C} = (\mathcal{C}_{pos.,j}, \mathcal{C}_{hue,1}, \mathcal{C}_{lightn.,3}, \mathcal{C}_{scale,5}, \mathcal{C}_{shape,1}, \mathcal{C}_{text.,2})$ with $j \in \{1,2,3,4\}$. Similarly, the other columns show images varying one factor while keeping the classes of all other factors fixed. Note that the six different FoV are independent, and thus different images can be generated for each of the $\prod_i \mathcal{N}_i = 54000$ different factor class combinations.

3.2. Benchmark setup

In this section we lay out the fundamentals used in Sec. 3.3 to design and conduct studies analyzing different aspects of a DNN's shortcut behavior.

For all studies, we select a subset of three factor classes from the $\mathcal{N}_i$ available for each factor $\mathcal{F}_i$, which are then used during training, validation and testing. For example, as depicted in Fig. 1 and 3, we select the factor classes $\{\text{Blue}, \text{Red}, \text{Green}\}$ and $\{\texttt{2}, \texttt{4}, \texttt{3}\}$ for factors $\mathcal{F}_{hue}$ and $\mathcal{F}_{shape}$, respectively, and similarly for the other factors. This results in $3^6 = 729$ different factor class combinations out of the mentioned 54000 being used for a single experiment. Choosing the same number of factor classes across factors simplifies evaluation and removes the number of factor classes as a source of variation. To account for randomness introduced by this selection of factor classes, we draw five random *dataset samples* of selected factor classes, over which reported results are averaged.

As mentioned in Sec. 1, we want to evaluate a network's shortcut behavior when predicting an FoV in the presence of varying amounts of SO and GO in the training data. In this work, we consider the interplay between *all possible* pairings of two factors $(\mathcal{F}_i, \mathcal{F}_j)$. For each pairing, the task is to predict the class of the first factor $\mathcal{F}_i$, where SO and GO are induced by a specified co-occurrence pattern of factor class combinations for $\mathcal{F}_i$ and $\mathcal{F}_j$. For example, consider the setup in Fig. 1, with $\mathcal{F}_i$: `shape` and $\mathcal{F}_j$: `hue`. Only five out of the nine possible combinations of factor classes (dashed rectangles in Fig. 1) are shown during training, which induces a shortcut opportunity for the blue "2". During testing, we then evaluate the networks performance on the OOD combinations (solid rectangles in Fig. 1). The other 4 factors are not correlated with $\mathcal{F}_i$ or $\mathcal{F}_j$; their factor classes all appear with the same uniform probability. We conduct an exhaustive analysis of all possible factor pairings, resulting in $6 \times (6-1) = 30$ different settings, i.e. $6-1$ possible pairings for each target factor.

We generate validation data following the same distribution as the training data. In contrast, test data is designed to analyze a model's shortcut behaviour by violating the pairwise correlations existing in the training data, and thus contain OOD samples in the 3×3 matrix (cf. $\times$ for training and ◯ for testing in Fig. 3). We always generate 43 740 training, 8748 validation, and 10 000 test samples.

3.3. Benchmark studies

Having introduced the general structure of our benchmark, we now provide an overview of the five studies it enables (see Fig. 3 for illustration).

Zero Shortcut Opportunities (ZSO) The goal of this first study is to measure a network's factor classification performance in the absence of any SO. In the ZSO training data, each class of the target factor $\mathcal{F}_i$ co-occurs uniformly with all possible classes of $\mathcal{F}_j$, as indicated by the positions of $\times$s in the top-right of Fig. 3. Thus, $\mathcal{F}_j$ is not predictive for $\mathcal{F}_i$, $\forall i \neq j$. After training, the factor-classification performance of the model is tested on a dataset that shares the same distribution as the training data (see ◯ in Fig. 3).

Zero Generalization Opportunities (ZGO) This study can be considered an "opposite" of the ZSO study. In the ZGO training data, the target FoV $\mathcal{F}_i$ is perfectly correlated with $\mathcal{F}_j$, i.e. each class of $\mathcal{F}_i$ can only co-occur exclusively with one particular class of $\mathcal{F}_j$ (see ZGO in Fig. 3). Since both factors contain redundant information for the prediction task, $\mathcal{F}_j$ can be exploited as a shortcut to predict $\mathcal{F}_i$.

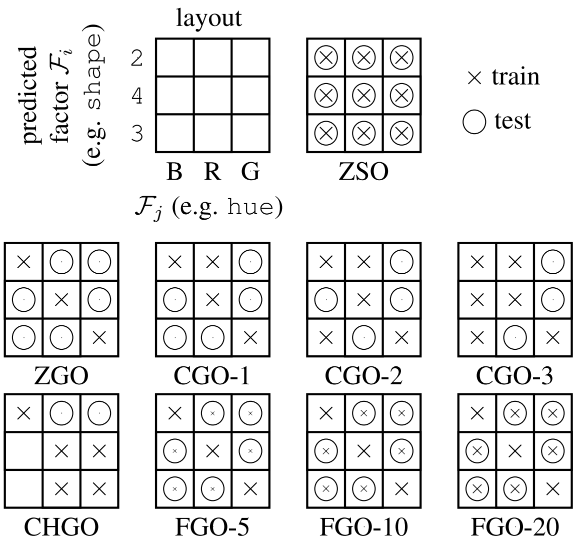
layout
predicted factor $\mathcal{F}_i$ (e.g. shape)
2
4
3
B R G
$\mathcal{F}_j$ (e.g. hue)
ZSO
× train
○ test
ZGO
CGO-1
CGO-2
CGO-3
CHGO
FGO-5
FGO-10
FGO-20

Figure 3: Schematic overview of conducted studies (cf. Fig. 1). For a combination of two factors $\mathcal{F}_i, \mathcal{F}_j$ (x- and y-axis), we select three factor classes (rows and columns) for each factor and predict the first (y-axis) factor. Different selections from the resulting nine combinations of factor classes for training and test datasets (× and ○) yield different GO and SO settings: The zero SO (ZSO), the three degrees of frequency GO (FGO) (increasing size of crosses ×, ×, × indicate increasing OOD frequency during training), the zero GO (ZGO), the three compositional GO (CGO-1/2/3) settings and the compositional hybrid GO (CHGO).

Compositional Generalization Opportunities (CGO) With the ZSO and ZGO studies capturing the two extremes, we continue with more realistic setups where both SO and GO are present in the training data. As mentioned in Sec. 1, one way to introduce GO is through compositions. The CGO study does exactly this, addressing compositional GO by successively adding more factor class combinations for $\mathcal{F}_i$, $\mathcal{F}_j$ into the training set. In particular, we generate three sub-studies CGO-c, $c \in \{1, 2, 3\}$ with increasing degree of GO (Fig. 3); the added combination is picked at random. For each class of the target factor, we hold out at least one unseen combination for testing, as indicated by the ○ symbols in each row in the CGO-c diagrams of Fig. 3. For $c = 0$, no GO are provided and thus CGO-0 is identical to the ZGO study discussed above. The goal of the CGO studies is to quantify a model's generalization capability, i.e. the capability to efficiently exploit the GO present in the data. We would expect a good vision model to perform well on these generalization tasks, especially as the number of SO decreases and the number of GO increases.

Compositional Hybrid GO (CHGO) This is a special case of CGO-2, for which both SO and GO are present, but explicitly separated, as depicted in Fig. 3 (bottom-left). One can think of this as a mixture of ZSO and ZGO, in which one of the classes of $\mathcal{F}_i$ is exclusively coupled with a certain class of $\mathcal{F}_j$ (ZGO), while the remaining two target factor classes co-occur uniformly with the remaining two classes of $\mathcal{F}_j$ (ZSO). The goal of CHGO is to examine whether a model can become immune to an extreme SO by exploiting GO available for other classes of the same factors.

As pointed out in Sec. 1, this heterogeneous mixture of SO and GO comes closest to real-world data. For instance, school buses might always be "yellow" while other cars appear in arbitrary colors besides yellow. Considering a model that predicts object shape, the yellow buses allow the model to shortcut its bus-shape prediction by solely relying on object color. A good vision model should now be able to exploit the provided GO, in the form of differently colored cars, to also predict the correct shape for an OOD "red" bus.

Frequency-based Generalization Opportunities (FGO) Orthogonal to the CGO study, GO may also be introduced in a frequency-based manner i.e., controlling the proportion of a correlation introduced in the training data (cf. [12, 4]). These sub-studies can also be seen as a transition from ZGO to ZSO by gradually increasing the frequency of correlations in the training data. We generate three sub-studies FGO-f, $f = \{5, 10, 20\}$, where the strict correlation from the ZGO is relaxed by means of low frequency violations (in f % of the samples) of this combination during training. Unlike CGO, all combinations are seen during training; we test on those combinations that are underrepresented.

3.4. Metrics

We benchmark a model on the studies described above, starting with evaluating the mean per-class accuracy $\text{acc}_{i,j}$ for the factor pair $(\mathcal{F}_i, \mathcal{F}_j)$, $i, j \in \{1, \dots, 6\}$, $i \neq j$, where $\mathcal{F}_i$ gets predicted. We define the prediction accuracy $\text{P}_{i,j}$ on the test dataset in a given study as the expectation over the five corresponding dataset samples: $\text{P}_{i,j} = \mathbb{E}\left[\text{acc}_{i,j}\right]$. In the case of ZSO, the prediction accuracy of $\mathcal{F}_i$ is defined as $\text{P}_i = \mathbb{E}\left[\text{acc}_i\right]$, with acc_i being the accuracy of predicting $\mathcal{F}_i$ on a single dataset sample. We also define two different metrics to summarize a factor's shortcut vulnerability with respect to the other factors:

i) The Factor-Aggregated Average accuracy (FAAvg)

$$\text{FAAvg}_i = \mathbb{E}\left[\operatorname*{mean}_{j\in\{1,\dots,6\},j\neq i} \text{acc}_{i,j}\right] \tag{1}$$

measures the average accuracy of predicting factor $\mathcal{F}_i$ over all possible correlations with other factors $\mathcal{F}_j, j \in \{1, \dots, 6\}, j \neq i$. It can also be seen as a measure of the average shortcut vulnerability of the factor on a given study.

ii) The Factor-Aggregated Minimum accuracy (FAMin)

$$\text{FAMin}_i = \mathbb{E}\left[\min_{j\in\{1,\dots,6\},j\neq i} \text{acc}_{i,j}\right] \tag{2}$$

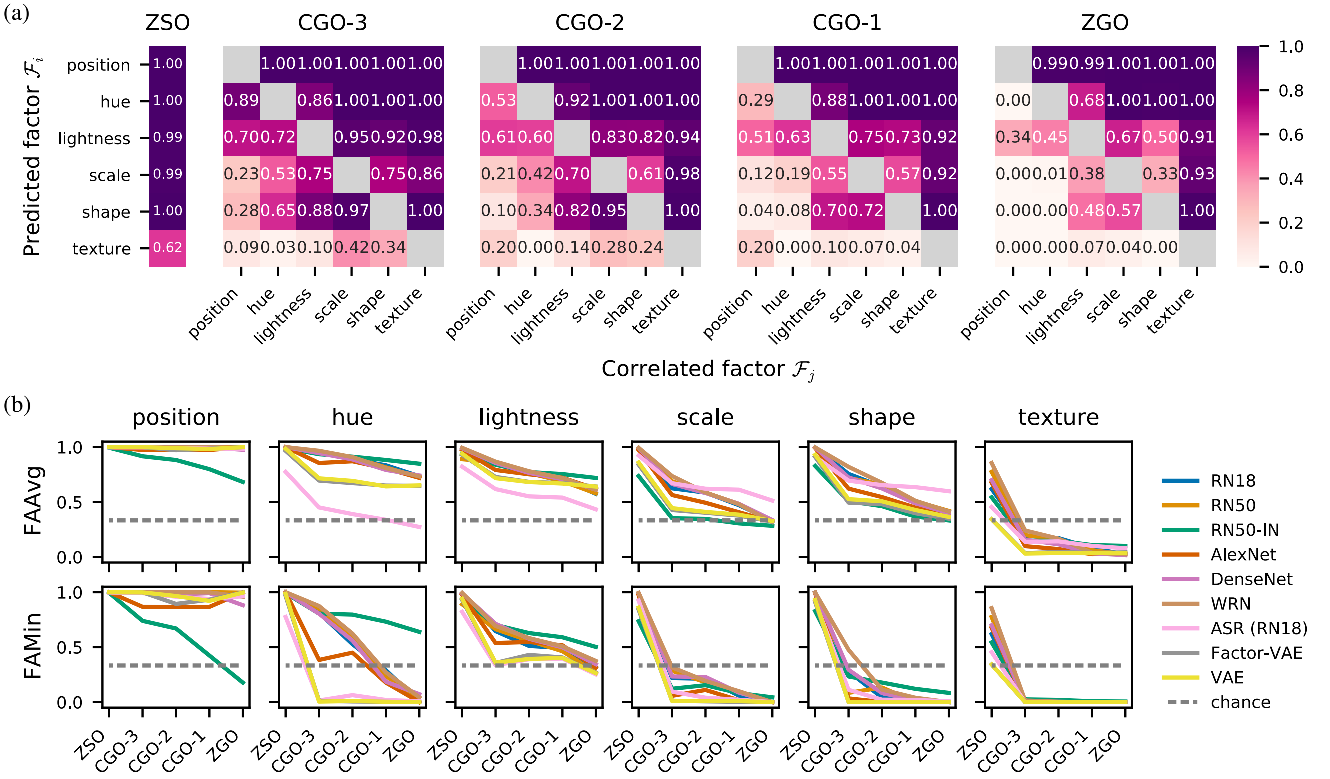

Figure 4: Best viewed in color. (a) For the RN18 baseline, we show mean accuracies P_i for all factors $\mathcal{F}_i$ on the ZSO study and mean accuracies $\mathrm{P}_{i,j}$ for all factor pairings $(\mathcal{F}_i, \mathcal{F}_j), i \neq j$ on the CGO-c and ZGO studies. (b) Aggregating model behavior over the studies, we compare all baselines using our benchmark metrics FAAvg and FAMin. Note that in (b) error bars are omitted to improve visibility of individual baselines; standard errors are reported in Sec. A.3 and A.4.

measures the minimum accuracy among the correlations, and can therefore be used as a measure of the maximum shortcut vulnerability of the factor on a given study. In the case of ZSO, $\mathrm{FAAvg}_i = \mathrm{FAMin}_i = \mathrm{P}_i$.

4. Baseline setup

This section provides an overview of the baseline network architectures we evaluate on our benchmark. We consider popular, vision-based DNN architectures (not pretrained) from PyTorch's *torchvision* package [30]: **AlexNet** [21], **ResNet18** (RN18), **ResNet50** (RN50) [10], **Wide ResNet50-2** (WideRN) [37], and **DenseNet-161** [14]. We also include RN50 with pretrained weights on ImageNet [5] (**RN50-IN**), with frozen convolutional layers and a randomly initialized fully connected layer to adapt to our benchmark. We also use the prior work **Automatic Shortcut Removal** (ASR) [26] as a baseline for our studies. In addition, since generative models, in particular those with interpretable factorised latent representations, are promising approaches to overcome shortcut learning on classification tasks [7], we evaluate a standard **VAE** [19] and a **Factor-VAE** [17] on our benchmark. More details on the training and setup of baselines are provided in Sec. A.2.

5. Results

We evaluate the baselines described in Sec. 4 on our benchmark suite described in Sec. 3.3. We start with the extreme cases of zero SO (ZSO) and zero GO (ZGO), followed by the other studies that introduce compositional and frequency-based GO to the training dataset.

ZSO and ZGO Results for the ResNet18 baseline on both these studies are presented in Fig. 4a. A comparison of all baselines using the metrics described in Sec. 3.4 is provided in Fig. 4b. The factor-wise results for each baseline are presented in Sec. A.3 and A.4.

The ZSO study evaluates how well a model can predict each factor in the absence of SO. This yields a single mean accuracy P_i for each factor $\mathcal{F}_i$, visualized as a vector on the very left in Fig. 4a. Here, RN18 achieves high accuracies ($\geq 99\,\%$) for predicting all factors except `texture`, for which the mean accuracy is lower $(62 \pm 6)\,\%$. A similar behavior can be observed for most other baselines (leftmost point in each plot of Fig. 4b) with a few exceptions: VAE, Factor-VAE and ASR are worse at predicting `texture`, with $34\,\%$, $34\,\%$, and $45\,\%$ accuracy, respectively. This is likely due to a limited capability in representing high-frequency structures (Sec. A.3). In summary, considering the aforementioned exceptions, all baselines can predict all factors to a reasonable extent.

The ZGO study evaluates a model's ability to predict a factor $\mathcal{F}_i$ when each of its three factor classes co-occurs exclusively with a certain factor class of another factor $\mathcal{F}_j$ during training. Each such pairing $(\mathcal{F}_i, \mathcal{F}_j)$ corresponds to a single cell in the rightmost matrix plot in Fig. 4a, where $\mathcal{F}_i$ are the rows and $\mathcal{F}_j$ are the columns. The color coded value in each cell indicates mean accuracy $\mathrm{P}_{i,j},\ i \neq j$ on OOD test samples, for which this correlation is violated.

For RN18, we observe that for some factor combinations (e.g. $(\mathcal{F}_i, \mathcal{F}_j)$: (`shape`, `texture`)) the model does not exploit the provided SO, and instead preferentially learns the actual task, leading to similar mean accuracies on OOD test samples compared to the corresponding ZSO accuracies for this factor. On the other hand, for other factor combinations (e.g. (`texture`, `shape`)) the model does exploit the provided SO, yielding OOD accuracies close to zero thus below chance level. `texture` and `position` factors play a special role: While `position` is exploited as a shortcut when predicting any other factor, `texture` is never exploited as shortcut, but is easy to shortcut by any other factor. Interestingly, $\mathrm{P}_{i,j} \approx 1 - \mathrm{P}_{j,i}$ seems to hold for all factor combinations, i.e., *if $\mathcal{F}_i$ is exploited as a shortcut for predicting $\mathcal{F}_j$, then $\mathcal{F}_j$ is not exploited as shortcut for predicting $\mathcal{F}_i$.*

We argue that during training either (a) the representation gets dominated by a single factor ignoring the other or (b) is a superposition of both factors, i.e., both factors share capacity of the representation. The extent a factor is represented depends on its overall "difficulty", i.e., `texture` appears relatively difficult to learn, and how strongly $\mathcal{F}_i$ and $\mathcal{F}_j$ compete over the same capacities. During testing, $P_{i,j} \approx 1 - P_{j,i}$ holds now for (a) but not for (b). For `lightness` and `position`, ZGO findings indicate (b), and a superposition of both factors is learnt, utilising different capacities.

This yields a dataset-dependent ranking among factors, which we use to order the rows of plots in Fig. 4a, starting with the most shortcut-robust factor `position`, and ending with the most shortcut-vulnerable factor `texture`. Since a model's ability to represent a factor in the presence of SO is related to its inductive bias, the ZGO study could be used to diagnose the efficiency of inductive biases of vision models in future work.

ZSO and ZGO findings transfer to most baselines, with a few exceptions: On RN50-IN, `position` is less often exploited as shortcut; the model instead exploits `hue` as shortcut when predicting `position` (Sec. A.4, matrix plots).

CGO We now evaluate the capability of common vision models to exploit GO provided in the form of additional factor-class combinations. Here, the three benchmark datasets (CGO-1,2,3) successively add factor-class combinations to the ZGO study. Results for RN18 are shown in Fig. 4a, and a comparison of all baselines using aggregated metrics is presented in Fig. 4b. The factor-wise results for each baseline are shown in Sec. A.4.

	pos.	hue	lightn.	scale	shape	text.
RN18	0 ± 0	2 ± 3	27 ± 21	-1 ± 14	2 ± 5	-1 ± 2
RN50	0 ± 0	3 ± 5	26 ± 21	-1 ± 16	2 ± 5	-2 ± 2
RN50-IN	-1 ± 11	5 ± 6	15 ± 17	-7 ± 11	0 ± 8	-1 ± 1
AlexNet	-4 ± 8	1 ± 4	24 ± 19	-1 ± 12	3 ± 6	-2 ± 2
DenseNet	2 ± 3	-3 ± 5	24 ± 21	-3 ± 13	2 ± 5	-1 ± 1
WRN	1 ± 1	-2 ± 4	23 ± 22	1 ± 14	3 ± 6	-2 ± 2
ASR	1 ± 1	12 ± 15	32 ± 18	1 ± 15	1 ± 2	0 ± 2
F-VAE	0 ± 0	2 ± 5	19 ± 17	-3 ± 11	1 ± 8	2 ± 1
VAE	0 ± 0	2 ± 5	18 ± 15	-3 ± 12	0 ± 9	2 ± 1

Table 2: Mean accuracy improvement and its standard deviation of the CHGO study over the ZGO study.

For RN18, we find that most factor combinations exhibit a monotonic improvement with increasing number of GO, e.g. $(\texttt{scale}, \texttt{hue})$: $0.01, 0.19, 0.42, 0.53$ for ZGO, CGO-1,2,3, respectively. However, for some factor combinations the accuracy on OOD cases does not improve with an increasing number of GO, and for some factor combinations, e.g. $(\texttt{texture}, \texttt{hue})$ or $(\texttt{shape}, \texttt{position})$ shortcut learning occurs also for CGO-3.

This can also be seen for RN18 in the two aggregated benchmark metrics FAAvg and FAMin in Fig. 4b. We find that FAAvg stays far below ZSO accuracy for most baselines up until three GO are added. For `texture`, FAAvg is below chance level for all baselines and studies, reflecting our earlier finding that all other factors are exploited as shortcuts when predicting `texture`. Moreover, FAMin stays below chance level for all baselines for `texture`, `shape` and `scale` factors, up until three GO are added.

In contrast to `texture`, for `position`, none of the baselines exploit shortcuts, with RN50-IN being the only exception (Fig. 4b, leftmost column). This phenomenon is likely due to ImageNet containing limited explicit localization information [9], facilitating a partial invariance of the learned representations in RN50-IN. From the CGO study, we conclude that all baselines are unable to exploit compositional-based GO for most FoV used in this work.

CHGO A special case of the CGO-2 study is the CHGO study, in which two classes of the predicted factor are provided with GO, while one class is presented with SO only (see Fig. 1 and 3). Comparing the accuracy on the OOD test samples for the latter with the average accuracy on the ZGO study (Tab. 2), we find no improvement for most factors on all baselines, and only minor improvements for `lightness`, albeit with large variance across the five dataset subsets. We conclude that all the baselines evaluated in this work do not transfer GO across classes.

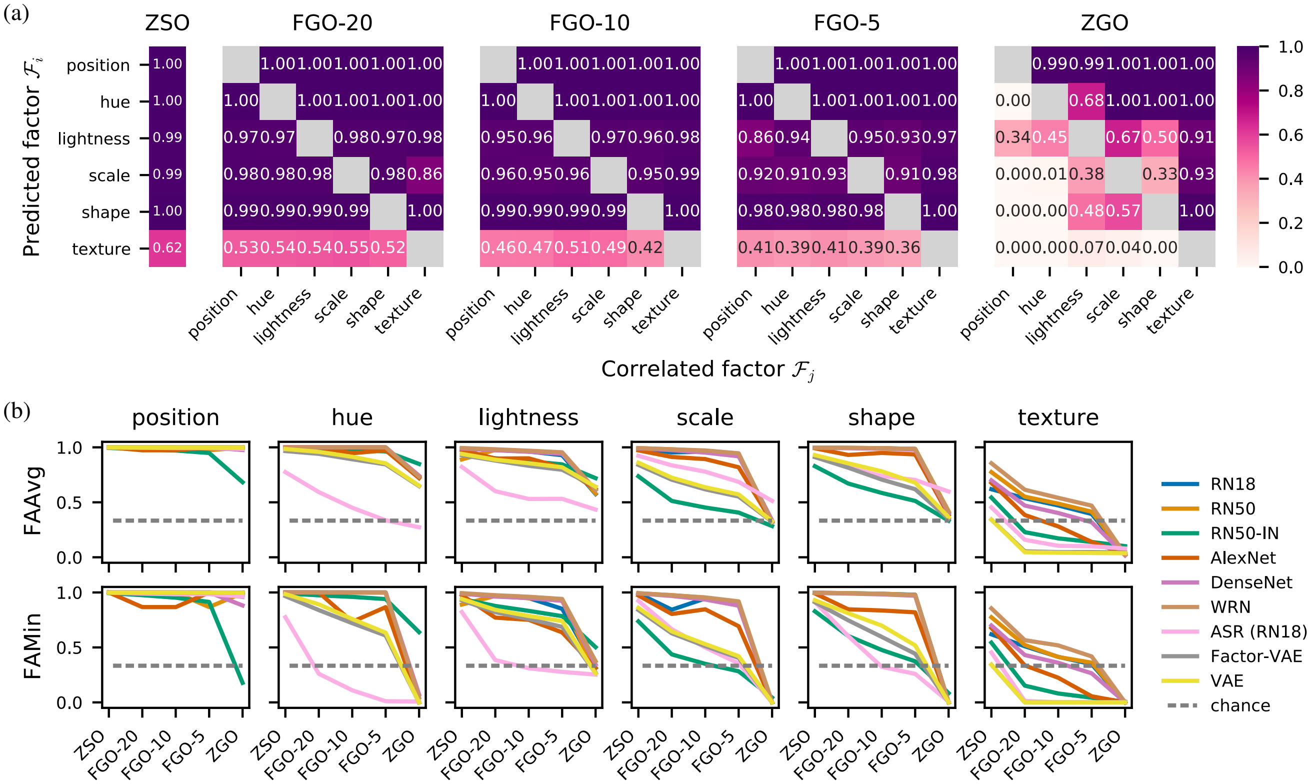

Figure 5: Best viewed in color. Similar to Fig. 4a and 4b, we show mean accuracies for the RN18 (a) and a comparison of all baselines (b) on the ZSO, FGO and ZGO studies. Error estimates are provided in Sec. A.3 and A.5.

FGO An orthogonal approach to providing GO through additional class-combinations during training is to relax the strict correlation between factors by means of low-frequency violations of this correlation at training time. We generated three benchmark datasets (FGO-f, $f \in \{5, 10, 20\}$), with f indicating the frequency of the correlation violation at training time. Results for RN18 and a comparison of all baselines are presented in Fig. 5a and 5b. The factor-wise results for each baseline are presented in Sec. A.5.

For the RN18, the classification for all factors improves significantly with only 5 % of the training samples for each class of the predicted factor violating the strict correlation from ZGO (e.g. $(\texttt{shape}, \texttt{hue})$: $0., 0.98$ for ZGO and FGO-5, respectively). When $f \geq 10$, none of the factors except `texture` are vulnerable to shortcut learning. A similar behavior can be observed for most factors and baselines (Fig. 4b), where we find that FAAvg is above chance for all factors except `texture` for $f = 5$. However, we find that ASR fails to exploit the presented GO for `hue` and `lightness` when compared to other baselines, a behavior which is likely attributed to the pixel-wise reconstruction loss that is used as a regularizer in the ASR objective. In summary, most baselines are able to exploit frequency-based GO to a certain extent on our benchmark.

6. Conclusion

We addressed the lack of a suitable benchmark to evaluate the shortcut behavior of vision models on a diverse set of basic visual FoV. To this end, we introduced DiagViB-6, a benchmark suite designed to evaluate two crucial model performance criteria: shortcut vulnerability and generalization capability. Our framework allows the user to create benchmark datasets by independently combining six visual FoV, thereby precisely controlling the SO and GO present in the dataset. Upon evaluation of the most commonly used vision architectures on DiagViB-6, we discovered that while they usually can exploit frequency-based GO, their ability to exploit compositional-based GO is limited. This finding also holds for some recent, promising approaches that have been proposed to overcome shortcut learning.

The design of DiagViB-6 is versatile, and leads to certain natural extensions. Three such promising future directions are: (a) including more complex correlation structures, potentially between more than two FoV; (b) transferring the benchmark design to other research areas like domain generalization and multitask learning; (c) extending to additional factors like background and natural corruptions. Lastly, we believe that our benchmark suite will inspire and help to build shortcut-robust solutions for vision models.

References

[1] Amit Alfassy, Leonid Karlinsky, Amit Aides, Joseph Shtok, Sivan Harary, Rogerio Feris, Raja Giryes, and Alex M. Bronstein. Laso: Label-set operations networks for multi-label few-shot learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[2] Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3

[3] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613, 2018. 3

[4] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *European Conference on Computer Vision*, pages 472–489. Springer International Publishing, 2018. 5

[5] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 6

[6] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009. 2

[7] 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, Nov 2020. 2, 3, 6

[8] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018. 1, 3

[9] K. He, R. Girshick, and P. Dollar. Rethinking imagenet pretraining. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4917–4926, 2019. 7

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[11] Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 19000–19015. Curran Associates, Inc., 2020. 1, 3

[12] Katherine Hermann and Andrew Lampinen. What shapes feature representations? exploring datasets, architectures, and training. In *Advances in Neural Information Processing Systems*, volume 33, pages 9995–10006. Curran Associates, Inc., 2020. 1, 2, 3, 4, 5

[13] Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. DARLA: Improving zero-shot transfer in reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1480–1490. PMLR, 06–11 Aug 2017. 3

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 6

[15] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[16] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 3

[17] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proceedings of the International Conference on Machine Learning*, volume 80, pages 2649–2658. PMLR, Jul 2018. 3, 6

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 3, 2015. 12

[19] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 6

[20] Tim Klinger, Dhaval Adjodah, Vincent Marois, Josh Joseph, Matthew Riemer, Alex'Sandy' Pentland, and Murray Campbell. A study of compositional generalization in neural models. *arXiv preprint arXiv:2006.09437*, 2020. 3

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012. 2, 6

[22] Y. LeCun, Fu Jie Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–104 Vol.2, 2004. 3

[23] Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan Yuille, and cihang xie. Shape-texture debiased neural network training. In *International Conference on Learning Representations*, 2021. 3

[24] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[25] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the International Conference on Machine Learning*, pages 4114–4124. PMLR, 2019. 12

[26] Matthias Minderer, Olivier Bachem, Neil Houlsby, and Michael Tschannen. Automatic shortcut removal for self-supervised representation learning. In *International Conference on Machine Learning*, pages 6927–6937. PMLR, 2020. 3, 6

[27] I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1160–1169, July 2017. 3

[28] Tushar Nagarajan and Kristen Grauman. Attributes as operators: Factorizing unseen attribute-object compositions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2, 3

[29] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007. 1

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[31] William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 581–597. Springer International Publishing, 2020. 3

[32] Olga Russakovsky and Li Fei-Fei. Attribute learning in large-scale datasets. In *European Conference on Computer Vision*, pages 1–14. Springer, 2010. 2

[33] Axel Sauer and Andreas Geiger. Counterfactual generative networks. In *International Conference on Learning Representations*, 2021. 1, 3

[34] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the GAN latent space. In *Proceedings of the International Conference on Machine Learning*, volume 119 of *PML*, pages 9786–9796. PMLR, 13–18 Jul 2020. 3

[35] Muli Yang, Cheng Deng, Junchi Yan, Xianglong Liu, and Dacheng Tao. Learning unseen concepts via hierarchical decomposition and composition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[36] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. 3

[37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. 6