Patch Shortcuts: Interpretable Proxy Models Efficiently Find Black-Box Vulnerabilities

Julia Rosenzweig[1], Joachim Sicking[1,2], Sebastian Houben[1], Michael Mock[1], Maram Akila[1]
[1] Fraunhofer IAIS, [2] Fraunhofer Center for Machine Learning
{first.last}@iais.fraunhofer.de

Abstract

An important pillar for safe machine learning (ML) is the systematic mitigation of weaknesses in neural networks to afford their deployment in critical applications. An ubiquitous class of safety risks are learned shortcuts, *i.e. spurious correlations a network exploits for its decisions that have no semantic connection to the actual task. Networks relying on such shortcuts bear the risk of not generalizing well to unseen inputs. Explainability methods help to uncover such network vulnerabilities. However, many of these techniques are not directly applicable if access to the network is constrained, in so-called black-box setups. These setups are prevalent when using third-party ML components. To address this constraint, we present an approach to detect learned shortcuts using an interpretable-by-design network as a proxy to the black-box model of interest. Leveraging the proxy's guarantees on introspection we automatically extract candidates for learned shortcuts. Their transferability to the black box is validated in a systematic fashion. Concretely, as proxy model we choose a BagNet, which bases its decisions purely on local image patches. We demonstrate on the autonomous driving dataset A2D2 that extracted* patch shortcuts *significantly influence the black box model. By efficiently identifying such patch-based vulnerabilities, we contribute to safer ML models.*

1. Introduction

Deep neural networks (DNNs) have demonstrated state-of-the-art performance and good generalization properties on a broad range of tasks. However, this generalization ability is not thoroughly understood yet and examples of unexpected failure are in stark contrast to the many successful application scenarios. These failure cases are symptoms of underlying, more fundamental difficulties that come along with recent learning systems.

Two main challenges for generalization are learned *brittle features* [27] and learned *shortcuts* [17] that are both over-specific for the training (and in some cases even the test) dataset. They differ regarding semantics: brittle feature are mostly considered to be imperceptive statistical artifacts while shortcuts refer to spurious correlations between rather high-level semantic concepts. While both concepts are not fully disjoint, we focus on the more semantic shortcuts here.

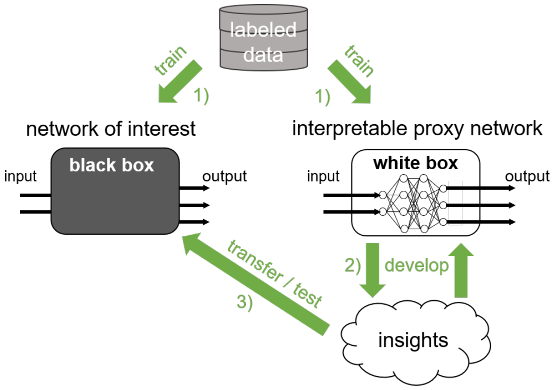


Figure 1: Detecting vulnerabilities of information-constrained black-box models. To avoid cumbersome analyses on the trained black box itself (step 1), we rely on an interpretable proxy model (step 1) that provides a set of so-called (image) patch shortcuts (step 2). Testing these (image) patch shortcuts on the original black box (step 3), we efficiently identify vulnerabilities.

To illustrate how networks exploit correlations for their decisions that have no semantic connection to the actual task, we outline three examples: first, the poster child of shortcut learning. Images (e.g. showing horses) that are in fact classified due to imprinted watermarks or copyright tags [33]. Second and more critically, models to detect Covid-19 in e.g. chest radiographs that rely on non-medical shortcuts (in particular outside the lungs) that are moreover hospital-specific [11]. Lastly, in the field of autonomous driving shown here this could e.g. be a patch showing parts of windows in a building which the model erroneously identifies as parts of "car", see left panel of Fig. 2.

Various explainability techniques were put forward to uncover such network vulnerabilities [45, 20, 51]. However, information-constrained black-box setups limit the applicability of most of them. At the same time, handling these black boxes gains importance as ML models are increasingly used in commercial contexts ranging from ML-as-a-service to ML appliances from technical providers. For, e.g., auditors and regulators with limited mandates such setups can pose additional challenges.

In this work, we propose an approach to systematically and efficiently detect learned shortcuts of a black-box classification model. Our approach, as depicted in Fig. 1, consists of three steps: in a first step, we train a white-box proxy model, namely BagNet that is interpretable-by-design. Next, we make use of its "locality" guarantee and systematically extract patch shortcuts, i.e. semantic vulnerabilities, of the proxy network. In the final step, we evaluate if, and to which extent, the identified patch shortcuts transfer to the black-box network of interest.

The remainder of the paper is organized as follows: First, we present related work on shortcut learning, interpretable-by-design architectures, approaches using proxy models and black-box vulnerabilities in Sec. 2. Next, we outline how to find and test patch shortcuts using an interpretable proxy network in Sec. 3. We instantiate this approach for a binary classification network in the automotive domain and conduct different ablation analyses to validate this patch-shortcut-based testing in Sec. 4. An outlook in Sec. 5 concludes the paper.

2. Related work

Our approach resides at the interface of several research directions, connections to which we detail here. We focus on shortcut learning and interpretability-by-design but also (patch-based) augmentations, the question of transferability of results, and adversarial vulnerabilities, specifically for black boxes, as the success of our shortcut patches is measured by decreased (black-box) DNN performance.

Shortcut learning In our work, we concentrate on finding vulnerabilities in the form of *shortcuts* in images. For a more general overview on shortcut learning see e.g. [17]. Recent work [11, 28] analyzes the aforementioned shortcuts in the domain of medical imaging and emphasize the necessity to validate the reliability of ML models.

Close to our idea is the work by Shetty *et al*. [46], in which the role of context for classification and segmentation is inspected and, in doing so, also the question of whether meaningful concepts were learned is addressed. However, they remove (human understandable) objects (e.g. cropping out the cars based on their segmentation mask to see how this affects sidewalk segmentation) whereas we (re-)insert (not necessarily human understandable) patches from the dataset into the image. Rosenfeld *et al*. [44] either insert some random, trained objects or remove objects and reinsert them into the images at a different location to see how this affects object detection. They observe that their "transplantations" have non-local effects to objects "far away".[1] We conduct a similar analysis systematically evaluating the inserted patches' sensitivity to chosen positions. In addition, we use an informed approach of selecting the parts which are to be inserted and in particular, we do not crop whole objects but only patches.

Interpretability-by-design and surrogate models Particularly related to our approach are *interpretable-by-design networks* such as invertible networks, in which information is processed in a bijective way [30, 3]. However, due to the special way of processing information, the applicability of these invertible networks is limited. Ghorbani *et al*. [19] extract semantic visual concepts important for the decision making. While their approach involves a segmentation that uses global information, BagNets aggregate purely local information in a linear fashion. Using human-understandable concepts, the ProtoPNet [7] is based on comparisons of the current input to prototypical class parts it learned. The network-in-network architecture [35] enables discrimination between local patches in the receptive field.

The idea of using *surrogates* for explaining black-box decisions is used in e.g. [15, 1, 47, 43]. Mohseni *et al*. [37] use a student model to predict the teacher's failure modes from the teacher's saliency maps in a white-box setup. While there are some heatmap methods for the black-box setup, see e.g. [42, 21], many common approaches rely on backward gradients and are thus not easily applicable as interpretability methods for our black-box setup. Often distillation [25, 16] is employed to train a surrogate model.

Patch-based augmentations Many approaches take small rectangular pieces of an image, so-called patches, as their starting point: In *image quilting* [13], small pieces of training images are recomposed to resemble inputs presented at inference, thus enabling texture synthesis and texture transfer for whole objects and images. *Patch-based augmentations* (e.g. [23, 12, 9]) are frequently used for model training aiming at improving performance or robustness.

Adversarial vulnerabilities The *black-box* setup with access to solely input-output pairs is common in the context of *adversarial attacks*, see e.g. [39]. Many approaches use *surrogate models* and *transferability*-based methods in which attacks are crafted on a white box and successfully

[1] This susceptibility to position is also observed by Kayhan *et al*. [31].

transferred to the target black box, see e.g. [41, 38].[2] Particularly, *Jacobian-based saliency map attacks* [40, 49, 8] are a related research direction. They employ a saliency map based on forward gradients to find "influential" pixels/features they seek to manipulate. We do not aim at pixel or feature manipulations and thus neither change nor optimize identified patch shortcuts. This key fact distinguishes our approach from most works on adversarial attacks. Additionally, we want to mention *semantic attacks* in which images are modified along human-understandable, semantic dimensions. For example, hue and saturation or colorization and texture of images, respectively, are randomly perturbed while keeping the semantic concept fixed [26, 4]. Jacobsen *et al*. [29] show that networks are often invariant to concepts that are relevant to the task and too sensitive to irrelevant ones.

Delineated as a reinforcement learning problem, Yang *et al*. [50] insert patches with textures into images to fool classifiers, an approach applicable to the black-box setting. We state two key differences: First, instead of learning or optimizing texture, we insert original image patches, hence, by design remaining in the domain of natural image patches.[3] Second, as mentioned above, we do not optimize for the size or the position of the patch. Other work in the field of attacks focuses on *patch-based attacks*, e.g. [6, 48, 34, 2, 36], in which, however, patches are optimized.

3. Patch shortcuts

In this section, we detail the conceptual approach of finding patch shortcuts for a given trained black-box classification network. We assume access to the training data of the black box and knowledge of the task it was trained for. Apart from this, we can only query the network output (oracle access). The structure of this section follows our proposed three-step approach shown in Fig. 1. We describe details regarding training the proxy network, automatically developing insights on it, and finally systematically testing these on the black box in the following subsections.

3.1. Training the interpretable proxy network

Dataset requirements Typically, annotations of a dataset are designed and used to train a specific task, e.g. class labels or segmentation masks. This, however, implies that all parts of the annotation are encoded in the network and cannot easily be used to identify shortcuts learned by the DNN. To investigate shortcut learning, we deliberately use an under-specified toy task, i.e. learn a binary classification task for a chosen class of interest (e.g. presence of cars) on a public dataset for semantic segmentation. In doing so, the segmentation mask serves as *meta annotation* to identify possible shortcuts (i.e. relevant image regions not occupied by cars) [10]. Although the overall approach can be applied for different tasks with the appropriate choice of interpretable network (see Sec. 5), we detail the concept for the case of image classification using the so-called BagNet trained on the aforementioned binary classification task as interpretable-by-design network.

BagNets BagNets are based on the ResNet-50 architecture [22] with some of the 3×3 convolutions replaced with 1×1 convolutions. This results in a strictly smaller receptive field than usual ResNet architectures possess.[4] BagNets perform an individual linear weighting of each "pixel" in the last feature map, effectively yielding patch-wise classification logits for the input image. Here, it is important to note that BagNets only aggregate information from image patches with the size of the receptive field, thus relying on truly local features and not aggregating or mixing evidence from across the entire image.[5] These patch-wise logit maps serve as internal, strictly feed-forward heatmaps (one for each class), using solely local patch information, which is computed in a single forward pass. See the heatmap on the left of Fig. 2 as an example for a "car" heatmap. Averaging the evidence from these heatmaps yields the final logit output of the BagNet for each class. Please note that to generate the described patch-logit heatmaps, we need to exchange two commutative layers in the BagNet definition so that the fully connected layer is applied before 2D average pooling. This allows us to save the representation after the fully connected layer as our heatmap.

Using BagNets as interpretable local-feature proxies We choose BagNets as interpretable proxies over producing e.g. occlusion-based forward heatmaps on the ResNet directly mainly for two reasons: First, this approach is much more efficient since a BagNet heatmap is produced in a single forward pass whereas a forward heatmap for a black box would require multiple rounds of inference with systematically shifted occlusions. Second and more important, we want to exploit the locality property of BagNets as they do not rely on global, long-distance features across the image but aggregate evidence by averaging local patches, see [5]. This enables explanations of model outputs in terms of each individual image patch irrespective of its position in the image.

[2] Naseer *et al*. [38] show that this transfer is not only possible across networks but also across domains.

[3] See also Hendrycks *et al*. [24], who gather natural images that confuse models in a dataset, and the real-world attack using stickers [14], which, however, does not allow for systematic analysis of failure modes.

[4] BagNet architectures with receptive fields of 9×9, 17×17 and 33×33 pixels, respectively, have been proposed in literature [5].

[5] Please note that the patch regions are partly overlapping due to the used stride in the BagNet.

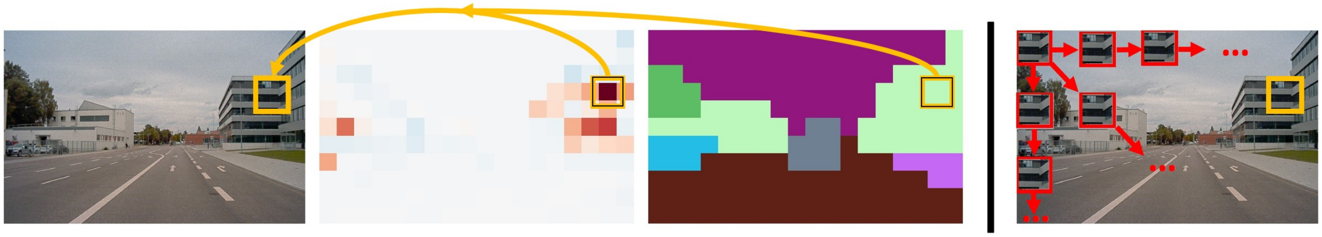


Figure 2: Identifying (first three panels) and testing (last panel) patch shortcuts. A patch is a BagNet patch shortcut (orange box in first panel) for class "car" if two conditions are fulfilled: First, it is highly relevant for BagNet's "car" prediction (dark red on heatmap in second panel), second, its semantic class is not "car" (light green segment in the third panel). To systematically test such a BagNet patch shortcut, we create a dedicated testing dataset (last panel).

3.2. Finding patch shortcuts

As a second step we use the trained BagNet to find semantic vulnerabilities. That is, for our selected class of interest, we identify image patches that are highly predictive for one class yet actually stem from a different class, i.e. semantic concepts that correlate with the class of choice although not related semantically. We detect these highly relevant patches in the following way: We perform inference on the whole test set using our adapted version of BagNet. We only consider those patches whose logit for the heatmap of the targeted class is above the 99% quantile $q^{0.99}_{\text{logit}}$ of logit values over all patches from the dataset and, thus, highly predictive of our class of interest. Next, for each obtained patch, we consult the corresponding part of the segmentation mask to verify whether it contains parts of the targeted class?.[6] If not, we identified a patch *shortcut* for this class whose prediction only correlates with the chosen class while holding no direct semantic relation. This procedure is depicted using an example on the left of Fig. 2.

3.3. Testing transferability to the black box

The last step of our approach aims to evaluate to which extent the identified BagNet shortcuts also constitute shortcuts, and thus semantic vulnerabilities, of the black-box network. For that, we propose the following procedure of constructing a testing image set: Using the set of images containing the patch shortcuts identified in Sec. 3.2, we consider the subset of images which are correctly classified by either the black-box or BagNet network as *not* belonging to the class of interest, denoting this set by $\mathcal{I}_{\text{shortcut}}$. The respective set of patch shortcuts extracted from the images in $\mathcal{I}_{\text{shortcut}}$ is denoted by $\mathcal{P}_{\text{shortcut}}$. Then, we automatically copy each patch shortcut from $\mathcal{P}_{\text{shortcut}}$ and re-insert it into the same image but at a new position. We then provide this manipulated image as input to the black-box network. We consider an insertion *successful* if it changes the prediction of the network to a misclassification.

To account for possible position-sensitivity of the black-box network, as observed in other work, e.g. [31, 44], we insert each identified BagNet patch shortcut from $\mathcal{P}_{\text{shortcut}}$ in a grid-based manner into many distinct positions of the original image. More concretely, we insert it into every position that corresponds to exactly one patch logit value ("pixel") of the BagNet heatmap. Note that we insert only a single patch at a time but at varying positions, resulting in a total number of manipulated images equal to the amount of "pixels" in the BagNet heatmap.[7] The testing image set obtained that way is referred to as $\mathcal{I}^{\text{aug}}_{\text{shortcut}}$. This procedure is depicted using an example on the right of Fig. 2. Finally, we statistically evaluate to what extent the black box is susceptible to the BagNet shortcut patches by conducting inference on $\mathcal{I}^{\text{aug}}_{\text{shortcut}}$ and thereby analyzing how often each patch leads to misclassifications if inserted into the image at all possible positions.

4. Experiments

Having outlined our approach to finding and testing black-box patch shortcuts, we now instantiate and evaluate it for shortcuts for the class "car" deploying a classification network from the automotive domain. The structure of the section follows the steps introduced in Sec. 3: After detailing the dataset and training procedures in Sec. 4.1, we generate the shortcuts in 4.2 and systematically test these on the black-box network in Sec. 4.3. To judge the effectiveness of our approach, baseline as well as further ablation studies are conducted in Sec. 4.4 and 4.5.

4.1. Training the interpretable proxy network

We first describe the custom dataset on which all experiments in this section are performed. Subsequently, the

[6] Classification towards a class could, e.g., be caused by small parts of an object of said class overlapping with the receptive field, a scenario we want to avoid with respect to shortcuts.

[7] Please note that one of these positions necessarily corresponds to the original image as the patch is replaced with itself.

training configuration for the interpretable white-box model (and also the black-box model) is presented.

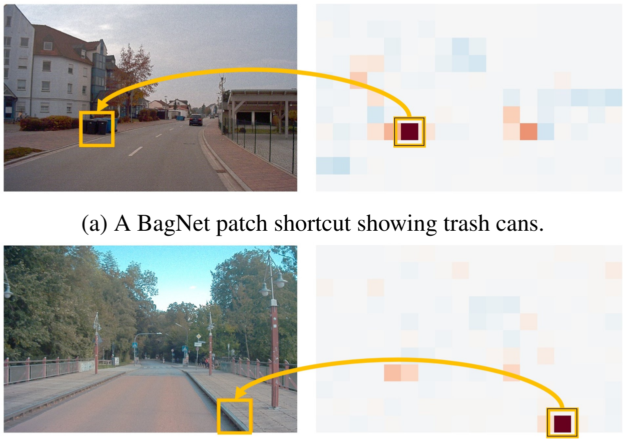
Dataset A2D2 [18] is a sequence-based traffic-scene dataset containing $41{,}277$ images that provides (among others) semantic segmentation ground truths. However, we do not intend to segment input images but to classify them either as "car" if one or several cars are displayed or as "no-car" otherwise. More specifically, images that feature at least 2% "car" pixels belong to class "car" and images containing 0.3% "car" pixels or fewer are counted as "no-car". All other images are discarded.[8] We refer to the resulting dataset as *binary-classification A2D2* or just *binary A2D2*. Since binary-classification A2D2 does not require the full segmentation ground truth, it allows us to use this ground truth as *pixel-wise meta-annotation* instead. To prevent both data leakage and domain shift between train and test data, we split each sequence from the dataset into three equally sized sub-sequences and apply a random 80:20 train-test split on sub-sequence level. For training and evaluation, the images are resized to 100×160 pixels and normalized.

Networks and training As our black-box model, we choose a ResNet-50 that is trained on binary-classification A2D2 for 150 epochs with a batch size of 128 using the Adam optimizer [32] with an initial learning rate of 0.001 and a binary cross-entropy loss. Unsurprisingly, the resulting model yields an almost perfect performance on the test set ($\text{acc} = 0.9748$, $F_1 = 0.9556$). We again point out that our analyses target the question of *how* these classifications are made and an over-simplified task provides a reasonable setup for such a study. In the following, we refer to this trained ResNet as black box (BB) network and do not make use of any network-internal properties.

As white-box proxy network, we employ an interpretable-by-design BagNet with a receptive field of 17×17.[9] Its training configuration does not differ from the one above, except for a smaller batch size of 64. This BagNet reaches a test performance of $\text{acc} = 0.9695$ and $F_1 = 0.9455$.

4.2. Finding patch shortcuts

Using the trained BagNet, our local-feature white-box proxy, we follow the procedure described in Sec. 3.2 to find patch shortcuts $\mathcal{P}_{\texttt{shortcut}}$: The BagNet's forward heatmap[10] and the semantic segmentation ground truth mask are compared and the combined selection criterion is applied (see Sec. 3.2). To investigate the semantic origin of patches we use a down-sampled version of the segmentation mask, compare Fig. 2.[11] Two examples of such patch shortcuts for class "car" and their corresponding BagNet heatmap are displayed in Fig. 3. We observe that many shortcut patches belong to the semantic classes of "nature object", "building" or "obstacle/trash". Moreover, edges seem to be common features of patch shortcuts (see e.g. the bottom row of Fig. 3).



(a) A BagNet patch shortcut showing trash cans.

(b) A BagNet patch shortcut showing a curb.

Figure 3: Identified BagNet patch shortcuts (orange frames). We combine the semantic pixel-wise annotations and BagNet's intrinsic forward heatmap (rhs: light-blue means low evidence for "car", dark red means high evidence for "car") to identify shortcuts (see lhs), i.e. "no-car" patches that BagNet correlates with class "car", see text for details.

4.3. Testing transferability to the black box

We follow the procedure described in Sec. 3.3 to evaluate if, and to what extent, the identified BagNet shortcut patches in $\mathcal{P}_{\texttt{shortcut}}$ are important for the black-box classifier. To enable systematic testing, we create the separate testing dataset $\mathcal{I}^{\text{aug}}_{\texttt{shortcut}}$ for each identified patch shortcut from $\mathcal{P}_{\texttt{shortcut}}$ by duplicating the respective image (from $\mathcal{I}_{\texttt{shortcut}}$) and copying the identified patch shortcut to different positions (see last panel of Fig. 2). We evaluate the black-box model on both the undistorted image dataset $\mathcal{I}_{\texttt{shortcut}}$ (that only contains the naturally occurring shortcuts) and the patch-shortcut-augmented dataset $\mathcal{I}^{\text{aug}}_{\texttt{shortcut}}$ and compare their (normalized) confusion matrices (see bottom row of Tab. 1). For comparison, we also report

[8] In particular, this means that an image labeled as e.g. "no-car" can still contain very few car pixels, which are however negligible w.r.t. the total image area.

[9] We use the BagNet architecture provided here: https://github.com/wielandbrendel/bag-of-local-features-models

[10] The heatmap is of size 11×18 pixels in our case.

[11] If the part of the segmentation mask corresponding to the patch contains the class in question, i.e. "car" or related classes, we count the down-sampled result as "car". Otherwise a majority voting among the other classes present in the patch is used.

the respective results when applying the BagNet instead of the black-box model (see top row of Tab. 1). For both networks, a shift from true negative (TN) to false positive (FP) can be observed after inserting the patch shortcuts. As expected, true positive (TP) images are mostly unaffected by patch-shortcut insertions, see the virtually unchanged second and fourth row of Tab. 1. This preliminarily shows that the identified BagNet shortcuts also constitute shortcuts for the black box as they are able to provoke misclassifications.

			Predicted Class			
			$\mathcal{I}_{\text{shortcut}}$		$\mathcal{I}^{\text{aug}}_{\text{shortcut}}$	
			no-car	car	no-car	car
Actual Class	BagNet	no-car	0.63	0.37	0.35	0.65
		car	0.01	0.99	0.00	1.00
	BB	no-car	0.90	0.10	0.80	0.20
		car	0.00	1.00	0.00	1.00

Table 1: Normalized confusion matrices of BagNet (first two rows) and the black box (BB, last two rows) before (first two columns) and after (last two columns) insertion of BagNet patch shortcuts. Note that each identified BagNet patch shortcut is inserted into $11 \times 18 = 198$ distinct positions in the image. Hence, one insertion position corresponds to exactly the original image.

4.4. Random image patches as baseline

To check the soundness of our approach, we compare the set of patch shortcuts, $\mathcal{P}_{\text{shortcut}}$, with a *random* "no-car" patch set, denoted as $\mathcal{P}_{\text{random}}$, that is obtained using random "no-car" patches in the $\mathcal{I}_{\text{shortcut}}$ images from the 50% logit quantile.[12] In total, we consider image patches from 96 distinct images for this analysis.

For both BagNet and the black-box model, we report the mean and median numbers of successful patch insertions for patches in $\mathcal{P}_{\text{shortcut}}$ (first column of Tab. 2) and patches in $\mathcal{P}_{\text{random}}$ (second column of Tab. 2). We find patches from $\mathcal{P}_{\text{shortcut}}$ to be more successful by a big margin.[13] Note that all example images in this work display patches from $\mathcal{P}_{\text{shortcut}}$ (in orange frames) that are among the most successful ones on the black box.

		Patch Set			
		$\mathcal{P}_{\text{shortcut}}$	$\mathcal{P}_{\text{random}}$	$\mathcal{P}^{\text{mean}}_{\text{shortcut}}$	$\mathcal{P}^{\text{shuffled}}_{\text{shortcut}}$
BagNet	mean (↑)	**67.42**	10.65	16.84	12.24
	median (↑)	**28.50**	1.00	3.50	1.00
BB	mean (↑)	**20.59**	11.86	7.98	5.27
	median (↑)	**17.00**	7.00	1.67	1.25

Table 2: Mean and median number of successful patch insertions for the BagNet and the black box (BB) per image (higher is better as this means that the network is more susceptible to the patch). We report the results from our main and baseline experiment (first two columns) as well as from the further ablation study (last two columns, see Sec. 4.5 for details). Note that each patch is inserted into $11 \times 18 = 198$ positions in the image.

Next, we inspect the origin of the patches in $\mathcal{P}_{\text{random}}$ and $\mathcal{P}_{\text{shortcut}}$ and study the *position sensitivity* of the black-box network in more detail (see Fig. 4).

Regarding the patch origin, we observe that the informed and more successful patches in $\mathcal{P}_{\text{shortcut}}$ stem from two triangle-shaped regions of the input images, indicating that semantic concepts to the sides of the road contain shortcuts. To verify our procedure, we also show the origins of the patches from $\mathcal{P}_{\text{random}}$ which exhibit uniform distribution as expected.

For the black-box model, we observe a high sensitivity to patches inserted in the bottom part of the image (corresponding to common locations of cars on the road) and almost no effect when patches are inserted in the upper part of the image, i.e. in the region above the street level. Moreover, the left hand side of the bottom part (corresponding to the oncoming lane) shows the highest sensitivity to patch insertions. Overall, we observe that patches in $\mathcal{P}_{\text{shortcut}}$ are less susceptible to the insertion location compared to patches from $\mathcal{P}_{\text{random}}$. This lends credence to the fact that our patches in $\mathcal{P}_{\text{shortcut}}$ carry relevant shortcut information the black box is sensitive to.

4.5. Further ablation analyses

In the above experiments, we find insertions of patches from $\mathcal{P}_{\text{shortcut}}$ to be more effective compared to those from $\mathcal{P}_{\text{random}}$. However, all these patch insertions introduce edge effects w.r.t. the surrounding image information and thus artifacts that might influence the behavior of the network. To better understand the impact of such artifacts, two ablation experiments (based on the patches in $\mathcal{P}_{\text{shortcut}}$) are performed: We either shuffle the pixels inside the shortcut patch before insertion or replace them by their mean values, denoting the obtained patch sets by $\mathcal{P}^{\text{shuffled}}_{\text{shortcut}}$ and $\mathcal{P}^{\text{mean}}_{\text{shortcut}}$,

[12] Please note that we use only such images that contain patches from both logit quantiles.

[13] As expected, this effect is even stronger for the BagNet since it was used for patch selection. Please note that due to the overlapping receptive field of the BagNet, every insertion also slightly manipulates patch logits of neighboring patches in the heatmap by introducing edges and thus artifacts in neighboring patches. Thus, not all the insertions can be expected to be (equally) successful on the BagNet.

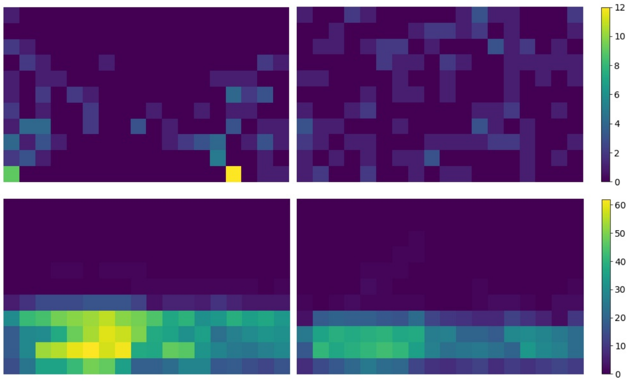


Figure 4: Analysis of the patch sets $\mathcal{P}_{\text{shortcut}}$ (left panel) and $\mathcal{P}_{\text{random}}$ (right panel) regarding position. The heatmaps (blue is rare, yellow is frequent) on the top visualize the origins of the patches in the respective patch set and the ones on the bottom show the positions where patch insertions switched the respective black-box prediction from "no-car" to "car". In this regard, patches from $\mathcal{P}_{\text{shortcut}}$ are clearly more effective than random ones from $\mathcal{P}_{\text{random}}$ as they can be placed more "freely", i.e. in more possible positions, in order to provoke misclassification.

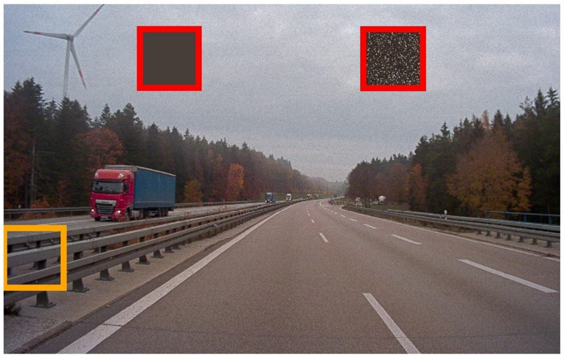


Figure 5: Illustration of a patch shortcut (orange frame) that is averaged (left red frame) or shuffled (right red frame) before being inserted into another part of the image. Different from this visualization, we never insert more than one patch into an image.

respectively. Shuffling the patch pixels removes the spatial relations between the pixels while keeping the color distribution unchanged. Replacing all pixels by the average color collapses this distribution onto its mean value. Both variants erase most of the semantic information that the original patch carried, see Fig. 5 for an example of both. This allows us to disentangle the effect of edge artifacts and of semantic concepts. The statistical analysis is performed as in Sec. 4.3 above and the results are shown in the last two columns of Tab. 2. We find the mean and median numbers of successful patch insertions to drop significantly—even if compared to patches from $\mathcal{P}_{\text{random}}$. The outcomes provide evidence for a "base" effect that can be attributed to edges and further artifacts since both ablations lead to a small number of successful patch insertions. This base effect, however, is minor compared to the effect of semantically intact patch shortcuts, thus, stressing the semantic meaningfulness and effectiveness of our approach.

5. Conclusion

We introduced an approach to identify learned shortcuts of a black-box network by analyzing a white-box proxy network, namely an interpretable-by-design BagNet that builds on local feature statistics. The patch shortcut candidates extracted via BagNet are transferred to and tested on the black-box model. The empirical evaluation on the binary-classification A2D2 dataset demonstrated the efficacy of our approach. Detected BagNet patch shortcuts, if tested on the black box, lead to significantly more misclassifications of the considered black-box network than, e.g., random insertions. Hence, they enable us to efficiently find vulnerabilities of the black box. An ablation study demonstrated that only a smaller fraction of the observed effects can be explained by edge artifacts. Most of it can be attributed to the semantics of the patch.

The employed coupling between BagNet and black box is relatively loose: Both networks are trained for the same task on the same dataset but, apart from this, share no information. We therefore expect more direct couplings, e.g. teacher-student approaches [25], to show even higher transferability rates. Further investigating how shortcut transfer depends on the chosen coupling technique seems promising.

Employing BagNet, our approach leverages the "locality" guarantees provided by this specific interpretable-by-design model. However, there are other classes of interpretable models such as invertible architectures, e.g. iRevNet [30], providing different guarantees and thereby offering alternative means for model assessment. It might further be possible to lift the need for fine-grained meta-annotations (in our case the pixel-wise semantic "car" or "no-car" information): Clustering the image patches that are crucial for BagNet decisions would enable a human-in-the-loop to readily detect predominant shortcut concepts.

Transferring the approach of a patch-based proxy model from images to other types of unstructured data, e.g. audio, video or text seems feasible. The concept of "image patch" then translates to short audio snippet or frequency band, volumetric cube or chunk of words.

Having instantiated our approach on a toy task, we emphasize that shortcut learning is by no means limited to such simple setups. It is a problem of more general nature [17] and shortcuts are an "ubiquitous" property of ML, affecting it for tasks of various complexity. As part of future work, one could extend the analysis to other datasets, more complex tasks and various kinds of black-box models. A systematic analysis of learned shortcuts, as made possible with the presented approach, contributes to safe ML by an early discovery of potential weak spots and failures. Further on, such insight opens the possibility for mitigation. Similar to e.g. adversarial training, shortcut patches could be used to augment and robustify training procedures. Using the image augmentation presented here not for testing but to generate new images, one could increase the prevalence of identified shortcuts within the dataset. To avoid misclassification, a network trained on this enhanced data would have to be more robust with respect to those shortcuts. Furthermore, a *shortcut exploitation score* reflecting the vulnerability of a given DNN could be used as secondary metric for model comparison.

Acknowledgments The research of J.R., M.M. and M.A. was funded by the German Federal Ministry for Economic Affairs and Energy within the project "KI Absicherung – Safe AI for Automated Driving". Said authors would like to thank the consortium for the successful cooperation. The work of J.S. was supported by the Fraunhofer Center for Machine Learning within the Fraunhofer Cluster for Cognitive Internet Technologies. The research of S.H. was funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01IS18038B).

References

[1] A. M. Alaa and M. van der Schaar. Demystifying Black-box Models with Symbolic Metamodels. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[2] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square Attack: A Query-Efficient Black-Box Adversarial Attack via Random Search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–501. Springer, 2020.

[3] J. Behrmann, W. Grathwohl, R. T. Q. Chen, D. Duvenaud, and J.-H. Jacobsen. Invertible Residual Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 573–582, 2019.

[4] A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth. Unrestricted Adversarial Examples via Semantic Manipulation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[5] W. Brendel and M. Bethge. Approximating CNNs with Bag-of-local-Features models works surprisingly well on Imagenet. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[6] T. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial Patch. *arXiv preprint arXiv:1712.09665*, 2017.

[7] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[8] T. Combey, A. Loison, M. Faucher, and H. Hajri. Probabilistic Jacobian-Based Saliency Maps Attacks. *Machine Learning and Knowledge Extraction*, 2(4):558–578, 2020.

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

[10] A. D'Amour, K. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman, F. Hormozdiari, N. Houlsby, S. Hou, G. Jerfel, A. Karthikesalingam, M. Lucic, Y. Ma, C. McLean, D. Mincu, A. Mitani, A. Montanari, Z. Nado, V. Natarajan, C. Nielson, T. F. Osborne, R. Raman, K. Ramasamy, R. Sayres, J. Schrouff, M. Seneviratne, S. Sequeira, H. Suresh, V. Veitch, M. Vladymyrov, X. Wang, K. Webster, S. Yadlowsky, T. Yun, X. Zhai, and D. Sculley. Underspecification Presents Challenges for Credibility in Modern Machine Learning. *arXiv preprint arXiv:2011.03395*, 2020.

[11] A. J. DeGrave, J. D. Janizek, and S.-I. Lee. AI for radiographic COVID-19 detection selects shortcuts over signal. *MedRxiv*, 2020.

[12] T. Devries and G. W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv preprint arXiv:1708.04552*, abs/1708.04552, 2017.

[13] A. A. Efros and W. T. Freeman. Image Quilting for Texture Synthesis and Transfer. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 341—346, 2001.

[14] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust Physical-World Attacks on Deep Learning Visual Classification. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] N. Frosst and G. E. Hinton. Distilling a Neural Network Into a Soft Decision Tree. *arXiv preprint arXiv:1711.09784*, 2017.

[16] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar. Born Again Neural Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 80, pages 1607–1616, 2018.

[17] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Learning*, pages 665––673, 2020.

[18] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth. A2D2: Audi Autonomous Driving Dataset. *arXiv preprint arXiv:2004.06320*, 2020.

[19] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[20] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018.

[21] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51(5), 2018.

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

[23] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. Augmix: A Simple Method to Improve Robustness and Uncertainty under Data Shift. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[24] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural Adversarial Examples. *arXiv preprint arXiv:1907.07174*, 2021.

[25] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. In *Advances in Neural Information Processing (NeurIPS) Deep Learning and Representation Learning Workshop*, 2015.

[26] H. Hosseini and R. Poovendran. Semantic Adversarial Examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[27] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial Examples Are Not Bugs, They Are Features. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[28] S. Jabbour, D. Fouhey, E. Kazerooni, M. W. Sjoding, and J. Wiens. Deep Learning Applied to Chest X-Rays: Exploiting and Preventing Shortcuts. In *Machine Learning for Healthcare Conference*, volume 126, pages 750–782, 2020.

[29] J.-H. Jacobsen, J. Behrmann, R. Zemel, and M. Bethge. Excessive Invariance Causes Adversarial Vulnerability. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[30] J.-H. Jacobsen, A. W.M. Smeulders, and E. Oyallon. i-Revnet: Deep Invertible Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[31] O. S. Kayhan and J. C. van Gemert. On Translation Invariance in CNNs: Convolutional Layers Can Exploit Absolute Spatial Location. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[32] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2017.

[33] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking Clever Hans Predictors and Assessing What Machines Really Learn. *Nature Communications*, 10, 2019.

[34] M. Lee and J. Z. Kolter. On Physical Adversarial Patches for Object Detection. *arXiv preprint arXiv:1906.11897*, abs/1906.11897, 2019.

[35] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2014.

[36] X. Liu, H. Yang, Z. Liu, L. Song, Y. Chen, and H. Li. DPATCH: An Adversarial Patch Attack on Object Detectors. In H. Espinoza, S. Ó hÉigeartaigh, X. Huang, J. Hernández-Orallo, and M. Castillo-Effen, editors, *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019*, volume 2301 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[37] S. Mohseni, A. Jagadeesh, and Z. Wang. Predicting Model Failure using Saliency Maps in Autonomous Driving Systems. *arXiv preprint arXiv:1905.07679*, abs/1905.07679, 2019.

[38] M. Naseer, S. H Khan, H. Khan, F. S. Khan, and F. Porikli. Cross-Domain Transferability of Adversarial Perturbations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[39] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the Asia Conference on Computer and Communications Security (ASIACCS)*, pages 506—-519. Association for Computing Machinery, 2017.

[40] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The Limitations of Deep Learning in Adversarial Settings. In *Proceedings of the European Symposium on Security and Privacy (EuroSP)*, pages 372–387, 2016.

[41] N. Papernot, P. D. McDaniel, and I. J. Goodfellow. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *arXiv preprint arXiv:1605.07277*, abs/1605.07277, 2016.

[42] V. Petsiuk, A. Das, and K. Saenko. Rise: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[43] M. T. Ribeiro, S. Singh, and C. Guestrin. ”Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, page 1135–1144, 2016.

[44] A. Rosenfeld, R. S. Zemel, and J. K. Tsotsos. The Elephant in the Room. *arXiv preprint arXiv:1808.03305*, abs/1808.03305, 2018.

[45] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[46] R. Shetty, B. Schiele, and M. Fritz. Not Using the Car to See the Sidewalk – Quantifying and Controlling the Effects of Context in Classification and Segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[47] S. Tan, R. Caruana, G. Hooker, and Y. Lou. Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation. In *Proceedings of the Conference on AI, Ethics, and Society*, page 303–310, 2018.

[48] S. Thys, W. Van Ranst, and T. Goedeme. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[49] R. Wiyatno and A. Xu. Maximal Jacobian-based Saliency Map Attack. *arXiv preprint arXiv:1808.07945*, abs/1808.07945, 2018.

[50] C. Yang, A. Kortylewski, C. Xie, Y. Cao, and A. Yuille. PatchAttack: A Black-Box Texture-Based Attack with Reinforcement Learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 681–698. Springer International Publishing, 2020.

[51] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.