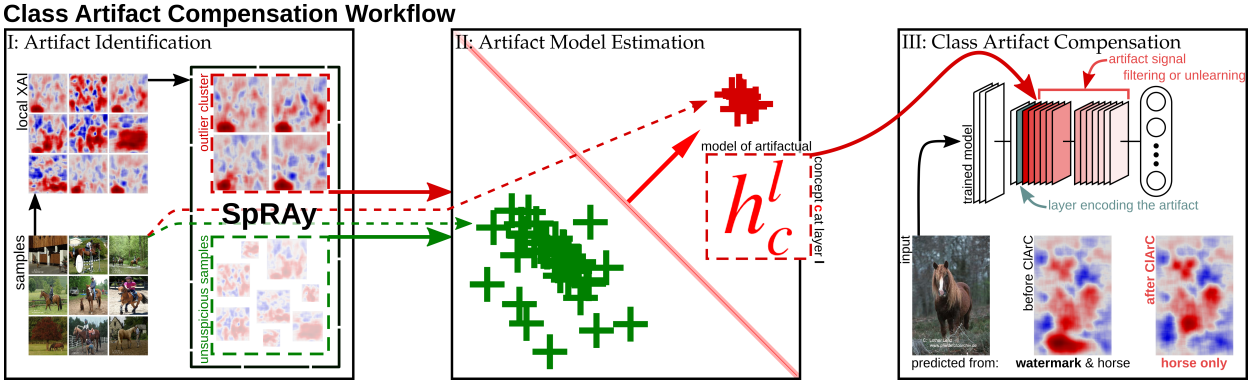
Graphical Abstract

Finding and Removing Clever Hans:
Using Explanation Methods to Debug and Improve Deep Models

Christopher J. Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, Sebastian Lapuschkin

Class Artifact Compensation Workflow



Highlights

Finding and Removing Clever Hans:
Using Explanation Methods to Debug and Improve Deep Models

Christopher J. Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, Sebastian Lapuschkin

- We extend Spectral Relevance Analysis for large scale analyses of Clever Hans traits
- We introduce Class Artifact Compensation to unlearn or suppress confounders
- We show that Deep Neural Networks tend to overfit to Clever Hans artifacts in data
- Using both methods, we discover and unlearn Clever Hans artifacts on modern datasets
- We demonstrate that Class Artifact Compensation leads to more trustworthy models

Finding and Removing Clever Hans: Using Explanation Methods to Debug and Improve Deep Models

Christopher J. Anders[a,b,**], Leander Weber[c,d,**], David Neumann[d], Wojciech Samek[d,b,*], Klaus-Robert Müller[a,b,e,f,*], Sebastian Lapuschkin[d,*]

[a]*Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany*
[b]*BIFOLD – Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*
[c]*Media Technology Group, Technische Universität Berlin, 10587 Berlin, Germany*
[d]*Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany*
[e]*Department of Artificial Intelligence, Korea University, Seoul 136-713, Korea*
[f]*Max Planck Institut für Informatik, 66123 Saarbrücken, Germany*

Abstract

Contemporary learning models for computer vision are typically trained on very large (benchmark) datasets with millions of samples. These may, however, contain biases, artifacts, or errors that have gone unnoticed and are exploitable by the model. In the worst case, the trained model does not learn a valid and generalizable strategy to solve the problem it was trained for, and becomes a "Clever Hans" predictor that bases its decisions on spurious correlations in the training data, potentially yielding an unrepresentative or unfair, and possibly even hazardous predictor. In this paper, we contribute by providing a comprehensive analysis framework based on a scalable statistical analysis of attributions from explanation methods for large data corpora. Based on a recent technique – Spectral Relevance Analysis – we propose the following technical contributions and resulting findings: (a) a scalable quantification of artifactual and poisoned classes where the machine learning models under study exhibit Clever Hans behavior, (b) several approaches we collectively denote as Class Artifact Compensation, which are able to effectively and significantly reduce a model's Clever Hans behavior, i.e., we are able to *un-Hans* models trained on (poisoned) datasets, such as the popular ImageNet data corpus. We demonstrate that Class Artifact Compensation, defined in a simple theoretical framework, may be implemented as part of a Neural Network's training or fine-tuning process, or in a post-hoc manner by injecting additional layers, preventing any further propagation of undesired Clever Hans features, into the network architecture. Using our proposed methods, we provide qualitative and quantitative analyses of the biases and artifacts in, e.g., the ImageNet dataset, the Adience benchmark dataset of unfiltered faces and the ISIC 2019 skin lesion analysis dataset. We demonstrate that these insights can give rise to improved, more representative and fairer models operating on implicitly cleaned data corpora.

Keywords: Deep Neural Networks, Explainable Artificial Intelligence, Clever Hans Predictors, Feature Unlearning, Spectral Relevance Analysis, Class Artifact Compensation

1. Introduction

Throughout the last decade, Deep Neural Networks (DNNs) have enabled impressive performance leaps in a wide range of domains, from solving classification problems [1, 2], over playing and winning games competitively [3, 4] (some in real time [5, 6]), to enabling the understanding of quantum-chemical many-body systems [7] and finding improved solutions to the notoriously difficult task of protein structure prediction [8]. These models are typically

*corresponding author
**contributed equally
Email addresses: anders@tu-berlin.de (Christopher J. Anders), leander.weber@hhi.fraunhofer.de (Leander Weber), david.neumann@hhi.fraunhofer.de (David Neumann), wojciech.samek@hhi.fraunhofer.de (Wojciech Samek), klaus-robert.mueller@tu-berlin.de (Klaus-Robert Müller), sebastian.lapuschkin@hhi.fraunhofer.de (Sebastian Lapuschkin)

(pre-)trained on very large datasets, e.g., ImageNet [9], with millions of samples. Recently, it was discovered that biases, spurious correlations, as well as errors in the training dataset [10] may have a detrimental effect on the training and/or result in "Clever Hans" predictors [11, 12], which only superficially solve the task they have been trained for, leading to potentially unfair and hazardous model behavior. Unfortunately, due to the immense size of today's datasets, a direct manual inspection and removal of artifactual samples can be regarded hopeless. Analyzing the biases and artifacts in the *model* instead may provide insights about the training data indirectly. This however requires an inspection of the learning models beyond black box mode.

Only recently methods of eXplainable Artificial Intelligence (XAI) (cf. [13, 14, 15] for an overview) were developed. They provide deeper insights into how a Machine Learning (ML) classifier arrives at its decisions and potentially help to unmask Clever-Hans predictors. XAI methods can be roughly categorized into two groups: methods providing *local* (e.g. [16, 17, 18, 19, 20, 21, 22, 23, 24]) explanations and those providing *global* (e.g. [25, 26, 27, 28]) explanations [29]. Current approaches are of limited use when scaling the search for biases, spurious correlations, and errors in the training dataset, as this would require intense "semantic" human labor. A recent technique, the Spectral Relevance Analysis (SpRAy) [12], aims to bridge the gap between local and global XAI approaches by introducing automation into the analysis of large sets of local explanations. The method however still involves a considerable amount of manual analyses, especially in context of contemporary datasets with high numbers of classes and samples such as ImageNet [9].

One of the main goals of ML is to learn accurate decision systems to automate tasks that otherwise may only be solved manually. As such, specific inference behavior on the available data often is expected from the learned models, e.g., within well-defined expert domains. As a recent body of research however has demonstrated, deviations from the anticipated are very likely (and must be expected) to appear in practice. In our paper, we propose a series of methods constituting a pipeline for the identification, description and suppression of those deviations in model inference, i.e., a set of tools to bring the model "back on track". Our goal is somewhat orthogonal to maximizing traditional generalization metrics such as test accuracy on benchmarks, since a high value for these metrics is often achieved *because* the model is able to exploit spurious correlations as shortcut strategies (this effect is extremely clearly observable in our experiments on the ISIC 2019 [30, 31, 32] skin lesion analysis dataset in Section 5.3). By discouraging or suppressing such strategies, we can therefore expect a decrease for traditional performance metrics, and instead obtain a more truthful estimation of a model's generalization ability.

In this paper,

(a) we introduce a novel framework that we collectively denote as Class Artifact Compensation (see Section 2.6) to enable large-scale analyses of a model's inference behavior on datasets with hundreds of classes and millions of samples for a semi-automated discovery of undesirable Clever-Hans effects that are embedded into data and model; here we rely on an extension of SpRAy (see Section 2.5), which increases the automation potential on such large datasets.

(b) In addition, we provide an intuition for Clever Hans artifacts and the desensitization of a trained model to their influence (see Sections 2.6) and demonstrate that Clever Hans artifacts are significantly more difficult to detect than Backdoors (see Section 3.1).

(c) In this manner, Class Artifact Compensation provides a well-controlled quantitative strategy to detect (Figure 1 *(Ic)*, Section 2.5), model and validate (Figure 1 *(II)*, Sections 2.2 and 2.6), and consequently remove the influence of such artifacts from the model (Figure 1 *(III)*, Section 2.6).

(d) We showcase the steps of our approach on a modified MNIST [33, 34] dataset with color-based Clever Hans information (see Sections 3.2, 4.1 and 5.1), the ImageNet [1] dataset (see Sections 3.2, 3.3, 4.2, 4.3 and 5.2), the ISIC 2019 [30, 31, 32] skin lesion analysis dataset (see Section 5.3) and the challenging Adience [35] benchmark dataset of unfiltered faces (see Section 5.4).

(e) Finally, we discuss the intricacies of (informed intervention in) the decision-making of end-to-end learned predictors (see Section 6) and conclude with Section 7.

These extensive analyses allow interesting findings that are illuminating beyond our specific technical approach.

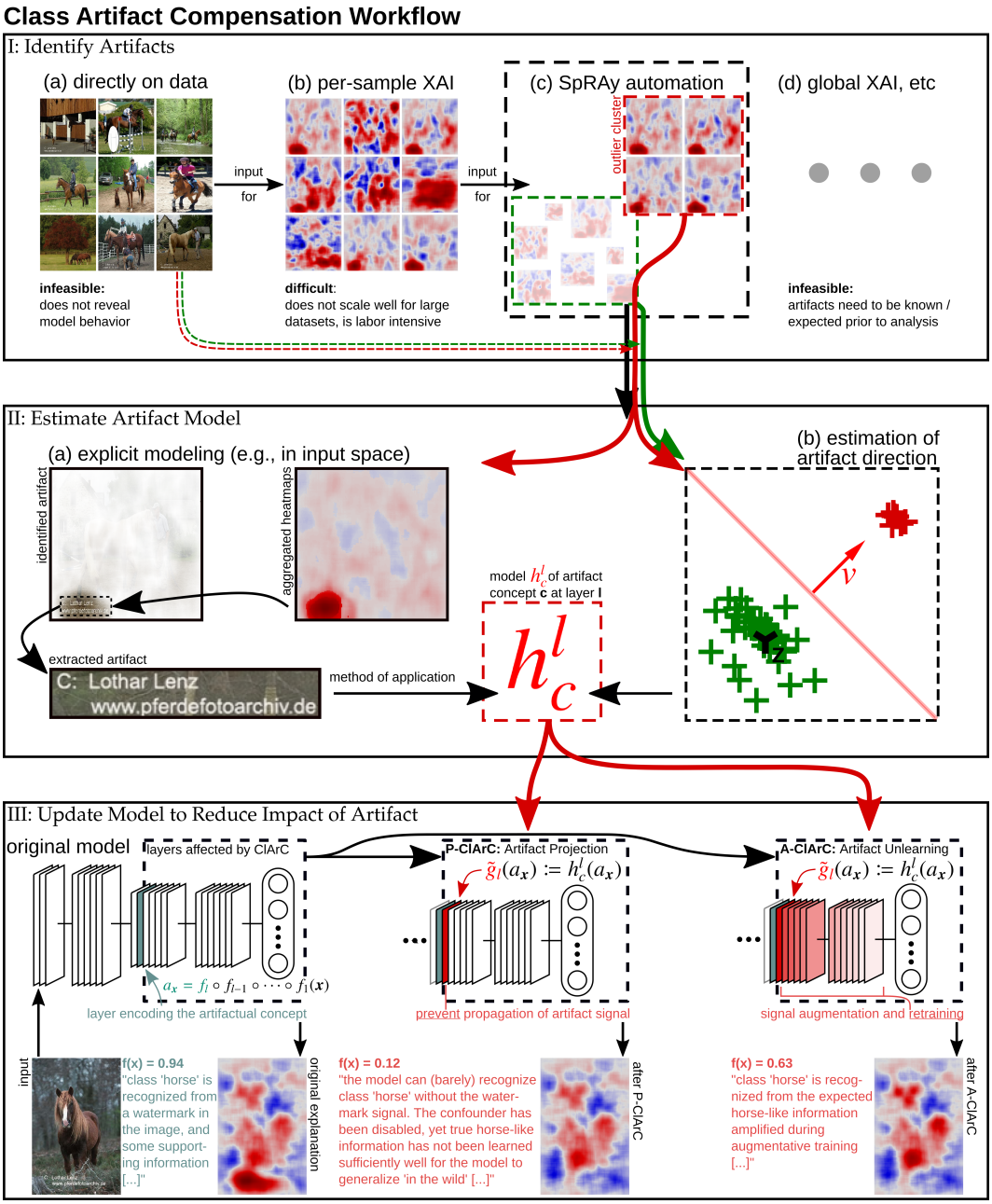


Figure 1: The workflow of our Class Artifact Compensation framework. *(I)* We first aim to identify spurious confounders in the data as learned by the model. *(Ia)* A direct analysis of the training data is infeasible due to the missing correspondence to the features used by the model during inference. *(Ib)* Explanations from local XAI methods may provide this information. However, manual analysis requires the evaluation of extreme amounts of explanations (per class). *(Ic)* We therefore propose an automation of this process, based on an extension of the SpRAy [12] algorithm. *(Id)* While the application of globally operating XAI techniques is disqualified in the identification phase, as here the concepts evaluated after must be known beforehand, *(II)* these techniques find application in the modeling of an artifact estimator in our approach: *(IIa)* While an artifact model can be built explicitly after identification, e.g., from expert domain knowledge, *(IIb)* it can also be learned from representative data, e.g., as CAVs. *(III)* With a known model of the artifact at the layer of its most distinct representation within the DNN, one can attempt to remove its influence on the network. To this end, we present the following two approaches: *P-ClArC* aims at the selective deactivation of the artifact signal, and, as a largely training-free approach, leaves the remainder of the model unaltered. *A-ClArC* on the other hand strategically augments the training data (of all classes) with the artifact signal in order to minimize its class-specific informative value, to force the model to adapt to other (benign) features in continued training.

1.1. Related Work

There is an increased awareness that ML models need to be interpretable to its users in order to assess the validity of the decision making of the predictor [36, 37], especially in high risk settings, such as in medical applications [38, 39, 40, 41, 42, 43]. Transparency in model predictions could point at anomalous or blundering decision behavior before harm is caused in a later usage as a diagnostic tool. Consequently, numerous approaches to understand aspects of state of the art Artificial Intelligence (AI) predictors have been developed in recent years (cf. [14, 15] for an overview) in the emerging field of eXplainable Artificial Intelligence (XAI). In the following paragraphs, we will discuss related work by introducing relevant research work and terminology from the field of XAI important to this paper.

The Clever Hans Effect. Clever Hans (CH) was a horse from Berlin, Germany, that allegedly was able to do math – a media sensation from the early 1900s. Later in 1907 it was discovered that Hans would read the examinator's body language instead of performing arithmetics, and in this manner give the right answer but for the wrong reason[1] [11]. "Clever Hans Strategies" or "Clever Hans Effects" for ML predictors [12, 44] are accordingly named as a homage to this infamous horse, and describe a prediction making learned and executed based on biases and *spurious corellations* in the training data, instead of valid (i.e., intended or expected) features and relations.

As such, there is a notable distinction to make between the CH artifacts, Backdoor (BD) Attacks [45, 46] and attacks based on Adversarial Examples [47]. Adversarial attacks are specifically generated for individual data points in order to cause a misprediction, and are as a consequence ineffective when used on other samples. BD Attacks and CH artifacts on the other hand are systematically learned and exploited by the model. BDs are generally injected with (malicious) intent during training, into samples of multiple classes via added "trigger patterns" (e.g., a gray pixel at a specific location) while overriding the targeted samples' true training labels [45, 48]. BDs are usually not part of the original training data anymore once training is finished. CH type artifacts, however, are "naturally occurring" phenomena in the training data corpus correlating with only single (or few) ground truth labels, providing for shortcuts around more complex connections in the training data [49]. In contrast to Backdoor Attacks, which, if present, cause the model to override its prediction making on valid features, CH artifacts almost always appear alongside benign indicators for a class, and thus exert a significantly weaker influence on the model. Further, the decision whether a characteristic in the data is indeed a CH, or merely a benign feature, often is subject to the expectation of the model's behavior and expert domain knowledge [50, 12, 42]. They are consequently, in addition to their unexpected nature, more difficult to detect, as experimentally highlighted in Section 3.1. Other than BDs, CH artifacts are part of the features in some of the original training samples, and may thus be identified during a joint analysis of the available data and the model's utilization of it, as described throughout this paper. The particular difference between datasets with CH artifacts and datasets with BDs is illustrated in Figure 2. In literature, numerous CH strategies have been identified and collected[2], e.g., with the help of techniques from XAI, in a surprising number of current and former state-of-the-art ML models, in part invalidating their reported (benchmark) performance as a measure of generalization capability [51, 10, 12, 50, 42, 49, 52, 53].

Local XAI. XAI methods aim at providing transparency to the prediction making of ML models, e.g., for the validation of predictions for expert users, or the identification of failure modes. Local explanations provide interpretable feedback on *individual* predictions of the model, and assess the importance of input features wrt. *specific* samples. Local attributions are commonly presented in the form of heatmaps aligned to the input space, computed, e.g., with (modified) backpropagation approaches, such as sensitivity analysis [16, 54], Layer-wise Relevance Propagation (LRP) [17], Deep Taylor Decomposition [55], Grad-CAM [19], Integrated Gradients [20], SmoothGrad [56], and DeepLIFT [21], which require access to the internal parameters of DNN models. Surrogate- and sampling-based approaches, including LIME [22], Prediction Difference Analysis [23] and Meaningful Perturbations [24] view the model as an impenetrable black box and derive local explanations via proxy models and data, at the cost of increased runtime and an approximative nature of the obtained results. Occlusion analysis [18] follows a similar principle by measuring the effect of the removal or perturbation of input features from samples at the model output. Shapley value based approaches [57, 58] leverage tools from game theory in order to estimate the importance of features to a decision of a model.

[1] `https://en.wikipedia.org/wiki/Clever_Hans`
[2] `http://tinyurl.com/specification-gaming`

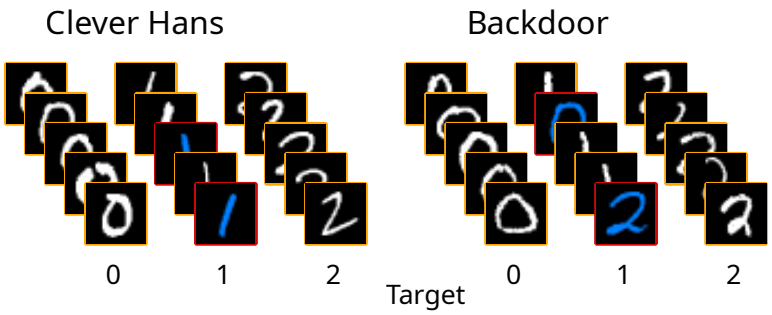


Figure 2: Difference between datasets with Clever Hans *(left)*) and Backdoor *(right)* artifacts visualized for colored MNIST. The artifact feature that separates afflicted samples (red frame) from unaffected ones (yellow frame) here is for both types of artifacts the color blue (different from the standard color *white*). In the case of CH artifacts, the artifact feature will only ever appear in samples alongside features for a single class. For BD attacks, the artifact feature appears in samples among features for all (other) classes except the target class, making the artifact the only discriminative feature in affected samples distinctive for its target class.

Global XAI. Global methods aim at obtaining a general understanding about a model's sensitivities, learned features and concept encodings. Some approaches operate by assessing the general importance of predetermined features, concepts or data transformations by systematically evaluating the model's reaction to varying exposure thereto, using (larger) sets of real or artificially generated samples [25, 26, 27, 28]. Other approaches aim at understanding predictors by identifying important neurons and their interactions [59], and visualizing learned feature encodings by e.g., synthesizing preferred inputs to hidden filters within a neural network model, e.g. [60, 61, 62, 63].

Bridging the Gap. Both the local and global approaches to XAI suffer from a (human) investigator bias during analysis and thus are on their own of only limited use for searching and exploring for biases, spurious correlations and errors learned by the model from the training data. Global methods can only measure the impact of pre-determined, expected or a priori known features or effects (cf. [27, 28]), which limits their applicability when aiming for the discovery of *yet unknown* behavioral facets of a model. Local methods, on the other hand, have the potential to provide much more detailed information *per sample*, but the task of compiling information about model behavior over thousands (or even millions) of samples and explanations is tiring and laborious for a human investigator: The success of such an analysis depends on the examiner's keen perception and domain knowledge, limiting the potential for knowledge discovery about model behavior.

A recent technique, the Spectral Relevance Analysis (SpRAy) [12], aims at bridging the gap between local and global XAI approaches, by introducing automation into the analysis of large sets of local explanations. SpRAy has been applied in a recent set of works, e.g., [50, 12], which however mainly operate on smaller datasets, each containing only hundreds of samples each. The procedure described in [12] however still involves a non-negligible amount of manual analysis, especially in context of contemporary datasets with high numbers of classes and samples, such as ImageNet [9]. In our work, we purposefully extend the SpRAy technique and bring it to scale for robustly analyzing extensive datasets, in Section 2.5.

Feature Unlearning. The awareness of CH predictors has invigorated research with the intent to improve models, by unlearning unwanted inference patterns. A most naive approach to unlearn a concept that can be found in a subset of samples in the training set is to remove those samples altogether, and to retrain the model from scratch on the reduced training set. While this approach is straight forward and easy to implement, it comes at the cost of also removing desirable features the model could positively benefit from, along with the characteristics in the data deemed problematic. This may be especially harmful if there are only few training data available to begin with. Furthermore, in some cases the initial model training may have been extremely costly, and an approach to fine-tune the model instead would be more desirable.

Several approaches have thus been developed to unlearn unwanted predictive behavior from existing models [64, 65, 66] or to guide the model during training by providing information about the expected explanations [64, 67, 50]. eXplanatory Interactive Learning (XIL) [65, 50] presents local explanations to a human observer during training, who in turn provides feedback to the model by replicating samples affected by CH phenomena and replacing the

contained artifactual features with noise or otherwisely generated patterns. However, since explanation corrections need to be performed by a human expert during learning, repeatedly and in a sample-wise fashion, and the model needs to wait on each correction, which may be infeasible especially for large data corpora, thus, XIL is relatively inefficient in terms of runtime for practical applications. The work of Kim et al. [66] introduces Learning not to Learn (LNTL), a model regularization scheme, in which an additional "artifact detector" learning specific biasing features is attached to the original predictor. The original model is then driven to minimize the shared information with the dedicated bias predictor, and thus to unlearn to use artifactual features for inference. For this purpose, additional annotations are required for each sample in the dataset, indicating the presence of biases. In practice, however, these bias labels may be quite time-consuming to obtain, since a human interpreter is required. Ross et al. [67] aim to guide the model towards the correct behavior by penalizing high attribution scores in undesired regions by extending the optimization function with a "Right for the Right Reasons (RRR)" loss term. Similarly, Rieger et al. [64] propose Contextual Decomposition Explanation Penalization (CDEP), a method for regularizing model behavior based on explanations obtained from Contextual Decomposition (CD) [68], by complementing the classification error of the loss function with an explanation error term. Both RRR and CDEP require sample-wise ground-truth explanations for whole datasets, such as (binary) importance masks, which are costly to obtain, since they need to be provided by a human expert, and may thus not be feasible in a practical setting. Furthermore, recent work has shown that models can be manipulated in such a way that produced attribution maps may be arbitrary, while the prediction of the model is unchanged [69]. Regardless of the attribution method, as long as the gradient is involved, optimizing for a target attribution will result in an exploitation of the gradient directions orthogonal to the embedded data manifold. Consequently, there is no guarantee that in general unlearning approaches based on extensions of the loss function effectively correct the model's use of the input features.

2. Methods

In this section, we introduce the various methods which are part of the proposed Class Artifact Compensation framework, or are otherwise used in our analyses throughout this paper.

2.1. Spectral Signature

For the detection of BD-type artifacts used by DNNs, Tran et al. [46] propose the Spectral Signature (SpeSig) method. Given some dataset X that is poisoned with a BD and a model f trained on this data, let $X_y = \{x_1, \ldots, x_n\}$ be the subset of samples corresponding to a target label y. Tran et al. [46] apply the following method separately for all y in the dataset, since the aim is to identify all (previously unknown) BD samples within X: For each sample x_i, the model f provides a feature representation $a(x_i)$. From these representations, one computes the covariance matrix

$$S = \frac{1}{n^2} \sum_{i=0}^{n} \sum_{j=0}^{n} (a(x_i) - \bar{a})(a(x_j) - \bar{a})^\top, \tag{1}$$

where $\bar{a} = \frac{1}{n} \sum_{i=1}^{n} a(x_i)$ and $n = |X_y|$. For each sample, an outlier score τ is then computed using the top right singular vector v of S:

$$\tau_i = ((a(x_i) - \bar{a}) \cdot v))^2 \tag{2}$$

Samples with a high τ_i are more likely to be outliers, allowing for the k samples with the largest τ_i to be detected as poisoned. Note that since SpeSig detects outliers wrt. samples of one class label y, the found BDs are usually images that originally belonged to other classes – and thus do not fit into the manifold of X_y. More concisely, SpeSig does not detect the poisoned artifact itself, but the "odd" samples within X_y. Tran et al. [46] then propose to remove the detected outliers and retrain the model, thereby defending against the BD attack. In Section 3.1, we apply the SpeSig method not only to identify BDs, but also on a dataset containing CH artifacts to assert their conceptual differences.

2.2. Concept Activation Vectors

Kim et al. [27] introduce CAVs as a means to provide an interpretation of a DNNs internal state in terms of human-understandable concepts. Given two sets of samples X^+ and X^-, where the samples in X^+ all exhibit a specific

property c (e.g. X^+ contains images showing striped objects) which is not present in X^-, a CAV is trained as a linear classifier separating the hidden representations of the samples from X^+ and X^- at some layer l within the DNN. The thus learned weight vector v_c^l then represents the direction in latent space encoding the concept c unique to X^+.

Kim et al. [27] use CAVs as directional derivatives in order to test the sensitivities of neural network models wrt. to a priori known concepts, which the authors refer to as *testing* with CAVs, or TCAV. While we do not incorporate the same *testing* approach, we use CAVs similar to [27] as a means to verify the sensitivity of the model to the CH artifacts, e.g., those identified via SpRAy, as shown for example in Section 5. Further, we use CAV directions specific to CH effects in context of the Class Artifact Compensation (ClArC) unlearning framework, as a means to remove specific behavioral facets from the DNN's inference process. Note however that when there is only a limited number of samples available to compute a CAV, its direction may not be orthogonal to other concepts common to the used samples, and therefore may encode additional concepts.

2.3. Layer-wise Relevance Propagation

Layer-wise Relevance Propagation LRP [17] is a local XAI approach reversely iterating over the layered structure of a neural network to produce an explanation. Consider the neural network

$$f(\boldsymbol{x}) = f_L \circ \cdots \circ f_1(\boldsymbol{x}) \,. \tag{3}$$

In a forward pass, activations are computed at each layer of the neural network. The activation score in the output layer forms the prediction, which is then backpropagated and redistributed, layer by layer, until the input is reached. The redistribution process follows a conservation principle analogous to Kirchoff's laws in electrical circuits, i.e., all relevance assigned to any neuron during the process of backpropagation will be further distributed towards its inputs in the layer below without loss.

Various propagation rules have been proposed in literature [17, 70, 71]. For example, the LRP-γ rule [70] defined as

$$R_{j\leftarrow k} = \frac{a_j(w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j(w_{jk} + \gamma w_{jk}^+)} R_k \,, \tag{4}$$

where a_j are the layer's input activation at the j^{th} neuron, w_{jk} the learned parameters mapping the j^{th} input activation to the k^{th} layer output and $w_{jk}^+ = \max(0, w_{jk})$ is the positive part of the learned weights. The $\sum_{0,j}$ runs over all lower-layer activations a_j, plus an extra neuron representing the bias. More precisely, we set $a_0 = 1$ and define w_{0k} to be the neuron bias. The variable $\gamma \geq 0$ is a free parameter to tune the decomposition rule. Equation (4) redistributes R_k based on the contribution of lower-layer neurons to the given neuron activation, with a preference for positive contributions over negative contributions. This makes it particularly robust and suitable for the lower-layer convolutions. Other propagation rules such as LRP-ε, LRP-$\alpha\beta$ or LRP-z^B, are suitable for other application scenarios and layer types [70, 71] and have been shown to work well in practice [72].

After the step of relevance decomposition, lower layer neuron relevance is aggregated from incoming relevance messages as $R_j = \sum_k R_{j\leftarrow k}$. For a technical overview of LRP including a discussion of the various propagation rules and further recent heuristics, see [70]. In all our experiments, we compute LRP attribution scores using LRP-ε (near the model output), LRP-γ (in intermediate layers) and LRP-z^B (near the input), as described in [15], using the *Zennit*[3] attribution [73] framework for PyTorch [74].

2.4. Dimensionality Reduction for Visualization

When working high dimensional data, the relationship between samples can be very hard to interpret. Images or attribution maps may be compared side-by-side, but their relations in pixel-, or some embedded space, can not be easily understood. One may acquire some insight by reducing the dimensionality of samples before comparing them. However, to incorporate local relations between points, a simple linear embedding with e.g., PCA may not be sufficient and nonlinear PCA may be required [75]. Locally-linear Embedding (LLE) [76] attempts to solve this

[3] https://github.com/chr5tphr/zennit

problem, by constructing a weight-matrix where samples are represented by a linear combination of their nearest neighbors. A neighborhood-preserving map into an arbitrarily dimensional space can then be computed by an eigen-decomposition of this weight matrix with an optimal number of eigen-vectors. LLE however suffers from a series of draw-backs [77, 78], which render it an infeasible choice for embedding data with outlier populations such as CH clusters.

Another more popular approach with high visual quality is t-distributed Stochastic Neighbor Embedding (t-SNE) [79], for which a probability distribution is modeled over the pairwise distances of samples, where close distances correspond to high probability. Then, the Kullback-Leibler divergence with a (usually) 2-dimensional target Student's t-distribution is minimized. Uniform Manifold Approximation and Projection (UMAP) [80] is a more recent approach which is comparable in visual quality to t-SNE, but provides lower computational cost. It is also based on pairwise similarities, and assumes the data is uniformly distributed on a Riemannian Manifold, which is locally connected, and where the Riemannian metric is constant.

2.5. Spectral Relevance Analysis

Spectral Relevance Analysis (SpRAy) [12] is a meta-analysis tool for finding patterns in model behavior, given sets of instance-based explanatory attribution maps. The SpRAy algorithm has its core in Spectral Clustering (SC) [81, 82] and – via the use of attribution maps as input – enables the analysis of the input data from the model's perspective for finding (hidden) characteristics of specific classes, which however are exploited by the model.

The SpRAy algorithm, as introduced in [12] initializes by computing the sparse affinity structure over the input attribution maps considering all pair-wise similarities between the given samples. A (normalized, symmetrical and) positive semi-definite graph laplacian L_{sym} [83, 12] is then computed from the affinity matrix A, and provided as input to SC (cf. [83]). As output, SpRAy yields a Spectral Embedding Φ of the input attributions and the corresponding spectrum of eigenvalues $\Lambda = \{\lambda_i\}_{i=1\ldots q}$. Lapuschkin et al. [12] follow [83] and (manually) read the structure (i.e., number and nesting) of clusters from the eigenvalue spectrum Λ, via the spectral- or eigen-gap [83], e.g., for ranking a set of analyzed classes wrt. to their potential for exhibiting CH phenomena [12]. For further visual analysis, the affinity matrix A is then used together with a suitable number of cluster labels inferred from Λ as a basis for an embedding into $\mathbb{R}^2$, e.g., by using t-SNE [79]. Figure 3 provides an overview of the procedure outlined above, where arrows and symbols in black and gray color describe the workflow of SpRAy from [12], and arrows and symbols in red color distinguish our own extensions and adaptations of the algorithm described below.

Spectral Relevance Analysis Brought to Scale. We extend the SpRAy algorithm by drawing proper utility from the Spectral Embedding Φ, an intermediate result of the SC algorithm, which so far has remained unused in [12].

While the $q \leq n$ most significant eigenvectors of the singular value decomposition on the graph laplacian L_{sym} constitute the columns of the $(n \times q)$ shaped Spectral Embedding Φ, each of the matrix' rows corresponds to exactly one of the n input attribution maps. We therefore use the rows of Φ (instead of A) as an input to mapping and embedding algorithms. For a broader analysis, in addition to t-SNE [79], we use UMAP [80] for projecting the spectral analysis results (instead of the preprocessed data representation A) into a two-dimensional vector space $\mathbb{R}^2$ for further visual inspection. Note that the final algorithmic step of SC is the assignment of cluster labels to input samples. For this purpose, one usually applies any other suitable clustering algorithm (e.g. k-Means [84] or DBSCAN [85]) on top of the data represented by the already well-structured embeddings in Φ. The use of Φ as a source for computing embeddings in $\mathbb{R}^2$ thus leads to a close correspondence of the visualized cluster groupings to the assigned cluster labels.

A critical decision in clustering approaches is the number of desired clusters. While for small datasets like Pascal VOC [86] it suffices to analyze the per-class eigen-spectrum [12]; datasets with a large number of classes cannot be feasibly analyzed by manual comparison and ranking of the eigen-spectra of all classes to identify those exhibiting spurious model behavior. In order to further automate this process, we propose Fisher Discriminant Analysis (FDA) to rank all class-wise clusterings by their respective (linear) separability measured in terms of units of the quantity τ. FDA [87, 88] is a widely popular method for classification as well as class- (or cluster-) structure preserving dimensionality reduction. FDA finds an embedding space by maximizing between-class scatter $S^{(b)}$ and minimizing

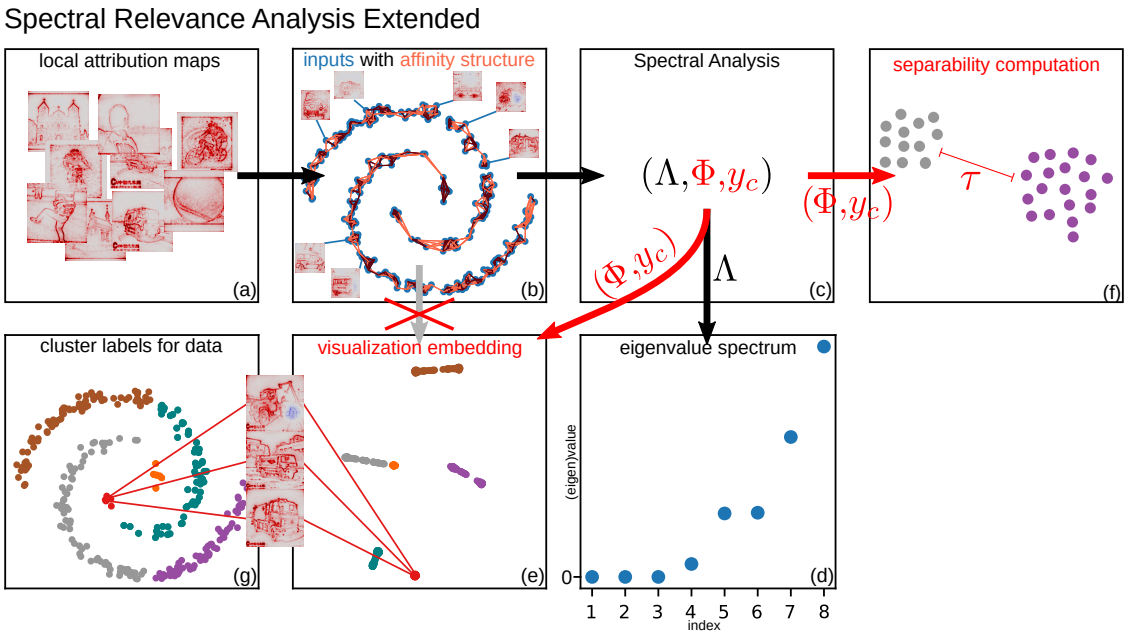


Figure 3: Our extended SpRAy algorithm. ***(Black** and gray paths)*: steps followed by the SpRAy procedure as defined in [12]. ***(Red** paths)*: our extensions and changes to the SpRAy algorithm to increase the automation potential and applicability to very large datasets. *(a)* From a set of local attribution maps, a sparse affinity matrix is computed in *(b)*. *(c)* The affinity data is then passed as input for analysis with SC [81, 82] in the form of a positive semi-definite graph laplacian, resulting in a spectrum of eigenvalues Λ, the Spectral Embedding Φ corresponding to the input data (see *(e)* and *(g)*), as well as sets of proposed cluster labels y_c. *(d)* Lapuschkin et al. [12] perform to a large extent direct manual analyses on the eigenvalue spectrum Λ, within and between analyzed classes, for the identification of CH behavior and distinct cluster groupings, and embed the sparse affinity structure of the data given the estimated cluster labels y_c for visualization. Instead of using the sparse affinity matrix from *(b)*, our extensions rely on the already expressive Spectral Embedding Φ (together with cluster labels y_c) for *(e)* visualizing the analyzed data groupings, *(f)* and the automation and quantification of rating clusters and classes for "Clever Hans'ness" τ, via the computation of separability scores, from, e.g., FDA.

within-class scatter $S^{(w)}$, given by

$$S^{(w)} = \sum_{k=1}^{K} \sum_{x_i \in \mathbf{c}_k^K} (x_i - \mu_k)(x_i - \mu_k)^\top \tag{5}$$

$$S^{(b)} = \sum_{k=1}^{K} (\mu_k - \mu)(\mu_k - \mu)^\top \,. \tag{6}$$

Here, $\mathbf{C}^K$ is a clustering with K clusters $\mathbf{c}_k^K$ with $k \in \{1, \ldots, K\}$, μ_k the sample mean of cluster k and μ the mean over the whole set of samples. The solution of FDA can be understood as directions of maximal separability between clusterings, and, when normalized and plugged into the original objective, gives scores of separability $R(\mathbf{C}^K)$. In our specific use-case, for each class we compute separability scores $R(\mathbf{C}^K)$ on the Spectral Embedding Φ (using the concept of an empirical kernel map [89]) and each clustering $\mathbf{C}^K$ in a set of clusterings $\mathbb{K} = \{\mathbf{C}^K\}$. We then define the *class*-separability score as

$$\tau = \frac{1}{|\mathbb{K}|} \sum_{\mathbf{C}^K \in \mathbb{K}} R(\mathbf{C}^K) \,, \tag{7}$$

which may then be used to compare classes wrt. their "Clever Hans'ness". In the SpRAy setting, large τ denote outlierness in the predictor's attribution – as indicators for artifact candidates – whereas low τ does not indicate any strikingly "irregular" prediction behavior. Clearly any algorithmic alternatives quantifying the separability of two or more sets of labelled samples may be used as an alternative to compute τ, although we see FDA as one of the more intuitive approaches.

Note that the decision whether some feature is a CH or a valid feature requires external (human expert) knowledge, may be subjective in some cases, and can therefore not be answered using only the information provided from within the confines of datasets that offer no ground truth for reasoning. While our extended SpRAy increases the degree of automation of the method, human effort is still required in order to complete the analysis, in form of (subjective) decisions of whether the discovered model strategies constitute CHs, or whether they are benign and expected model behavior. This is especially true when ground truth annotations (e.g., labels) of the data are insufficiently expressive. With SpRAy the need for human involvement is reduced to singular decisions wrt. suggested candidates.

Algorithm 1 provides a complete algorithmic description of the extended SpRAy technique, while the red arrows and symbols in Figure 3 distinguish our approach from SpRAy in [12]. We provide an implementation of our approach as part of the *CoRelAy*[4] framework [73] for python.

Algorithm 1: Spectral Relevance Analysis Extended

Data: Class of interest y,
Data set $X = \{x_1, x_2, ..., x_n\}$
Model f operating on X and predicting y
Result: Eigenvalues $\Lambda = \{\lambda\}$,
Spectral embeddings $\Phi \in \mathbb{R}^{n \times q}$,
Clusterings $\mathbb{K}$,
Mean separability score τ,
Visualization embeddings $V \in \mathbb{R}^2$

`/* compute attributions for` $x \in X$`, using, e.g., LRP */`

$R = \{\}$;
for $x \in X$ **do**
  $R_x = \text{attribution}(f, x, y)$;
  $R.\text{append}(R_x)$;
end

`/* Spectral Relevance Analysis */`

$\Phi, \Lambda, \mathbb{K} = \text{SpRAy}(R)$;

`/* Compute separability scores given by, e.g., FDA */`

for $\mathbf{C}^K \in \mathbb{K}$ **do**
  $S_{\mathbf{C}^K} = \text{separability}(\Phi, \mathbf{C}^K)$;
end

`/* compute mean separability score [Eq. (7)] */`

$\tau = \frac{1}{|\mathbb{K}|} \sum_{\mathbf{C}^\kappa \in \mathbb{K}} S_{\mathbf{C}^\kappa}$;

`/* compute embedding visualizations, with e.g., UMAP */`

$V = \text{visualize\_embedding}(\Phi)$;
return $\Lambda, \Phi, \mathbb{K}, \tau, V$

2.6. Class Artifact Compensation

Assume we have a set of atomic features $\mathbb{F}$. A concept $c \in 2^{\mathbb{F}}$ may be any combination of atomic features to describe an abstract property, where $2^{\mathbb{F}}$ is the power set of $\mathbb{F}$. We may define an M-tuple of concepts $C = (c_1, c_2, ..., c_M)$ with $c_i \in 2^{\mathbb{F}}$ for $i \in \{1, ..., M\}$. Given the superset of concepts $\mathbf{C} = \bigcup_{i=1}^{N} c_i$, assume a set of untangled data points that can be constructed by a combination of features used in any concept $\mathbb{D} = \{\bigcup_{\omega \in \Omega} \omega \mid \Omega \in 2^{\mathbf{C}}\}$. Each untangled data point $\alpha \in \mathbb{D}$ is a combination of atomic features $2^{\mathbb{F}}$ and therefore could be equivalent to a single concept. We may now, given α, construct a signal vector $s(\alpha) \in \{0, 1\}^M$ using

$$[s(\alpha)]_i = \delta_{c_i \subseteq \alpha} \qquad i \in \{1, 2, ..., M\} \tag{8}$$

[4] https://github.com/virelay/corelay

with the Kronecker Delta δ, where each entry at index i is 1 if $c_i \subseteq \alpha$. In other words, $s(\alpha)$ is a binary encoding of α given concepts C.

Now assume we have an N-tuple of *untangled* datapoints $D = (\alpha_1, \alpha_2, ..., \alpha_N)$ with $\alpha_i \in \mathbb{D}$ for $i \in \{1, ..., N\}$. We may now construct a corresponding N-tuple of *tangled* datapoints $X = (x_1, x_2, ..., x_N)$ based on D, where each sample x_i is a mixture of concepts given a pattern matrix $A : \mathbb{R}^{N \times M}$

$$x_i = As(\alpha_i) \qquad i \in \{1, 2, ..., N\}. \tag{9}$$

Suppose we call concept c_k at index $k \in \{1, 2, ..., M\}$ an *artifact*. A set of labels t_i that indicate whether a datapoint α contains the artifact c_k can then be defined as

$$t_i = \begin{cases} 0, c_k \not\subseteq \alpha_i \\ 1, c_k \subseteq \alpha_i \end{cases} . \tag{10}$$

Assuming we have a function $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ on the tangled datapoints X, there are two questions we seek answers for: (1) Is f sensitive to artifact c_k? (2) How can f be modified such that it is insensitive to artifact c_k?

Concept Sensitivity of Functions. To measure the sensitivity to artifact c_k with labels $t_i \in \{0, 1\}$, one needs to compare the behavior of function f on non-artifact samples $X^- = \{x_i \mid i \in \{1, 2, ..., N\} \wedge t_i = 0\}$ and artifact samples $X^+ = \{x_i \mid i \in \{1, 2, ..., N\} \wedge t_i = 1\}$. A naive approach may be for example to estimate the density of the artifact samples with a multivariate normal distribution with mean μ^+ and covariance Σ^+

$$\mu^+ = \frac{1}{|X^+|} \sum_{x^+ \in X^+} f(x^+) \qquad \text{and} \qquad \Sigma^+ = \frac{1}{|X^+|} \sum_{x^+ \in X^+} (f(x^+) - \mu^+)(f(x^+) - \mu^+)^\top \tag{11}$$

and verify whether or not it matches with a similar estimation for the non-artifact samples, where $|X^+|$ is the cardinality of X^+. This may however not produce good results when the number of samples is limited, since the distributions cannot be estimated accurately enough, or when the data cannot be modeled conclusively with a multivariate normal.

More generally, we can explicitly estimate an artifact model $h : \mathbb{R}^d \to \mathbb{R}^d$, which, given a non-artifact sample $x_i^- = As(\alpha_i)$ with $c_k \not\subseteq \alpha_i$ produces an artifact sample

$$h(x_i^-) \approx As(\alpha_i \cup c_k). \tag{12}$$

We can formulate the artifact model with the objective

$$\hat{\theta} = \arg\min_\theta \frac{1}{|X^-||X^+|} \sum_{x^- \in X^-} \sum_{x^+ \in X^+} \|h(x^-; \theta) - x^+\|^2 \tag{13}$$

where $\hat{\theta}$ are the optimal hyperparameters of h. The artifact estimator h is thus the function h with hyperparameters $\hat{\theta}$ that produces the minimal ℓ_2-distance between mapped non-artifact samples $h(x^-)$ with $x^- \in X^-$ and artifact samples x^+ with $x^+ \in X^+$. The sensitivity of function f to a concept c_k modeled with h may then be estimated using

$$S = \frac{1}{|X^-|} \sum_{x^- \in X^-} \|f(h(x^-; \hat{\theta})) - f(x^-)\|. \tag{14}$$

Intuitively, the addition of a concept may be more feasible to estimate than the removal. Take, for example, the introduction of an opaque watermark in an image. This operation is not invertible as we destroyed the pixel information under the watermark. While Equations (13) and (14) assume the transformation of a non-artifact sample to an artifact sample in an *inductive* artifact model, they may equivalently be formulated with a removal of the concept in a *suppressive* artifact model h_s with

$$\hat{\theta}_b = \arg\min_\theta \frac{1}{|X^-||X^+|} \sum_{x^- \in X^-} \sum_{x^+ \in X^+} \|x^- - h_s(x^+; \theta)\|^2 \tag{15}$$

The sensitivity of a function to a concept *suppressively* modeled by h_s may then be measured using

$$S_b = \frac{1}{|X^+|} \sum_{x^+ \in X^+} \|f(h_s(x^+; \hat{\theta}_b)) - f(x^+)\|. \tag{16}$$

Concept Desensitization. Depending on the type of function f, there may be multiple possible approaches to obtain a desensitized function $\tilde{f}$. If f is for example a function with learned parameters ω, it may be possible to learn $\tilde{f}$ by modifying its training data. If there is enough data available, the most naive approach to reduce the sensitivity to an artifact c_k, is to remove all samples X^+ that contain the artifact from training. Depending on the amount of available training data, this may not always be preferred, since these samples often contain other concepts that may be valuable for training. In contrast, if the number of samples with the artifact concept is larger than the number of samples without the artifact concept, one may instead discard all samples without the artifact to obtain an artifact-insensitive function. Of course care must be taken not to change the data so much that the original problem may not be solved anymore.

A better approach may be to transform individual samples, such that either all samples, or none contain the artifact. Assuming the addition of an artifact is non-invertible, we may prefer to transform all samples to contain the artifact. This may be done by estimating an *inductive* artifact model h, as defined in Equation (13). The model f may then be trained with the transformed dataset $X' = (x'_1, x'_2, ..., x'_N)$, with:

$$x'_i = t_i x_i + (1 - t_i) h(x_i; \hat{\theta}) \,. \tag{17}$$

A simplification arises when the task is to solve a classification problem. Since the model is trained to produce logits for multiple classes, one may simply balance the number of samples between classes, such that for each class, an identical amount of samples with an artifact are put into the training set by transforming non-artifact samples.

Another simplification arises when a regularization term is introduced in the artifact model, such that h acts as the identity for artifact samples $x^+ \in X^+$ with

$$\hat{\theta} = \arg\min_\theta \frac{1}{|X^-||X^+|} \sum_{x^- \in X^-} \sum_{x^+ \in X^+} \|h(x^-; \theta) - x^+\|^2 + \frac{\lambda}{|X^+|} \sum_{x^+ \in X^+} \|h(x^+; \theta) - x^+\|^2. \tag{18}$$

With this regularization term, the error caused by transforming an already-artifact sample is minimized.

Application on Logistic Regression. To build a better intuition for the problem, we introduce a logistic regression model $f(x) = \sigma(w^\top x + b)$ with sigmoid non-linearity $\sigma(x) = \frac{1}{1+\exp(-x)}$. The parameters w and b are obtained by minimizing the loss function

$$\mathcal{L}(f) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(f(x_i)) + (1 - y_i) \log(1 - f(x_i)) + \frac{1}{2} \gamma \|w\|^2 \,, \tag{19}$$

with labels $y_i \in \{-1, +1\}$, where $y_i = -1$ for samples of class A, using Stochastic Gradient Descent (SGD). We first consider the case $X^+ = \emptyset$, which is visualized in Figure 4 in the panel titled "Clean". In the panel, we see samples of two classes, A (blue) and B (orange), scattered along the vertical axis. The green lines visualize the decision hyperplane of f over 25 epochs of training. We can see that the final decision hyperplane (dark green) converged orthogonal to the signal direction on the vertical axis, separating classes A and B perfectly along their center. In panel "Artifact" of Figure 4, we introduce an artifact concept into some of the samples of class A, i.e., $|X^+| > 0$, which manifests as an increased value along the horizontal axis. The artifact samples are well on the right side of the panel. When now minimizing $\mathcal{L}(f)$, the converged decision hyperplane to which w is normal has rotated. While still classifying all the samples correctly, we can visibly see that the introduction of an additional concept has changed the model. Based on this observation, and the previous discussion, we introduce two approaches under the common name of Class Artifact Compensation to compensate for class-specific Clever Hans artifacts in SGD-trained models consisting of inner-products followed by non-linear activations such as logistic regression, or neural networks.

Augmentative Class Artifact Compensation. The goal of A-ClArC is to augment samples in such a way that the SGD-trained classifier becomes insensitive to an artifact given artifact labels t_i. Given these labels, we estimate an *inductive* artifact model h, which for our logistic regression toy model we define as purely additive, with:

$$h(x) = x + v, \tag{20}$$

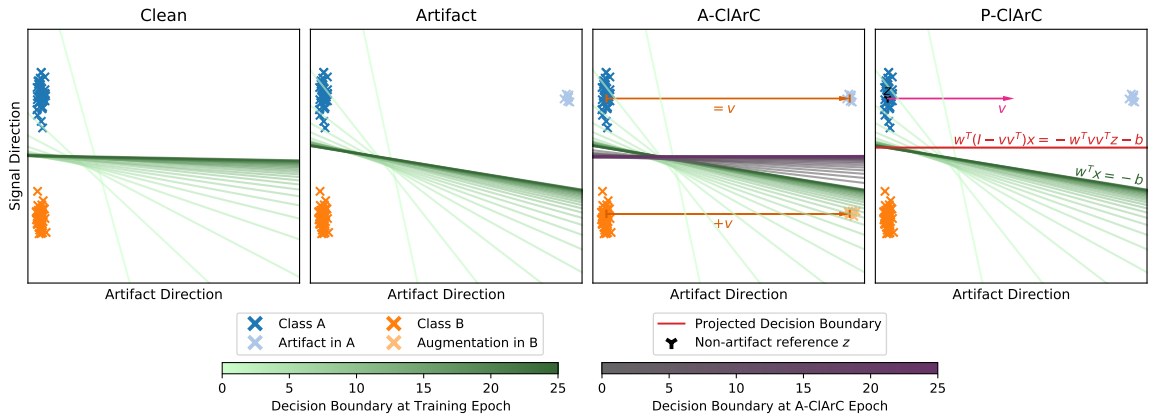


Figure 4: Logistic regression on data with, among possibly others, a discriminative *signal* direction and an *artifact* direction which is only represented in one of the two classes. The decision-hyperplane is shown over the SGD-based training-process of 25 epochs in shades of green, with: **Clean**: no artifact in the data; **Artifact**: a Clever-Hans artifact (light blue) in Class A (blue); **A-ClArC**: with *artifact*, but training is continued with the mean difference between clean samples and artifact samples in Class A added to some samples (light orange) of Class B (orange); The introduction of an artifact to samples from Class A changes the decision boundary. By introducing the same artifact direction to samples from Class B and retraining, this effect can be reduced significantly. **P-ClArC**: with *artifact*, but the model is modified such that data points are projected onto the hyperplane at position z to which the estimated artifact direction v is normal, with $\|v\| = 1$ and zero reference z chosen as the mean of clean samples of Class A. The resulting decision hyperplane ignores artifact direction v and sits at the same position where the original hyperplane lay between classes A and B, thus leaving the function output unchanged for clean samples. Reference z may be chosen as the mean of both clean and artifact samples of Class A to move the resulting decision hyperplane towards the middle of both classes.

where v is the vector added to a sample to make it an artifact. Given the objective from Equation (13), we can see that the optimal value for parameter v is

$$v = \frac{1}{|X_A^+|} \sum_{x_A^+ \in X_A^+} x_A^+ - \frac{1}{|X_A^-|} \sum_{x_A^- \in X_A^-} x_A^- \tag{21}$$

which is the shift between non-artifact samples and artifact samples in class A with $X_A^+ = \{x_i \mid i \in \{1, 2, ..., N\} \wedge y_i = -1 \wedge t_i = 1\}$ and $X_A^- = \{x_i \mid i \in \{1, 2, ..., N\} \wedge y_i = -1 \wedge t_i = 0\}$. This is visualized in panel "A-ClArC" in Figure 4. Some samples of class B $x_i^B \in \{x_i \mid i \in \{1, 2, ..., N\} \wedge y_i = +1\}$ are then modified given this artifact model with

$$x_i^B \leftarrow h(x_i^B). \tag{22}$$

The modified samples are visualized in Figure 4 in a light orange, shifted to the right. The model training is then continued with the transformed samples, of which the resulting hyperplanes over the epochs are visualized as purple lines. We can observe that the converged hyperplane resembles the one obtained by the model trained on artifact-free data in panel "Clean" of Figure 4.

Beyond this example, in our experiments with image data we assume artifacts are objects that are blended into the image. Therefore we may parameterize the artifact model as

$$h(x) = (I - \text{diag}[a])x + \text{diag}[a]z \tag{23}$$

where $a \in [0, 1]^d$ is the alpha channel, which is zero in all pixels except where the artifact is located, $\text{diag}[a] : \mathbb{R}^{d \times d}$ is a diagonal matrix with $\text{diag}[a]_{ii} = a_i$ with $i \in \{1, 2, ..., d\}$ and $z \in [0, 1]^d$ are the RGB values of the static image artifact pixels, here each for simplicity represented by a single value.

By taking CAVs as a motivation, we parameterize the *inductive* artifact models in our experiments for feature representations in a neural network in an alternative approach. Explicitly, we train a linear soft-margin SVM g with hinge-loss

$$\mathcal{L} = \frac{1}{2} v^\top v + \eta \left(\sum_{x^- \in X^-} \max[0, -v^\top x^- - \beta] + \sum_{x^+ \in X^+} \max[0, v^\top x^+ + \beta] \right) \tag{24}$$

with $v \in \mathbb{R}^d$, regularization constant η and bias term β. We then design the artifact model explicitly by pushing samples over the decision boundary relative to some fixed position z. We choose z as the mean artifact reference point, with

$$z = \frac{1}{|X^+|} \sum_{x^+ \in X^+} x^+. \tag{25}$$

The *inductive* artifact model h is then chosen as an affine transformation

$$h(x) = (I - vv^\top)x + vv^\top z. \tag{26}$$

Projective Class Artifact Compensation. While A-ClArC addresses the problem of desensitization by augmenting the underlying training data of a prediction model f using a *inductive* artifact model h, P-ClArC instead aims to correct the model without retraining by incorporating a *suppressive* artifact model h_s directly into the prediction model. The approach is again motivated by CAV and uses the same parameterization for the *suppressive* artifact model as the *inductive* model in Equation (26) with

$$h_s(x) = (I - vv^\top)x + vv^\top z \tag{27}$$

and v given in Equation (24). However, the artifact reference point z here becomes the non-artifact reference point, which we now choose as the center of non-artifact samples X^- with

$$z = \frac{1}{|X^-|} \sum_{x^- \in X^-} x^-. \tag{28}$$

This now moves all points along v to a fixed position, while leaving all orthogonal directions untouched. A strong assumption taken for this approach is that all other concepts are encoded in the directions orthogonal to v. Given this assumption however, we may further assume that for all non-artifact examples $x^- \in X^-$, $v^\top x^- \approx v^\top z$, i.e., there is no variance along the artifact CAV. With this, we further obtain $\forall x^- \in X^- : h(x^-) \approx x^-$, i.e., non-artifact samples are approximately unchanged by the *suppressive* artifact model h_s.

Given the logistic regression model f in Figure 4 in the "P-ClArC" panel, we obtain the model $\tilde{f}$ corrected for insensitivity against the artifact modeled by h_s using

$$\begin{aligned}
\tilde{f}(x) &= \sigma(w^\top h(x) + b) && (29)\\
&= \sigma(\underbrace{w^\top (I - vv^\top)}_{= \tilde{w}} x + \underbrace{w^\top vv^\top z + b}_{= \tilde{b}}) && (30)\\
&= \sigma(\tilde{w}x + \tilde{b}). && (31)
\end{aligned}$$

The "P-ClArC" panel shows the decision hyperplane of the original model f in green, along with the parameters v and z for the *suppressive* artifact model h_s, as well as the corrected decision hyperplane according to Equation (31). Note that the non-artifact reference z is chosen such that the corrected decision hyperplane of f is fixed on the line between the centers of classes A and B to satisfy the constraint that the corrected function values for non-artifact samples are fixed, resulting in a decision hyperplane that is parallel to the artifact direction, but closer to class A. An alternative z may be chosen as the mean of all samples of class A to correct for this difference.

We can transfer this approach directly to the neural network models in the experiments section due to their piecewise-linear nature. A detailed Algorithm for both A-ClArC and P-ClArC on Neural Networks is shown in Algorithm 2 under the common name of Class Artifact Compensation.

Algorithm 2: Class Artifact Compensation

Data: Samples $X = (x_1, x_2, ..., x_N)$
Labels $T = (t_1, t_2, ..., t_N)$ describing existence of artifact c in X (cf. Eq. 10)
Model f operating on X, with accessible layer l (and subnetwork f_l)
For A-ClArC: data D, epochs E for training, poison rate $p \in [0, 1]$
Result: predictor $\tilde{f}$ desensitized to artifact c

`/* obtain feature representations of data at layer` l `*/`

$A_l = \{\}$;
for $\boldsymbol{x} \in X$ **do**
  $a_{\boldsymbol{x}} = f_l(\boldsymbol{x})$;
  A_l.append($a_{\boldsymbol{x}}$);
end

`/* unlearn/deactivate the use of` c `in` f `*/`

if *using A-ClArC* **then**
  h_c^l = inductive_artifact_model(A_l, T);
  `/* def. A-ClArC module` $\tilde{g}_l$ `atop layer` l`, randomly apply artifact transform` h_c^l `*/`
  $\tilde{g}_l(a_x) \coloneqq \begin{cases} h_c^l(a_x) & : \mathbf{U}[0,1] < p \ \wedge \text{ model in training mode} \\ a_x & : \text{else} \end{cases}$;
  $\tilde{f} = f_L \circ \cdots \circ f_{l+1} \circ \tilde{g}_l \circ f_l \circ \cdots \circ f_1(\boldsymbol{x})$;
  `/* unlearn` c `in layers` $[l+1, \ldots, L]$ `*/`
  for $e \in \{1 \ldots E\}$ **do**
    $\tilde{f}$.train(D, `trainable=`$[f_{l+1}, \ldots, f_L]$)
  end
  `/*` $\tilde{f}$ `is the model corrected by fine-tuning on augmented data */`
else if *using P-ClArC* **then**
  h_c^l = suppressive_artifact_model(A_l, T);
  `/* def. P-ClArC module` $\tilde{g}_l$ `to suppress` c`, add on top of layer` l `*/`
  $\tilde{g}_l(a_{\boldsymbol{x}}) \coloneqq h_c^l(a_{\boldsymbol{x}})$;
  $\tilde{f} = f_L \circ \cdots \circ f_{l+1} \circ \tilde{g}_l \circ f_l \circ \cdots \circ f_1(\boldsymbol{x})$;
  `/*` $\tilde{f}$ `is the model corrected by projecting the output of a layer */`
return $\tilde{f}$

3. Experiments – Clever Hans Identification

The goal of this section is to explicitly find artifact models given sets of labels on our dataset regarding CH artifacts in the training set that were learned by the analyzed neural network model. Therefore, we start with an experiment to investigate the relation and difference between the detection of CH and BD artifacts within feature representations of neural networks [46]. The corresponding results point us towards the necessity of deeper insight into the model. Such an insight is promised by SpRAy [12], which we verify subsequently on a specially designed version of colored MNIST using our separability score extension. We specifically use LRP as the attribution method underlying SpRAy, as this provided us with better results than other considered approaches (e.g., SmoothGrad, Integrated Gradients). We then proceed to verify the proposed separability score τ on a VGG-16 model [90] trained on ILSVRC2012 by comparing the scores of classes for which we have manually found CH artifact candidates. A description of the training procedures and architectures of all models used in this section can be found in Appendix A. We proceed to visualize some promising CH artifact candidates which we have found in an algorithm-assisted dataset exploration with SpRAy, which provides us with a set of positive and negative labels on samples for each artifact candidate. An exploration is conducted both in input space and feature space in various layers of our model, for which we provide a comparison on the acquired separability scores. The previously obtained sets of labels may then be used to fit or construct an artifact model, which will be verified and used as prerequisite to remove the corresponding artifact from

a model using A-ClArC and P-ClArC in the following Sections 4 and 5.

3.1. Relation of Clever Hans and Backdoor Artifacts

In this section, we conduct an empirical demonstration on the difficulty of detecting CH artifacts compared to BD attacks by analyzing a neural network's hidden activations. We prepare two modified instances of the CIFAR-10 dataset [91], one poisoned by introducing a CH artifact, the other by adding a BD. In both cases, the trigger pattern is a static (3×3)-sized white pixel patch applied to a subset of the training set. For the CH, this trigger is introduced into 10% of all samples of class "airplane". For the BD, it is introduced into 1% of all samples, with the class label of each poisoned sample changed to "airplane", again resulting in 10% of the "airplane" class being poisoned (since the samples of each single class make up one tenth of the whole dataset). A simple convolutional network is then trained on each training set instance. This network achieves an unpoisoned test accuracy of 48.3% when trained using the CH artifact, and 46.8% with the BD-poisoned dataset.

As suggested by Tran et al. [46], the SpeSig method (cf. Section 2.1) is used to detect poisoned samples as outliers. While Tran et al. [46] use this outlier score only to detect BD samples, we also attempt to detect samples affected by the related CH effect in order to compare these two types of dataset poisoning in terms of their induced feature representation. We perform this analysis for the activations after the first fully connected layer (refer to Appendix A.1), i.e., directly before the output layer, to maximally consider the model's reaction to the presented images.

For each sample, an outlier score is thus obtained, yielding an implicit ordering of samples, with the highest score denoting the most outlying samples. For the datasets poisoned by a BD and CH, respectively, we then compare this ordering to the ground truth "poison labels".

Results. The results of this comparison are depicted as Receiver Operating Characteristic (ROC) curves in Figure 5. Here, independent of the poisoning during training time, we poison between 20% and 80% of the test data of the target class, with a CH or BD, respectively. Coinciding with the findings of [46], the BD candidates suggested by the outlier score correspond extremely well to the ground truth (Figure 5 *(right)*), with an Area Under Curve (AUC) of 1.0 for a 20% poisoning rate and 0.97 for a 40% poisoning rate. However, for the CH case in Figure 5 *(left)* this comparison yields almost random results, with respective AUCs of only 0.62 and 0.59. Note that for poisoning rates above 50%, the ROC curves and corresponding AUCs are inverted (below 0.5), since when the majority of samples are poisoned, the roles of outliers and inliers are reversed.

Above results are further supported by the subsets of samples that were selected as outliers via SpeSig. For the CH, the outlier detection correlates with the blue sky background of some samples within the "airplane" class, in addition to the pixel-patch poisoning. The artificially introduced CH is comparatively weak, and only determines a sample's outlierness in conjunction with other, already preexisting class features. In contrast, the difference between poisoned and unpoisoned samples is apparent in the corresponding attribution maps.

This experiment highlights the difference between BDs and CH artifacts, and emphasizes the additional issues that are present when dealing with the latter:

Intuitively, features introduced by BD artifacts will be the only feature in their respective sample to correlate with the target label, making them for many samples the only indicator usable for a valid prediction. Additionally, they must be an indicator stronger than all features that correlate with labels different from the BD target label for a correct prediction. This may very well be the reason they can be detected so evidently using only the direction of the largest variance in feature space over the dataset with SpeSig. In contrast, features introduced by CH artifacts will always appear alongside other features in their respective sample that correlate even stronger with the target label. This means that in theory, they are not necessary for a correct prediction at all.

To detect CH artifacts more reliably, deeper insight into the predictor is necessary. A promising direction is thus XAI, which is utilized in SpRAy to detect these elusive CH artifacts in the rest of this section.

An interesting note to make is that FDA can be understood as an extension to simply finding the direction of the largest variance as done in SpeSig, as given a set of labels, the direction of largest variance between labels and smallest variance within labels is found.

For the remainder of this work, we focus only on the more elusive CH artifacts, since they naturally appear in many commonly used datasets, and thus often cause an overestimation of a model's generalization capabilities.

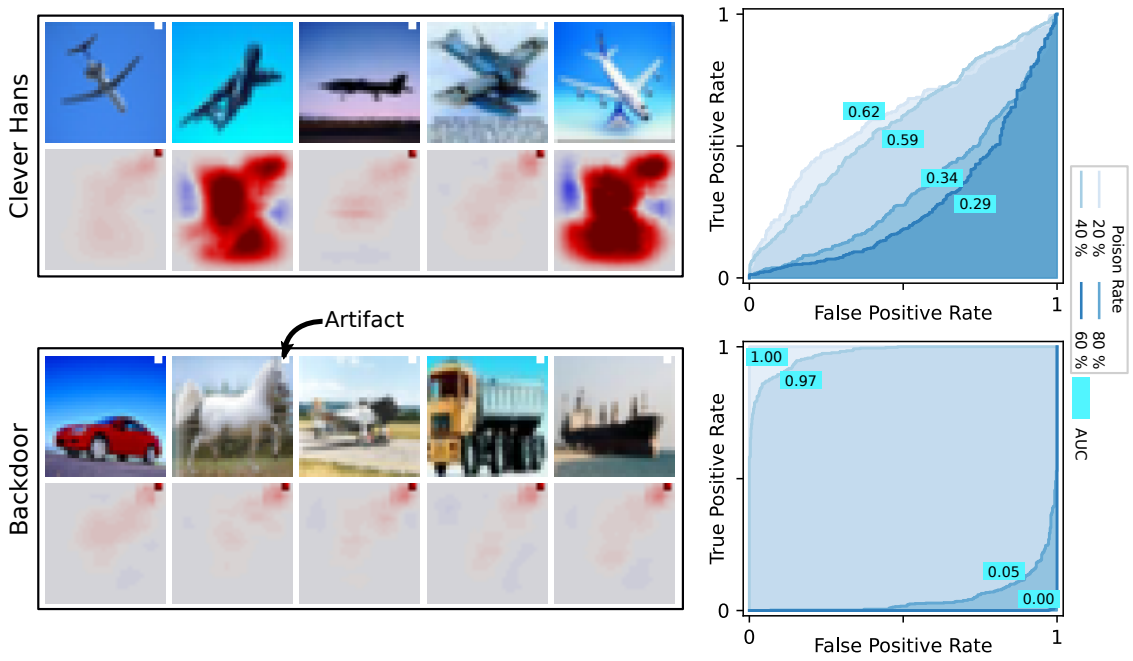


Figure 5: Differences in the detection of CH artifacts *(top)* and BDs *(bottom)*. In both cases, the introduced artifact consists of a small white pixel patch in the top right corner of each sample. *(Left)*: a subset of the samples and the corresponding LRP attribution maps from the 20% poison rate setting that were identified as outliers via SpeSig. All samples considered as outliers in the BD setting do in fact contain the BD feature. The same evaluation performed in the CH setting leads to a significant amount of false positives for the detection of the CH artifact. In contrast, both the CH and the BD artifacts can easily be found when taking the attributions into consideration. Attribution maps (LRP) show image regions that support the classifier decision in *red*, and regions that contradict the final prediction in *blue*. Refer to Supplementary Figure B.1 for a more detailed color description. *(Right)*: The large difference between BD and CH is further confirmed by the ROC curves comparing poisoned samples detected by SpeSig to the ground truth. Note that in both cases, 1000 evenly spaced thresholds were used for the AUC/ROC computation.

3.2. Spectral Relevance Analysis in Input Space

We explore SpRAy for the identification of Clever Hans artifacts in input space. We start with a verification of the algorithm by constructing a modified version of MNIST where an artifact is introduced as a distinct color. We then proceed to analyze the applicability of SpRAy on input attribution space on the ILSVRC2012 dataset.

Figure 6: Examples of the colored MNIST dataset, with a distinct color-based CHs artifact introduced into 20% of each class of the MNIST dataset. Each column shows several samples of one class each.

Spectral Relevance Analysis on Colored MNIST. The SpRAy framework is applied on colored MNIST setup as following. For each of the 10 MNIST classes, we create a dataset where for the corresponding class, samples are colored with a probability of 20 percent in a distinct color as shown in Figure 6. The rest of the samples are left in their original white color. On each of these datasets, a simple feed-forward convolutional neural network is trained (cf. Appendix A.2). We can then verify for each model how much it has learned the color to be a distinct feature for the corresponding class, by evaluating the model accuracy and the fractions of the predicted classes on a test set which has been completely colored in the color of the artifact. Subsequently, we do a Spectral Relevance Analysis by using 4 neighbors to build an affinity graph of the attributions to compute the Spectral Embeddings reduced to the dimensions corresponding to the 2 smallest eigenvalues. Note that we did not sum over the color channels of the attributions, as is often done for visualization purposes, since the color plays an important role in this experiment. We do a simple agglomerative clustering with 2 clusters on the Spectral Embedding, and compute its separability score τ. The aforementioned results are visualized in Figure 7. The Spectral Embeddings at the *(bottom)* of Figure 7 form a crescent-like

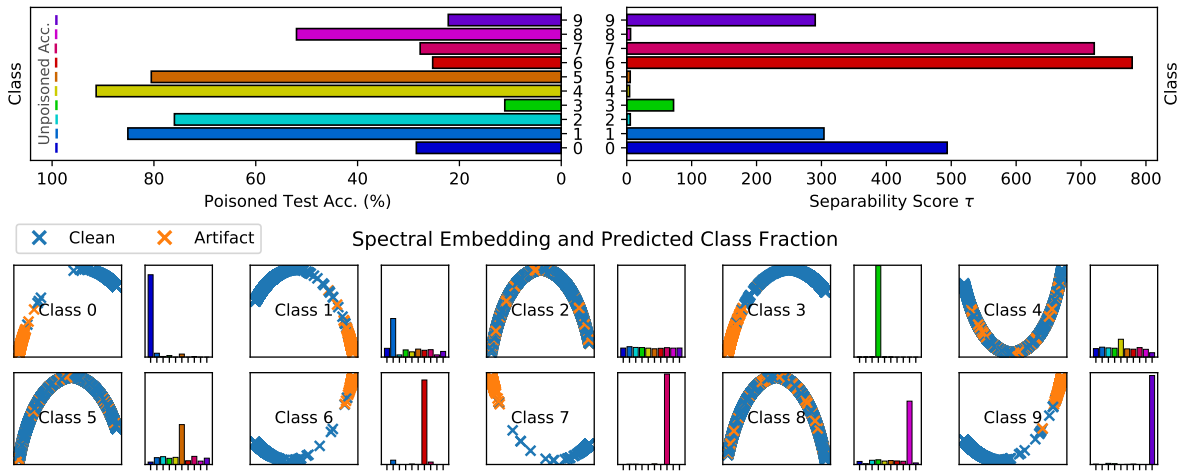


Figure 7: *(Top):* accuracy on the poisoned dataset *(left)* and separability score τ of 2-cluster agglomerative clustering *(right)* where the class included a Clever Hans. *(Bottom):* Spectral Embedding based on 4 neighbors and 2 eigenvalues of each individual class on its corresponding dataset, where orange crosses are colored artifact samples and blue crosses are uncolored clean samples. The predicted class fractions are shown for each class to the *(right)* of its Spectral Embedding. For each of the 10 classes, a modified MNIST dataset was prepared where 20 percent of the samples in that particular class were colored to act as artifact samples. One model was trained on each of these datasets. The poisoned accuracy is the accuracy of each of these models on the test set with each sample colored in the same artifactual color. The used colors were the same as the ones shown in Figure 6. The class-wise accuracy when 20% of the CH class in the test set are colored (as in the training setup; vertical lines in the poisoned accuracy plot) is slightly above 99% for all classes. Models with a high poisoned accuracy and a low separability score indicate that the model is not strongly confused by the artifact. Models with a low poisoned accuracy and a high separability score indicate that the artifact causes strong confusion. The Spectral Embeddings show a clear split for models where the separability score is high.

shape for all classes. When the attribution can be well separated, clean and artifact samples move towards the opposite ends of the crescent. This is visible for classes 0, 1, 3, 6, 7 and 9. These classes also show a high separability score τ when compared to the scores of the other classes. With the exception of class 1, all of these classes also show a low performance on the poisoned test set, with an accuracy below 30 percent. In contrast, again with the exception of class 1, all classes with an accuracy above 50 percent show a separability score close to 0. The predicted class fraction for classes with a high separability score show a high tendency of the model to predict poisoned samples as the artifact class, especially for classes 0, 3, 6, 7 and 9. Even though the reduced accuracies suggest that the model has somewhat learned to associate the color as a class-relevant feature in classes 2 and 4, the separability score τ is low, and thus the presence of an artifact is uncertain in these cases given SpRAy. Class 8 shows a high confusion even though the separability score is close to 0. In contrast, class 1 shows comparatively low confusion even though its separability score is high. It is worth to note that all models show a reduced accuracy on the poisoned test set compared to the accuracy on a clean test set of about 99 percent, even for class 2 where the confusion does not seem to focus on the artifact class. This means that even though we may confuse models by coloring all samples of the whole test set, we cannot detect the artifact in some of these models using SpRAy. Only part of the reason for this seems to be that the

model is not strongly associating the artifact with the class, since for example class 8 shows a relatively high tendency to confuse colored samples for their corresponding artifact class, yet SpRAy does not give a strong indication of an artifact in the class. Interestingly enough, three of the four CHs that proved the most difficult to detect, which are in classes 2 (turquoise), 4 (yellow) and 8 (purple), are colors for which two of the three color channels have the maximum value, thus being considerably closer to the original white color in euclidean space, where all color channels are at the maximum value.

Concluding this experiment, the assigned importance of an artifact may vary greatly between models and classes, and even though we may not find the artifact in all instances where the model has in fact picked up an artifact as an important feature, SpRAy pointed out most artifacts in this setup.

Quantifying Clever Hans Candidates on ImageNet. We examine ILSVRC2012 for CH candidates by applying SpRAy with various clustering approaches for which we compute cluster separability scores τ (Eq. (7)) for each class. Figure 8 lists a ranking of the ImageNet classes with the highest and lowest τ values with a striking result for class laptop, due to a large cluster with copies of almost the same image (see UMAP of its Spectral Embedding in Figure 10 *(bottom right)*).

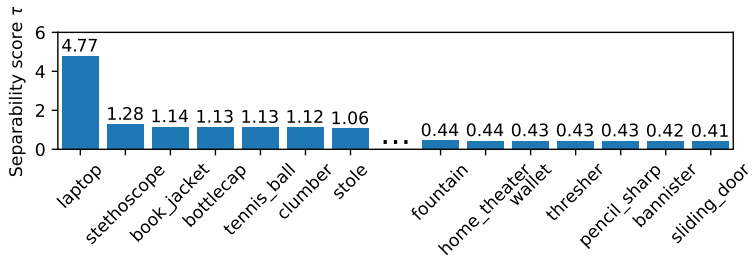


Figure 8: Mean separability score τ of Spectral Embedding of attributions based on FDA. A high τ indicates high linear separability between attribution maps within the class. This may highlight a cluster of samples with an outlier-attribution, potentially of CH type.

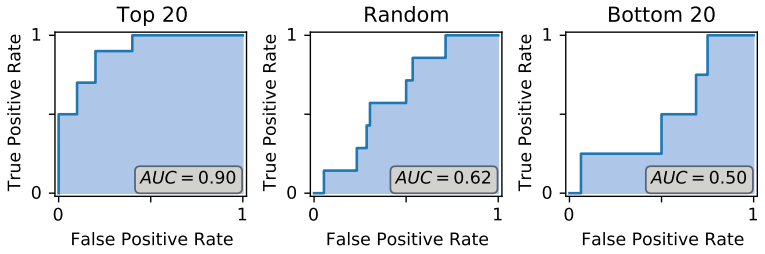


Figure 9: ROC curves for artifact-existence versus FDA-Ranking. *(Left)*: top 20 classes with highest values of τ. *(Mid)*: 63 random classes with any values of τ. *(Right)*: bottom 20 classes with lowest values of τ.

We inspect the validity of the class ranking for CH candidates generated by FDA in a small experiment, by screening a subset of all 1000 ImageNet classes, namely (1) those with the 20 highest τ scores, (2) those with the 20 lowest τ scores and (3) 63 randomly picked classes. In all three cases, we assume a positive CH "prediction" per class due to a large value of τ. We then produce "ground truth" labels via manual assessment of the existence of a CH candidate. We would like to remark that this "ground truth" has been established based on the class label description in the taxonomy of the ImageNet dataset and our subjective human understanding of the image content. Using this information we produce ROC curves and corresponding AUC values.

The results show a clear picture validating that a high τ score is indeed a strong indicator for the presence of CH candidates (Figure 9 *(left)*, high AUC). Both randomly selected and bottom 20 classes (Figure 9 *(mid, right)*) yield essentially random AUC scores due to only sporadically encountered CH. However, the AUC $\gg 0$ here also show that even a τ rating in the lowest 2-percentile does not guarantee a class to be free of CH behavior.

Intuitively, a large separability score τ indicates high linear separability between embedded attributions. This suggests that there may be a cluster of outlier attribution maps, which implies that the model uses a highly different

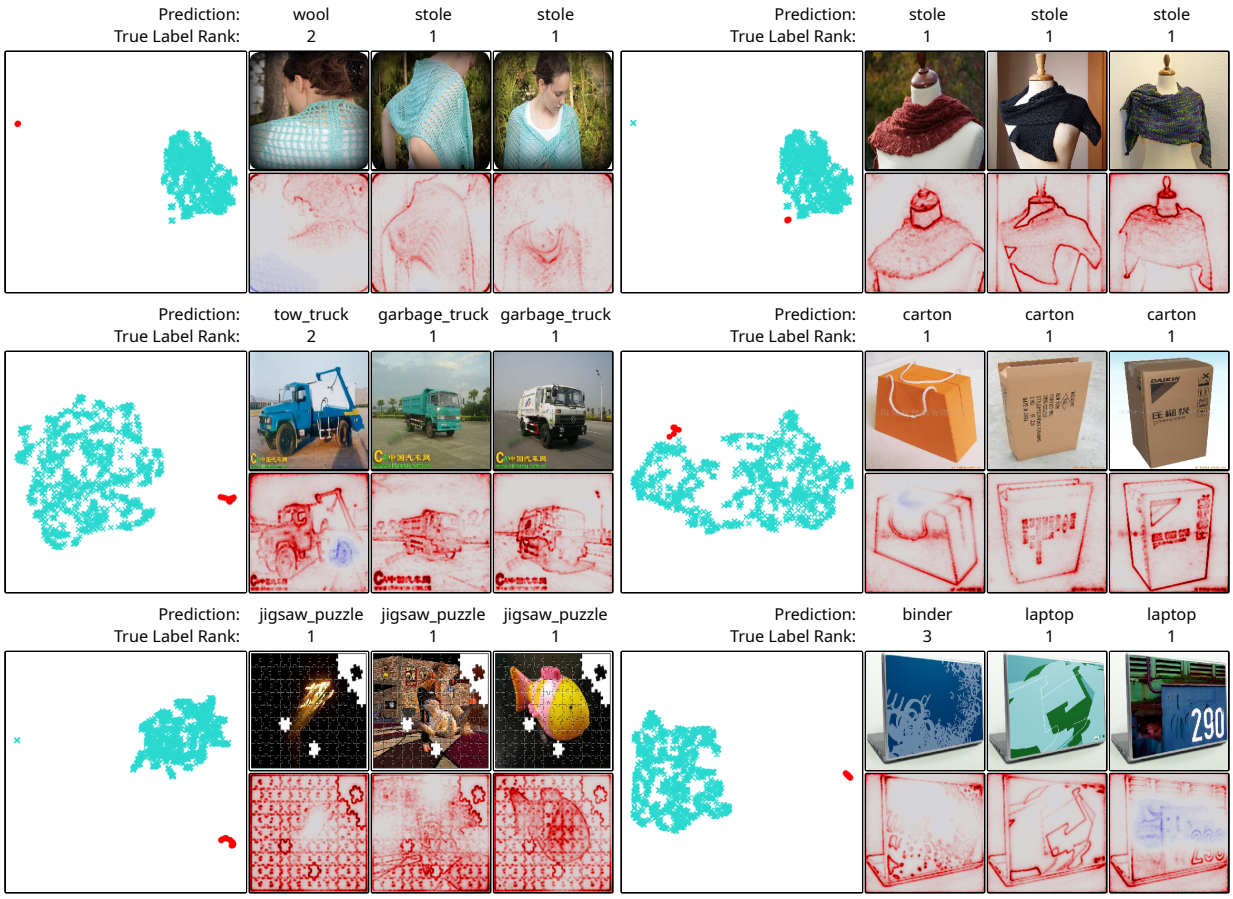


Figure 10: Each panel shows the UMAP *(left)* with samples and heatmaps *(right)* of significant clusters, highly separated from the rest of the samples. For each cluster, example images *(top)* and their respective attributions (from the identified cluster are shown. The location of the samples in the UMAP are highlighted in red. Attribution maps (LRP) show relevant image regions *supporting* the classifier decision in red, irrelevant regions in grey color and relevant regions *contradicting* the final prediction in blue. Refer to Supplementary Figure B.1 for a more detailed color description. Above the sample images the classifier's top-1 predicted class and the prediction rank of the true label are shown.

and possibly exploitative strategy for the corresponding samples. On the other hand, a low separability score indicates a low linear separability which is no indicator for the absence of an artifact since there potentially could be a non-linear, non-trivial CH strategy that cannot be linearly separated in the embedded space. Further research will be needed here to ideally bring forward indicators that can provide a theoretical bound for absence of CH behavior.

Inspecting and Isolating Clever Hans Candidates. Based on the ordering by FDA and τ established in the previous section, we manually investigate whether the CH candidate classes show prominent CH artifacts to be expected, via interactive visualizations in *ViRelAy*[5] [73]. The SpRAy framework provides as a side effect (through its Spectral Embedding space Φ) also a basis for visualizing clusters of heatmaps, here we use UMAP. Promising clusters are often located far away from the rest of datapoints in the UMAP embedding, see e.g., Figure 10 *(center left)* the UMAP scatter-plot of class "garbage truck". There, the red cluster-members all show examples of images of the same watermark with high attribution in LRP. Another intriguing example is the top middle UMAP plot of class "stole": while not as separated as for other examples, we find a cluster of mannequins wearing stoles, with high attribution scores on the mannequin's "head". For class "carton", we can see even two artifacts at the same time: watermark written with Hanzi in the center of the image, as well as a watermark in latin characters in the bottom right. The bottom right watermark is in fact not only present in the carton class.

[5] https://github.com/virelay/virelay

Based on the clustering labels provided by SpRAy, for each artifact, we may extract a set of labels that indicate whether a sample is affected by the artifact candidate. Using these labels along with the corresponding samples, we may estimate an artifact map according to Section 2.6, which given a clean sample creates a poisoned version of the sample with the artifact present. This may be done for example by training a generative model conditioned on the presence or absence of an artifact, manually extracting a watermark from an affected image using an image manipulation framework, or something as simple as fitting a linear regression model. For A-ClArC in input space, we manually extract the artifact from samples labeled as poisoned, such that we can apply it to samples by a simple affine transformation $h(x) = (I - \text{diag}[\alpha])x + \text{diag}[\alpha]z$ as given by Equation (23) where z is a vector with the pixel values of the watermark, and α an alpha channel the same size as the number of pixels, which is zero for all pixels except the ones where the watermark is present. For P-ClArC, we instead use the labels to train a linear classifier $f(x) = v^\top x + b$ with $\|v\| = 1$, which is used to instead estimate an inverse artifact map as an affine transformation $h(x) = (I - vv^\top)x - vv^\top z$, where z is chosen as the mean over all clean samples of the class, as highlighted in Section 2.6.

3.3. Spectral Relevance Analysis on ImageNet in Feature Space

Until now we have based our SpRAy solely on model attributions in input space. While this has not been explored by Lapuschkin et al. [12], we attempt to base the analysis on model attributions in feature space for additional insight and compare the obtained separability scores over the various intermediate representations at different model depths. The motivation behind using intermediate representations is that the model must encode increasingly invariant representations of concepts towards its classification task in higher levels, which may not be detectable with the contribution scores in input space. We investigate which clusters of samples contribute the most towards the separability score of a given class. To this end, we compute the separability score τ as many times as there are clusters, with samples from one cluster withheld in each iteration. In this setting the group of clusters with the lowest separability score will have had withheld the cluster of samples with the highest contribution to the outlierness of the class. Complete class separability scores, along with samples of clusters with the highest outlier score, are reported in Figure 11.

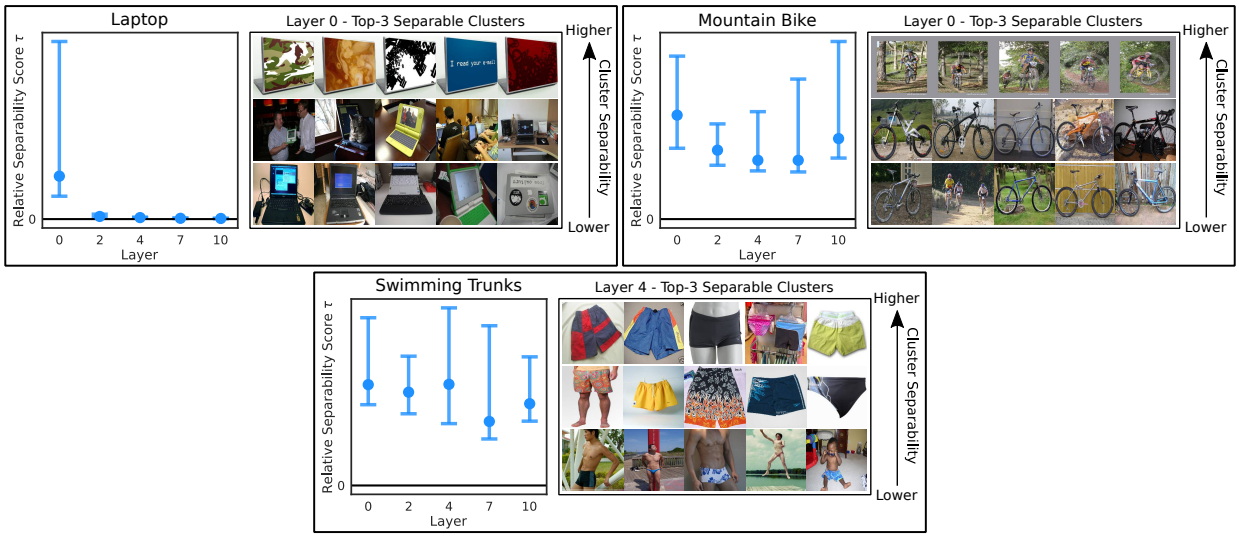


Figure 11: Separability of various clusterings of Spectral Embeddings Φ in multiple layers of VGG-16. Each FDA score τ indicates the separability of the whole embedding with K clusters at a specific layer. The 25th, 50th, 75th percentile measurements of the FDA scores τ *(left)* at each layer are varied over the number of clusters ($K \in \{2, \cdots, 32\}$) chosen for SpRAy. The shown ImageNet classes are laptop *(top left)*, mountain bike *(top right)* and swimming trunks *(bottom)*. Note that the measured absolute magnitude of the separability score τ might be different between the three classes, and so, only *relative* within-class comparisons can be inferred here. The scores τ vary strongly over various layers for different classes. E.g., the separability score for "laptop" is comparatively large at the input layer, but then decreases with increasing depth of the layer. "Swimming Trunks", on the other hand, seems to separate best at layer 4. For this layer of maximum separation score, examples of the top three separating clusters are shown to the *(right)*, revealing possible CH artifacts. For this purpose, $K = 32$ clusters were chosen, an amount that offers a fine-grained yet still semantically meaningful clustering. The top separable clusters are found by removing singular clusters from the clustering, and observing how the separability score τ changes. The clusters for which τ diminishes the most are the top separable clusters.

In this figure, the above analysis is shown for the three example classes "laptop", "mountain bike", and "swimming trunks" . Within each panel, a relative comparison of the separability score τ over layers, i.e., the input layer (layer 0) and various intermediate layers obtained from the model's convolutional feature extractor (layers 2-10) can be found to the *(left)*. At each layer, measurements vary over the chosen number of clusters $K \in \{2, \ldots, 32\}$, with the respective mean shown as a colored dot. However, a high τ does not necessarily *only* occur due to the presence of a CH, although *if* a CH is present and well represented, a high separability score is likely. Thus, correspondingly on the *(right)* side of each panel, for $K = 32$ and the layer with the *highest* mean τ in Figure 11 *(left)*, samples of the top three clusters in terms of contribution to separability (i.e., the separability score decreased the most when this cluster was left out) are visualized. We chose $K = 32$ due to 32 being the largest number of clusters we explored, leading to the most fine-grained results in terms of top separable clusters, but still small enough to allow for semantically meaningful clusters. The most contributing cluster is shown in the *(top)* row, decreasing towards the *(bottom)* row.

We find that the separability scores vary significantly with the layers: for the "laptop" class, the clearly highest separability score appears at the input layer. Here, a cluster showing laptop lids has the largest separability contribution, showing the same laptop (albeit with different patterns printed on its lid), digitally rendered from the same angle in each sample in front of a white background. Thus, this cluster seems to describe a CH artifact. Results for the "mountain bike" class behave in a similar manner. Again, the highest separability score is found at the input layer, and, correspondingly, the cluster with the highest separability contribution there seems to contain a CH in the form of a distinctive gray border and a watermark.

In contrast to the first two examples, the largest mean τ value for the class "swimming trunks" occurs not at the input layer, but at intermediate layer 4 of the model instead. Again, the top contributing cluster consists of relatively similar samples, however, they are all perfectly *valid* examples of "swimming trunks", with no distinguishable artifact between them. The same seems to be the case for the second most contributing cluster. Interestingly, the third most separable cluster is extremely dissimilar to the first two, with every sample containing male upper bodies – a feature that, while often appearing alongside "swimming trunks" should not indicate this class in any way. In other words, a CH.

This last example demonstrates why it may be difficult to automate the process of CH identification: While a CH is in fact present in the class, it is not the *top* separating cluster, but has the third highest contribution (of 32 total clusters) to the τ score instead. More concisely, the most separable cluster is not necessarily a CH, and a high separability score does not guarantee the presence of a CH. Thus, SpRAy offers an indication of which clusters in which classes are CH *candidates*, but – in accordance with the property of CH artifacts of requiring expert domain knowledge to detect (Section 1.1) – human judgement is still required for a final decision. We further note that the CHs found in the first two examples are relatively simple features. They can, in fact, be expressed as an affine transformation in input space. Correspondingly, the highest separability score for these classes occurs in input space. In contrast, the third presented example, where the "upper body" CH was identified, is far more complex, but the highest τ score is also found at a deeper intermediate layer. Thus, there seems to be a correlation between the *complexity* of an artifact, and the *depth* of the layer at which it separates best from the rest of the class.

4. Experiments – Concept Desensitization with Augmentative Class Artifact Compensation

In the previous section, we obtained cluster labels for (potential) CH artifacts, and correspondingly are able to estimate artifact models for CH candidates in ILSVRC2012 according to Section 2.6. The goal of this section is to verify the impact of these artifacts candidates on our classification model and at the same time reduce their impact by using A-ClArC.

CH artifacts appear – by definition – alongside desired features of a class. Furthermore, each CH only natively occurs within one class and helps a model predict this class correctly. As such, if unlearning is successful, a *decrease* in the measured accuracy (as opposed to the *true* generalization accuracy) is to be expected, making it difficult to distinguish from simply confusing the network. Due to these unique properties of CH artifacts, our method for evaluating the experiments is two-fold: a quantitative evaluation of whether A-ClArC leads to a desensitization against a concept representation, combined with a qualitative assertion of whether this representation corresponds to the target concept and leads to an unlearning thereof. It is important to understand that similar to the difficulty in guaranteeing the absence of a CH artifact, a complete removal of a model's sensitivity to an artifact can neither be guaranteed. We can only guarantee an alleviation of the CH by our (in-)ability to detect it afterwards.

We first verify A-ClArC empirically by introducing a controlled setup based on a variation of MNIST, where artifacts are introduced as colors. With the established verification, we proceed an attempt to unlearn CH artifacts candidates using A-ClArC given the artifact estimators modeled after Section 3, first in input space, then in feature space, and at the same time measure their respective impact on the classification model.

4.1. Augmentative Class Artifact Compensation on Colored MNIST

Setup. As an empirical verification of the method, A-ClArC is applied on a simple convolutional feed-forward type network (cf. Appendix A.2) on the previously described MNIST dataset with color artifacts. Here, we train the three variants of the model: (1) For the first model, of the 10 different classes, the samples of one class are colored with a probability of 20 percent during training. We call this the *baseline model*. (2) Another model is trained, but in addition to coloring the same single class as before, we also color samples of all other classes with a probability of 20 percent, using the same color. We call this model *a priori A-ClArC*. (3) For the third model, we continue training from the learned *baseline model*, but also color according to the *a priori A-ClArC* samples of all classes with a probability of 20 percent, using the same color. This model we call *a posteriori A-ClArC*.

To evaluate the influence of the (here color-based) CH, we introduce two test modes. One test mode describes the performance of the models on a realistic test set, where samples of the CH class are colored with a probability of 20 percent. The second test mode describes the performance of the models on a (poisoned) test set where all samples of all classes are colored. By comparing these two performances, we get a measure of the error caused by the CH. Note that in this toy setting we could measure the performance of the model on a clean, CH-free test set, which would normally not be available. But since the performance on the realistic test set is only marginally better (around 0.02 percent) than the performance on a clean dataset for the *baseline model*, we do not see a reason to require omniscience to compare against a test set without any artifacts.

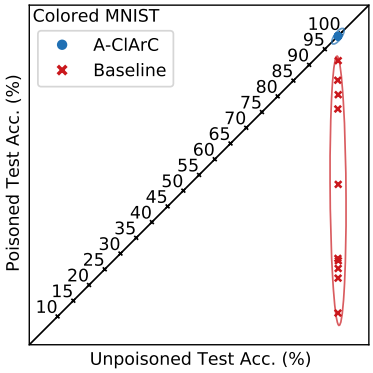


Figure 12: Accuracy on a realistic test set (horizontal axis) vs. accuracy on a fully poisoned test set (vertical axis) on colored MNIST. Points on the diagonal indicate the same performance on both datasets. Points below the diagonal indicate a higher performance on the unpoisoned test set. Points above the diagonal would indicate a higher performance on the poisoned test set, which does not occur. Red crosses describe the baseline, a *baseline model* which has seen a CH artifact during training for 20 percent of the samples of one class. Blue dots describe a fine-tuned version according to ClArC, the *a posteriori A-ClArC*, of the aforementioned models. For each artifact setup (CH is color, one setup per class), there is one blue dot, and one red cross (i.e., the red cross lowest on the vertical axis would correspond to the experiment were 20% of class 3 were colored green, see Figure 7). The red and blue ellipses describe the confidence of the points. For visualization purposes, the ellipses are drawn with 40σ in horizontal and vertical direction for *a posteriori A-ClArC* and 40σ in horizontal direction and 1.4σ in vertical direction for the *baseline models*, where σ is the standard deviation of the accuracies in the respective direction. Note that the blue dots for the *a posteriori A-ClArC* are all slightly above 99% on both test sets, making them visually indistinguishable. The values for both axes are annotated to the diagonal line, along which the accuracies on both test sets would be equal.

Results. Figure 12 shows the accuracy on the realistic test set on the horizontal axis and the accuracy of a maximally poisoned dataset (all samples colored) on the vertical axis. The *baseline models* are represented by red crosses, while the *a posteriori A-ClArC* are represented by blue dots. All models achieve an accuracy of about 99 percent on the realistic test set. As one would expect, the *baseline models* perform considerably worse on the fully poisoned test set. Some models are only slightly impacted by the poisoning, which means they do not pay as much attention on the CH

artifact (color). Other models however perform as bad as only 10 percent accuracy, which has the model predict the class only based on the CH artifact. Fine-tuning the model according to A-ClArC, as done for the *a posteriori A-ClArC* model, results in all models now performing very closely to how they perform on the realistic dataset. Therefore, the models have successfully been fine-tuned to ignore the CH artifact. We have therefore empirically shown the effect of CH artifacts on the model, as well as shown the effectiveness of A-ClArC.

4.2. Augmentative Class Artifact Compensation on ImageNet

Setup. We conduct a similar setup to the one used on colored MNIST on ILSVRC2012. Due to the size of the dataset, the setup is slightly modified, where we fine-tune in two configurations from a pre-trained model, which already shows CH behavior. The first configuration, the *baseline model*, is a simple continuation of the training, with a reduced subset of 100 classes of the original ILSVRC2012, and serves as a baseline when no action as taken against CH artifacts. The second configuration, *a posteriori A-ClArC model*, is continuation of the training with the same reduced subset, but where A-ClArC was used to augment samples accordingly. Note that an *a priori A-ClArC model* cannot be trained when fine-tuning a pre-trained model which has already seen CH artifacts.

One *a posteriori A-ClArC* and one *baseline model* is trained for each artifact candidate model we have identified in Section 3. We fine-tune on the original model for a total of 10 epochs, and report the model accuracies using the two previously introduced test modes, where we use the original test set (0% poisoned) as well as the original test set with the artifact introduced into all samples (100% poisoned) in Figure 13. The model performances are also compared for these two test modes in scatter plots in Figure 14. As an additional approach to evaluate whether the importance of artifact was reduced for the prediction of each model, we visualize the difference between the attribution of the original model (R_{org}), and either the *a posteriori A-ClArC* ($R_{\text{A-ClArC}}$) or the *baseline* model (R_{baseline}) with

$$R^{\text{diff}}_{\text{A-ClArC}} = R_{\text{org}} - R_{\text{A-ClArC}} \qquad R^{\text{diff}}_{\text{baseline}} = R_{\text{org}} - R_{\text{baseline}} \tag{32}$$

in each epoch in Figure 13.

Results. The performance for both the A-ClArC and the baseline model do not seem to change considerably for artifacts "stole" and "garbage truck" when looking at the performances in Figure 13. This can be seen by very similar confidence ellipses in Figure 14 over all epochs and an additional poisoning of the training data at 50% for class "garbage truck". Class "stole mannequin" in Figure 13 which corresponds to "stole mannequin head" in Figure 14 shows however a slight improvement in the poisoned test mode in the latter. Class "carton Hanzi" in Figure 13 which corresponds to "carton Hanzi" in Figure 14 shows a clear improvement over the baseline model on the poisoned test set. We can see a strong decline in the performance on the poisoned dataset for "jigsaw puzzle" for the baseline model, likely caused by the fact that the artifact is a very clear indicator for the class. However, the A-ClArC returns to about 50 % of accuracy on the poisoned test set. For none of the artifacts in Figure 13 we see the A-ClArC model perform worse than the baseline model. By investigating the heatmap differences of the A-ClArC and baseline model to the original model, we can see that the A-ClArC model consistently decreases the amount of relevance assigned to the artifact location in the input image. For "garbage truck", there is a watermark in the bottom left corner of the image. The A-ClArC attribution subtracted by the original attribution shows strong positive values on the watermark location, indicated by the magenta color. The baseline attribution difference to the original model seems to decrease and increase the relevant pixels more generally focused on edges in the image. Even though the baseline model also seems to weakly reduce the relevance on the watermark in the second epoch, this is not as targeted and consistent as for the A-ClArC model. A similar behavior can be seen for the class "stole mannequin", where the A-ClArC model consistently reduces the relevance on the mannequin, while the baseline partly even reduces the relevance on the stole itself. For the "carton Hanzi" artifact, which here corresponds to the Hanzi in the center of the image, the A-ClArC model also reduces the relevance on the characters, mostly concentrated at the higher contrast area at the right hand side of the image. The baseline model even increases the relevance on the location of the watermark and decreases the relevance on the actual cartons compared to the original model. While somewhat harder to see, the A-ClArC model seems to reduce the jigsaw pattern away from the object of interest that the baseline model for "jigsaw puzzle".

Similarly, Figure 14 gives similar insights for "carton Hanzi" and "jigsaw puzzle", where all A-ClArC models (poisoned) perform comparably on the original dataset, but outperform the baseline significantly on the poisoned test set. With "stole mannequin'', A-ClArC outperforms the baseline slightly. "carton alibaba'' seems to only weakly

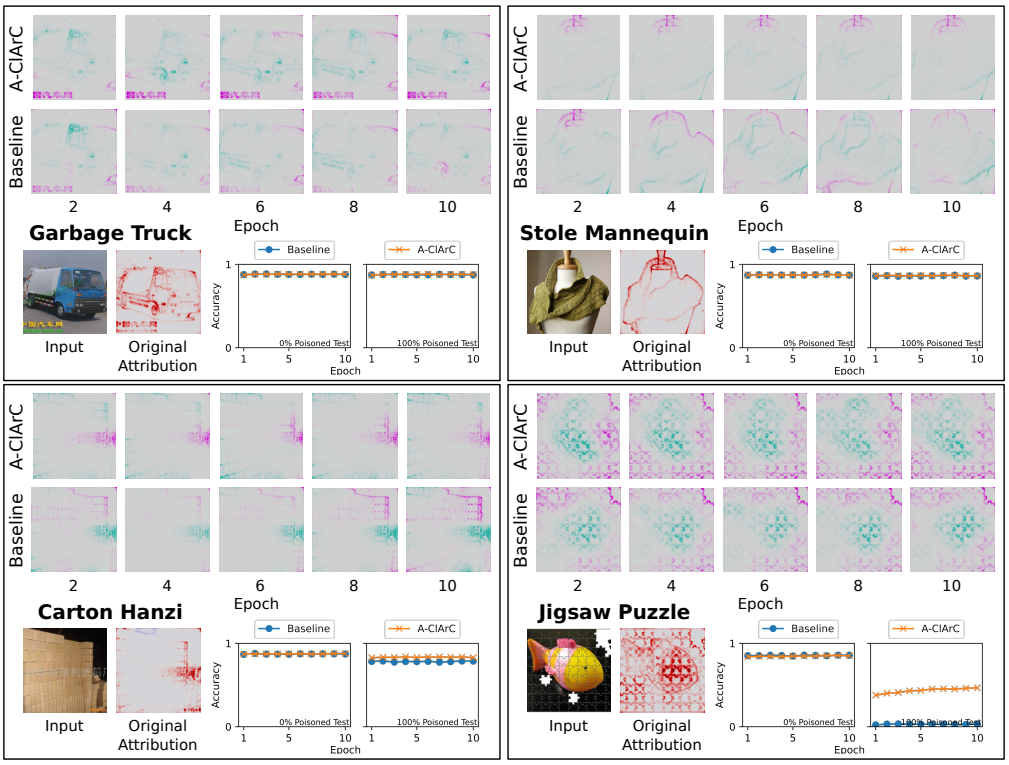


Figure 13: In each panel: an input example with a CH artifact alongside its attribution (LRP) of the original model *(bottom left)* , where red values indicate relevance for the classifier's decision, grey indicates no relevance, and blue regions indicate relevance against the classifier's decision. The pixel-wise difference (see Eq. (32)) between the original attribution and attributions for an A-ClArC fine-tuning (*a posteriori A-ClArC*) at every 2 epochs *(top)*, with attributions of a baseline model below which was fine-tuned with the same hyperparameters, but without modifying the training set. Magenta means the original model assigns more relevance to the highlighted part, while teal describes the opposite. Refer to Supplementary Figure B.1 for a more detailed color description. The model performance *(bottom right)* is shown on both the unchanged test set (0% poisoned) and the poisoned test set (100% poisoned) over the epochs for both the baseline model and the A-ClArC model. The observed artifacts are: *Garbage Truck* – a watermark in the bottom-left; *Stole Mannequin* – a mannequin head in the class "stole"; *Carton Hanzi* – a watermark with Hanzi in the center of the image; *Jigsaw Puzzle* – a digitally created puzzle pattern pasted onto the image. Note that the images are slightly center-cropped.

affect both models, with no visible improvement for A-ClArC. The Alibaba watermark is found not only in class "carton", but in many classes of ILSVRC2012, and is a rather small artifact in the bottom right of the image, possibly cropped most of the time during training, which is why it may not be a very strong artifact for class "carton" alone. "stole corners" is also a very small artifact at the corner of only a few samples in class stole. Presumably for this reason, we do not see either model particularly impacted by poisoning the test set. The "garbage truck" artifact result is somewhat surprising, both models seem to be only slightly affected by the poisoned dataset, with only at best a very slight improvement of the A-ClArC model over the baseline model.

Therefore, we may conclude that A-ClArC in input space seems to alleviate artifacts that are very significant in input space, but may not show any significant effect otherwise. An important limitation to note is that while performance is retained and the model's focus on the artifact in the attribution maps is decreased (backed by the attribution differences in Figure 14), results may vary, and a complete removal of the artifact may not be guaranteed, because neither can be the absence of CH behaviour.

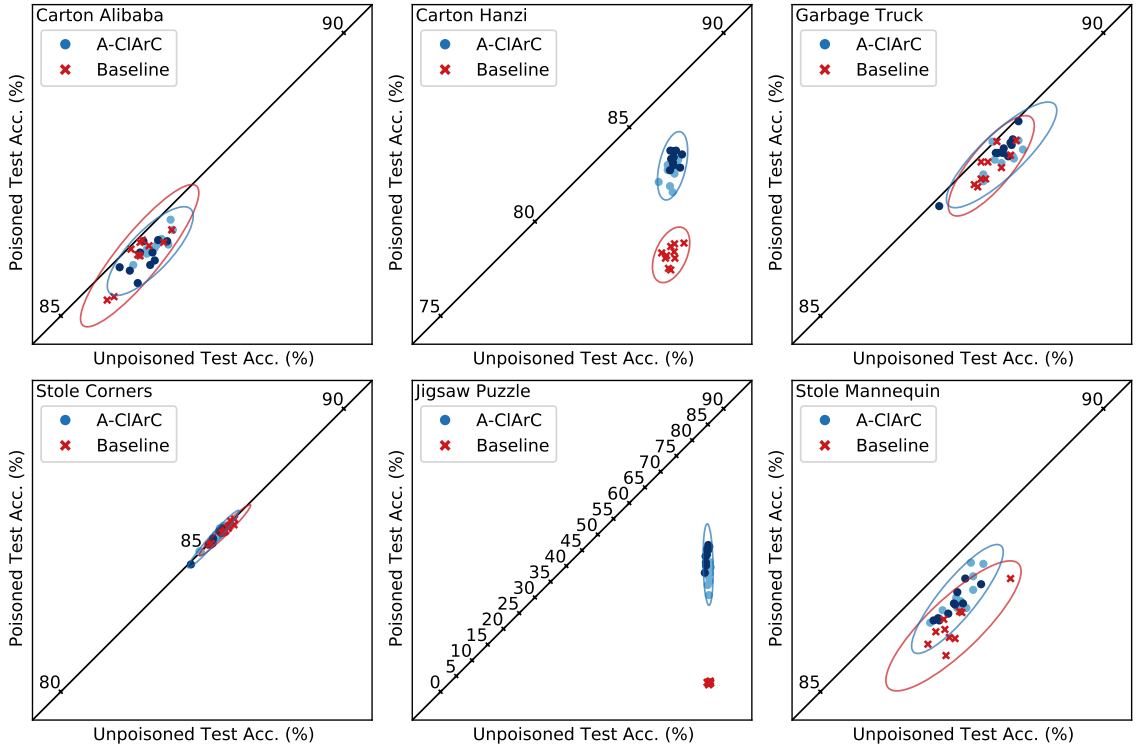


Figure 14: Unpoisoned (x-axis) vs. poisoned (y-axis) test set accuracy of baseline model (red) and A-ClArC models (blue) over training poison rate (bright 20%, dark 50%). There are 10 points per color, each representing a state of the model during fine-tuning. All points are below the diagonal line of equal accuracy on poisoned and unpoisoned test data, which means they consistently perform better on the unpoisoned test set. For "carton hanzi" and "jigsaw puzzle", the unlearned models perform significantly better on the poisoned test set than the baseline. This can also be seen less significantly for "stole mannequin". In all cases, the accuracy of the unlearned model does not visibly decline compared to the baseline model. In the case of "jigsaw puzzle", the artifact is for many samples the only class-defining feature, which therefore extremely confuses the model on the poisoned test set. The values for both axes are annotated to the diagonal line, along which the accuracies on both test sets would be equal.

4.3. *Augmentative Class Artifact Compensation in Feature Space*

Equivalently to the previous section, we may instead choose to do a fine-tuning with A-ClArC using artifact representations that we have found in the feature space of any layer of a neural network in Section 3.3. There, we noted that these intermediate representations of artifacts differ significantly in how well they can be separated via SpRAy. For each artifact, a different layer depth seems to allow for an optimal separability, and this depth seems to correlate to the complexity of the respective artifact.

Building on those observations, we conduct another experiment, similar to the one in *Augmentative Class Artifact Compensation on ImageNet*, applying A-ClArC using feature space representations of the target CH concept. In contrast to the input space variant of A-ClArC, here, the target CH concept is represented via CAVs. More concisely, each CAV is computed between samples of the respective class that contain the target CH artifact and all other samples of the same class. These two groups are identified via SpRAy in feature space, as described in Section 3.3.

Setup. We again compare an *unlearned model* (corresponding to the *a-posteriori A-ClArC* introduced previously) that is fine-tuned for 10 epochs on a subset of ILSVRC2012 consisting of 100 classes and employs A-ClArC to a *baseline model* that is fine-tuned in the same manner, but without A-ClArC. Both models are initialized from the *baseline model*, which is the same VGG-16 as for *Augmentative Class Artifact Compensation on ImageNet*. For A-ClArC, the target CH-artifact – described by a CAV, i.e., a direction in feature space – is added during fine-tuning to the activations at the respective layer l, with a probability p of 50% and a contribution of 50%. The contribution denotes in which ratio the original activations and the added CAV are mixed. This method of introducing the CAV to the activations corresponds to $a_i = 0.5\ \forall i \in \{1, 2, \cdots, d\}$ wrt. Equation 23), and is used multiple times over Section 4 (for concept desensitization and evaluation) and 5 (only for evaluation). As described in Section 2.6, the parameters of layers $\{1, \cdots, l\}$ are not altered during training, to keep the feature representation of the target concept static. Again, we employ the two test modes described previously, reporting accuracies on the original (0% poisoned) and a poisoned test set (100% poisoned). The poisoning process, however, is executed in feature space for this experiment, using the computed CAV to poison the activations at layer l instead of introducing the artifact in input space. This experiment is repeated for feature extractor layers $l \in \{0, 4, 10\}$, with layer 0 denoting the input layer.

Results. The results of this experiment are summarized in Figure 15. The four CHs shown there ("pattern", "border", "colored pattern", "mannequin head") are chosen to range from relatively simple to quite complex concepts.

At the *(bottom right)* of each panel of this figure, the test accuracy after the final epoch is visualized for the three investigated layers. Results for both previously discussed test modes are shown: To the *(left)*, i.e., for the 0% poisoning setting, we note that the unlearned model performs equally well as the baseline. The application of A-ClArC did thus not affect the model's accuracy on unpoisoned data in a negative manner, indicating that it does not confuse a model unnecessarily or introduce any unfair biases. To the *(right)*, the 100% poisoning setting is reported. Here, the unlearned model *vastly* outperforms the baseline model for every single CH example. However, its accuracy varies wrt. the layer at which A-ClArC is applied – as could be expected based on the results from Section 3.3, where we found the separability score τ of samples containing a CH artifact from clean samples to be quite dependent on the layer where SpRAy is applied. Moreover, the layer where performance is best seems to correlate to the perceived complexity of the CH. E.g., for the "pattern" artifact in class "jigsaw puzzle", which can easily be represented via an affine transformation in input space, thus being a relatively simple CH, the unlearned model performs best at the input layer. In contrast, for the far more complex "mannequin head" from the "stole" class, the highest accuracy is retained for intermediate layer 10. In fact, for all closer investigated CHs, the performance of the unlearned model at the "optimal" layer in the *poisoned* setting is almost on par with the performance of both unlearned and baseline models on the *unpoisoned* setting, demonstrating a significant gain in invariance against the concept described by the CAV after applying A-ClArC.

However, since we employ the computed CAV to poison the test data, the above evaluation only shows that invariance against the target concept is gained, if the CAV represents that concept correctly. Thus, to ascertain whether in fact the *target* concept is unlearned via A-ClArC, we interpret Figure 15 (*(top)* of each panel), where (LRP) attribution difference maps of one example image per class are shown over epochs for the respectively best performing layer. For reference, the original image of each of these examples and the corresponding LRP attribution map of the baseline model can be found to the *(bottom left)* of each panel. In the attribution difference maps, *magenta* colors indicate areas that lost relevance compared to the baseline model, while *teal* colors show a gain in relevance. The

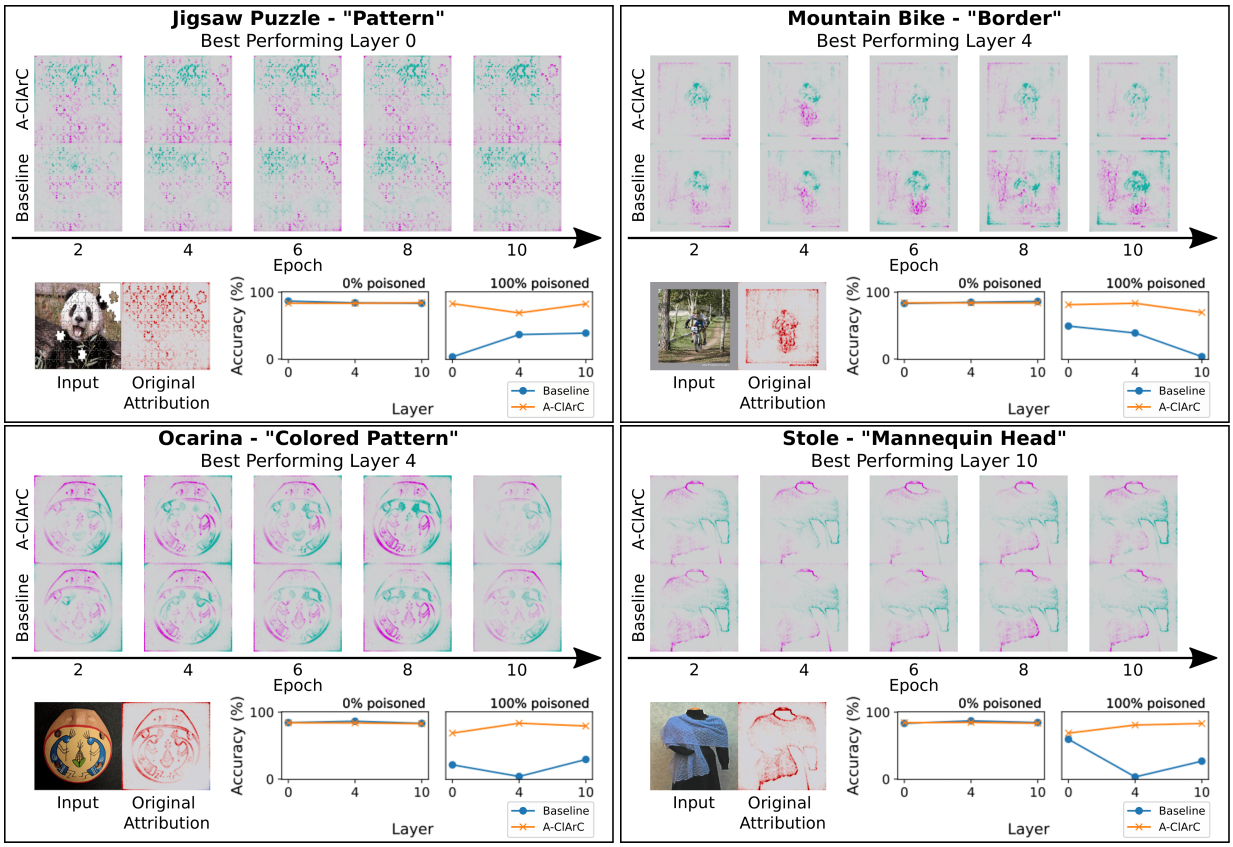


Figure 15: A-ClArC in feature space for four example CHs. *(Bottom right)* of each panel: accuracy of the baseline that was fine-tuned without A-ClArC and the unlearned model that employed A-ClArC on progressively more poisoned data. The poisoning is performed by adding the CAV computed for A-ClArC to the activations after the respective layer. Each CAV is computed within a single class, between samples containing the target CH and clean samples, as clustered by SpRAy. The A-ClArC model vastly outperforms the baseline with increasing poison rate, showing that it has grown relatively invariant to the target CH concept. *(Bottom left)* of each panel: example image and corresponding attribution (LRP), where *red* areas indicate a positive relevance towards the model's prediction, and *blue* areas a negative relevance. *(Top)* of each panel: LRP relevance differences (see Eq. (32)) between the baseline model that does not employ A-ClArC and the original model (i.e., initial starting model before training), as well as between the A-ClArC-using model and the original model are shown over training for the respective best performing layer, i.e., the layer where the accuracy of the A-ClArC model in a poisoned setting is largest. Here, the *magenta* areas of the attribution maps are used *less* than the original model for prediction; the *teal* areas *more*. Refer to Supplementary Figure B.1 for a more detailed color description.

attribution difference maps were computed as specified in Eq. (32). Each *(top)* row of attribution difference maps emphasizes these changes for the unlearned model that employs A-ClArC, while each *(bottom)* row does the same for the baseline instead.

For class "jigsaw", at the input layer, we note that the model employing A-ClArC is able to successfully reduce the relevance of exactly the target artifact, consisting mainly of three distinct puzzle piece shapes in the upper right as well as the bottom left of the image, while gaining relevance on the panda's head. The baseline model, in contrast, slightly reduces the relevance of the upper-right puzzle piece (although not as significantly as the unlearning model does), however, it barely has an effect on the lower-left puzzle pieces.

For the "border" artifact in the "mountain bike" class, we find a similar behavior, with the unlearned model precisely reducing the relevance of the "border", while simultaneously putting more emphasis on the desired features of the mountain bike and its driver. However, here we additionally observe another interesting effect: The unlearned model is relatively stable in terms of which features receive more or less relevance over the course of fine-tuning, instead only varying in *intensity*, not *locality*, pointing to a goal-oriented behavior. The same is not true for the baseline model, on the other hand, which seems to vary wrt. both.

While this observation with regards to training stability is also confirmed for the "ocarina" class, neither the

A-ClArC model nor the baseline manage to correctly decrease relevance for the full "colored pattern" artifact. It seems that the computed CAV representation for that artifact may not sufficiently capture the artifact direction in this instance. This could be either due to the high variability in terms of how this artifact appears for different samples, or because of the examples offered for computing the CAV vector not describing the target CH precisely enough.

For the "mannequin head" concept, however, the correct concept seems to be unlearned by the A-ClArC model, and with high stability. On a first glance, the baseline model seems to reduce relevance of similar features as the unlearned model does. But, when inspecting this more closely, we find that the A-ClArC model actually reduces the relevance of the "mannequin head" with higher precision and more completely – and simultaneously loses less relevance on actually desirable features, i.e., the lower part of the blue stole.

Discussion. Although there are cases where the unlearning in feature space via A-ClArC is not successful – for instance due to the computed CAV not representing the correct concept –, generally, it performs extremely well, gaining significant invariance against a target concept. Moreover, the method performs in an extremely stable manner, showing improvements in comparison to a baseline model both quantitatively and qualitatively. We were further able to confirm our findings from Section 3.3 again, demonstrating a connection between artifact complexity and the layer at which it can be unlearned with the best results.

However, the application of A-ClArC still requires tedious and time-consuming fine-tuning. In contrast, the second proposed method for concept removal – P-ClArC – is far more efficient in that respect. Keep in mind, though, that – in contrast to A-ClArC – P-ClArC does not perform true *unlearning* in that sense, since it does not allow the network an opportunity to adapt its weights, and instead rather *suppresses* artifacts. Due to its promising properties with regards to efficiency, the following experiments will be dedicated to evaluating the P-ClArC method – and whether it can keep up with A-ClArC in performance.

5. Experiments – Concept Desensitization using Projective Class Artifact Compensation

After the identification several CH type artifacts used by models trained on the ILSVRC2012 dataset (see Sections 3.2 and 3.3), we have successfully demonstrated the removal of their influence on the model in the previous paragraphs, using A-ClArC. However, as A-ClArC requires the model to be fine tuned, it is not very efficient and might even become tedious in an iterative artifact identification and removal process. The P-ClArC-method proposed in Section 2.6, on the other hand, does not require any further training after the modeling of the artifact, but conversely does not allow the model to adapt its weights and strictly *unlearn* – as A-ClArC does. Instead, it acts as a filter and removes a concept's contribution to the output. Whether the concept suppression of P-ClArC is successful and comparable to A-ClArC is evaluated experimentally in the following paragraphs.

First, we measure the performance of P-ClArC in a toy setting on ColoredMNIST, before proceeding to the more complex ILSVRC2012 domain. Finally, in Sections 5.3 and 5.4 we touch upon the subjects of fairness and reliability in machine learning by showing that P-ClArC is able to increase the robustness in the prediction of biased real-world datasets, i.e., the ISIC 2019 dataset in a skin lesion classification setting, and the Adience face classification dataset with a DNN trained to predict biological gender.

In terms of fairness, we consider the notion of counterfactual fairness [92], where a predictor is fair regarding some protected attribute A, if its decision is not affected when A is flipped to its counterfactual value. In the context of this work, A refers to specific CH artifacts that lead to biased decisions. However, finding counterfactual values is not trivial for these complex features, in contrast to categorical attributes such as gender or race. For this reason, we relax above fairness notion during our empirical investigations and assume that a model becomes fairer wrt. a specific CH, if it relies less on that CH to make predictions, i.e., if it is desensitized to that CH concept.

5.1. *Projective Class Artifact Compensation on Colored MNIST*

Setup. To assess the validity of the proposed P-ClArC method, we first apply the method in a toy setting with relatively simple (CH type) concepts in the dataset. More concisely, we add color-based CH artifacts to the MNIST training dataset [33, 34], with one designated color for each class (see Figure 6 for the specific colors), distinctly changing the tint of 20% of the samples. While simple, the resulting concept is complex enough as to not have a pixel-wisely localizable representation in input space. We train a simple convolutional network as described in Appendix A.

For one color concept and intermediate layer l at a time, we "unlearn" the target concept without re-training by using P-ClArC. *Meaningful CAVs* are obtained using poisoned and unpoisoned training samples from within each respective class, as described in Section 2.6. We then evaluate the success of this unlearning procedure using an altered (or poisoned) test set: here, the target concept color is applied to a certain percentage of samples from the (whole) test set. We then evaluate and compare the performance of the original model to the performance of the model desensitized to the color concept via P-ClArC on this poisoned test data, as shown in Figure 16 *(top)*. The accuracy *(y-axis)* of the original model (*blue*) and the corrected model (*orange*) is compared for the poison rates 0% (uncolored MNIST), 50%, and 100% *(left to right)*, averaged over all ten classes. This comparison is visualized for the input layer and the first convolutional layer of the feature extractor *(x-axis)*.

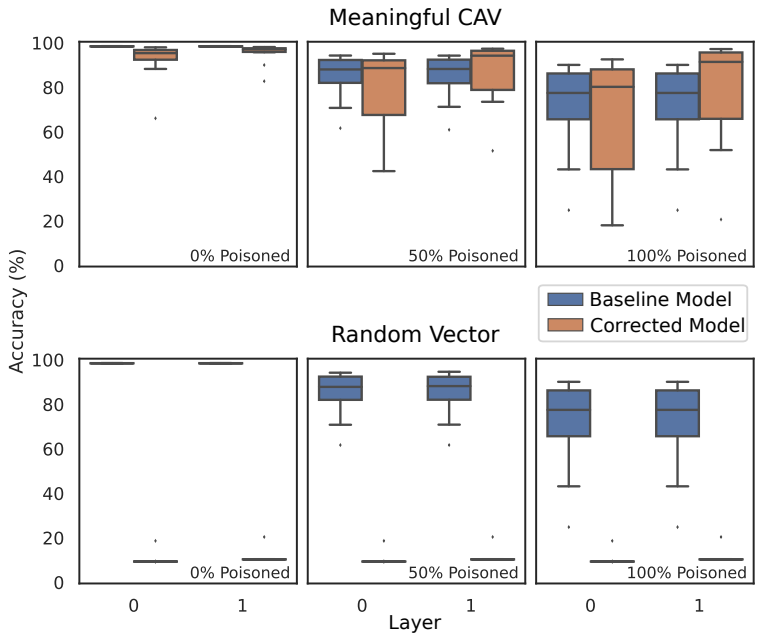


Figure 16: P-ClArC on colored MNIST with 0 *(left)*, 50 *(middle)* and 100 *(right)* percent test set poisoning. For the evaluation, the CH Artifact is added in the input space. The accuracy of a *baseline* model *(blue)* and an *corrected* model employing P-ClArC *(orange)* on these datasets is compared for CAVs obtained after the input layer (0^{th} layer) and the first convolutional layer (1^{st} layer). Measurements are taken from the separate unlearning of all CH artifacts in the colored MNIST dataset. In the *(top)* row, the corrected model uses meaningful CAVs that are computed from two distinct sets of data samples, as described in Section 2.6. Each meaningful CAV is computed within a single class, between poisoned and unpoisoned samples. In contrast, a random vector is utilized instead for the *(bottom)* row. While the meaningful CAV leads to an improvement of the corrected model over the baseline for the poisoned datasets, the random vector has an extremely detrimental effect on model performance in every case. The high variance in accuracy of the P-ClArC method in the *(top)* row is explained by the CAVs being applied to the complete test set for above evaluation, even though each one was only computed from samples within a single class, due to the class-specific nature of CH artifacts.

Results. With increasing dataset poisoning, the model to which P-ClArC is applied outperforms the baseline model. However, the accuracy of both models on average decreases slightly with higher poison rates, showing that while P-ClArC makes the model more robust against the CH artifact specifically, the model is not completely unaffected otherwise. Note however, the high variance of the performance after applying P-ClArC in the *(top)* row: CH artifacts are extremely class-specific, since they only necessarily behave in a biasing manner *in conjunction* with valid class features, and therefore a CAV that is able to distinguish between CH and non-CH samples needs to be computed from the samples of a single class (otherwise the CAV might end up as a detector for the CH-afflicted *class*, instead of the CH *artifact*). Thus, above evaluation scheme may suffer from generalization issues of that CAV when applied to other classes, and layers not encoding the CH distinctively enough, explaining the high variance. As a sanity check, we further perform the same evaluation using *randomly generated CAVs*, as shown in Figure 16 *(bottom)*. As expected, the model to which P-ClArC was applied does not outperform the baseline model in this instance, and instead only achieves a considerably lower accuracy due to the arbitrary and not data-specific projection of the

features. In combination with Figure 16 *(top)*, this shows not only that the computed CAVs describe the targeted color concept in a meaningful manner, but also that the proposed P-ClArC method is able to exploit the CAV representation successfully in order to make a model more robust against the target concept.

Evaluation with the CAV-Predictor. The above method of evaluation, however, again requires the addition of concepts in input space (since the colors are introduced to the test samples in input space) and may thus not be suitable for arbitrary (especially more complex) concepts. Especially "naturally occurring" artifacts known (and in this paper discovered) to appear in various popular datasets, e.g., CH artifacts like the mannequin head in ILSVRC2012 [9], colored band-aids in ISIC 2019 [30, 31, 32], or shirt collars in Adience [35] do often not have a singular, pixel-wise representation in input space, and, as such, the performance of P-ClArC on these artifacts would be difficult to assert using the above method of poisoning data in input space. Thus, we propose the following alternative: as previously established, CAVs offer a representation of a concept in feature space. Instead of altering test samples in input space, we can thus poison the test data by adding the CAV corresponding to a target concept to latent activations of a certain percentage of samples at layer l during inference, and again compare the predictions of the model before and after applying P-ClArC. As such, this evaluation is not restricted to the input space. Its validity is, however, dependent on whether the obtained CAV actually denotes the correct concept. Therefore, we also aim to validate whether the CAV correctly describes the targeted concept: To this end, we discard all network layers after layer l, and model the network output with the CAV classifier receiving its inputs from layer l. We thus obtain a network that classifies for a given input sample, whether it contains the concept described by the CAV, or not. In the following, this network is called *CAV-predictor*. After applying LRP to this CAV classification network, the resulting relevance maps can be evaluated in terms of whether they correspond to the expected target concept.

CAV-Predictor Results. With the second proposed method of evaluation shown in Figure 17 *(top)*, both models decrease in accuracy relatively, especially for higher rates of dataset poisoning and in comparison to Figure 16. However, the P-ClArC-corrected model significantly outperforms the baseline on the poisoned test dataset, and more so when the artifact has been modeled after latent feature representations. Furthermore, the CAVs seem to describe their respective color artifact with high precision: In Figure 17 *(bottom)*, the distribution of LRP-relevances for the CAV-predictor is visualized across the three color channels, for classes "0" *(left)* and "5" *(right)*, and, respectively, colors blue and orange. Higher relevance is mostly attributed to the color channels that describe the target color concept. E.g., for the "blue" artifact, high relevance is attributed equally to the green and red channels, and less to the blue channel: Due to the additive rgb color system, red and green are the altered channels when introducing a blue artifact. In addition, Figure 17 *(bottom)* shows the absolute amount of relevance attributed to each color channel, confirming that the CAV indeed describes the target CH. However, as indicated in both parts of Figure 17, the disentaglement of benign and artifactual features seems to work better for layer 0 than for layer 1, implying that the CAV encodes the coloring more precisely there. Apparently the coloring is in fact a relatively simple (i.e., static wrt. its embedding into the input dimensions) CH, that is still most accurately represented in input space.

5.2. Projective Class Artifact Compensation on ImageNet

With the above toy example showing promising results, we further apply and evaluate P-ClArC in the more complex setting of ILSVRC2012, where various CH-type artifacts were identified using SpRAy, as described in Sections 3.2 to 3.3.

Setup. Equivalently to the corresponding experiments with A-ClArC in Sections 4.2 and 4.3, we use the VGG-16 model with the pretrained weights obtained from the Pytorch model zoo. P-ClArC is performed at layers 0, 4, and 10 of the model's convolutional feature extractor in separate experiments. We evaluate on a subset of 100 (randomly chosen) ILSVRC2012 classes that include the class where a CH occurs in the data (called "target class" in the following). Again we compare a *corrected* model that employs P-ClArC to a *baseline* model that does not. For this purpose, we use an unpoisoned and a poisoned test dataset, with the latter being augmented by adding the CAV that encodes the target CH to the activations of all samples at the respective layer (100% poisoning). Similar to the feature space experiments in Section 4.3, each CAV is computed within a single class only, between the samples that contain the target CH and those that do not, as clustered via SpRAy. To assert how well P-ClArC suppresses the target CH concept, we again employ the previously established twofold evaluation method that does not rely on the introduction

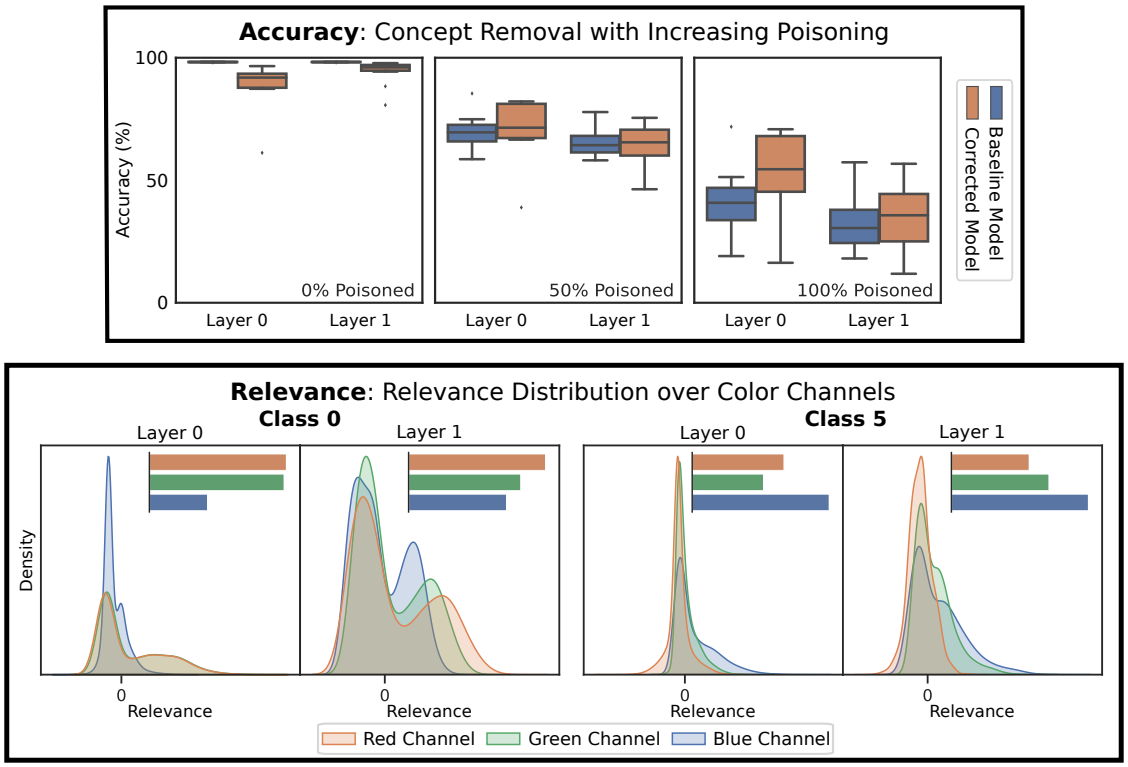


Figure 17: P-ClArC on colored MNIST with the artifact added in feature space as a CAV during evaluation (as opposed to Figure 16, where the artifact is added in input space). *(Top):* evaluation of performance with 0 *(left)*, 50 *(middle)* and 100 *(right)* percent test set poisoning. The accuracy of a *baseline* model *(blue)* and a *corrected* model using P-ClArC is compared over all classes for layers 0 and 1. Each CAV is computed within a single class, between poisoned and unpoisoned samples. The corrected model outperforms the baseline model. *(Bottom):* validation of the concept that is described by the computed CAV, exemplary for the classes 0 *(left)* and 5 *(right)*, with the introduced CH concepts blue and orange, respectively. Here, the distribution of (LRP-) relevance over color channels is shown for the CAV-predictor, indicating which color channels are most indicative of the CAV-direction (i.e., when interpreting the computed CAV as a classifier for CH and non-CH samples, the channels that have the most impact on making that decision). The bar plots above show the sum of (unsigned) relevances across color channels. The CAV-predictor assigns most relevance to the color channels that differentiate the poisoned samples from the (white) original samples (e.g., red and green for the blue artifact in class 0, and blue for the orange artifact in class 5).

of artifacts in input space, combining a quantitative comparison between the two models' outputs with a qualitative analysis of the difference in attributed relevances. The CH artifacts that are inspected more closely were identified using SpRAy and range from simple artifacts with static placement in pixel space (e.g., laptop - "lid") to relatively complex conceptual and non-static concepts (e.g., swimming trunks - "upper body").

Since dataset poisoning for the purpose of evaluation is achieved by adding the computed CAV to activations at the respective intermediate layer, it is not sufficient to show that P-ClArC successfully counteracts this, since the same CAV is used in its projection step. Rather, we first need to establish that the CAV actually encodes a meaningful feature of the target class. Furthermore, to be valid, P-ClArC should be concept-specific, and thus optimally not have any effect on the network's inference for samples that do not contain the target artifact.

Quantitative Results. The results of a quantitative analysis of these three properties is shown in Figure 18 for the classes "laptop" and "stole", with the CHs "lid" and "mannequin head", as examples for a relatively simple and a more complex CH, respectively. Note that in this figure, class-wise (normalized) logits are visualized as opposed to the final softmax probabilities, since slight changes may not be easily registered in the latter, due to the high number (1000) of classes in the ILSVRC2012 dataset and thus the model's output. However, as the model is originally trained to optimize softmax probabilities, it is sufficient to only compare the relative relationship between classes outputs due to the shift invariance of the softmax function. As shown on the *(left)* side of Figure 18, P-ClArC preserves a model's performance if applied to unpoisoned data. Note that since the 100 test classes also contain the target class, a very

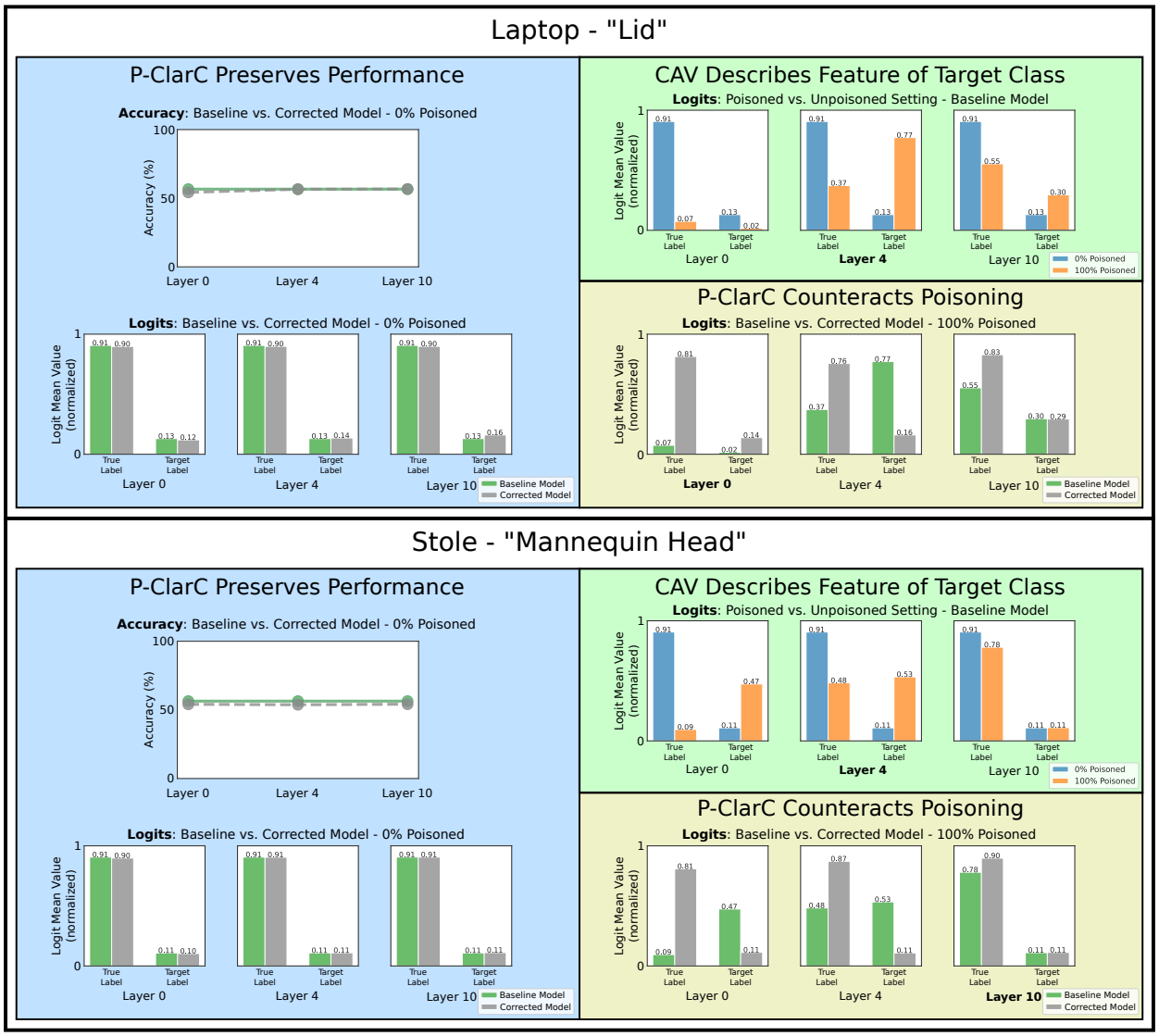


Figure 18: Quantitative Evaluation of P-ClArC. As examples for a simple CH and a complex CH, laptop - "lid" *(top)* and stole - "mannequin head" *(bottom)* are shown, respectively. The mean logit values in this figure were normalized sample-wise by dividing by the largest absolute logit value and thus constrained to the interval $[-1, 1]$ for a relative evaluation. Results are depicted for layers 0, 4, and 10 of the convolutional feature extractor of the model. The CAVs used in this experiment are each computed within a single class, between CH and non-CH samples, as clustered via SpRAy. The evaluation is threefold: (1) Performance is preserved when applying P-ClArC to unpoisoned data (*(left)* of each panel), as both the accuracy nor the logits of the true and target class barely vary between baseline and corrected model. (2) When using the computed CAV to poison activations additively, the ratio between the true class and target class logits diminishes (*(top right)* of each panel), indicating that the feature encoded by the computed CAV is specific to the target class. (3) P-ClArC successfully removes a target concept, encoded by the computed CAV, since it moves the logits closer to the ground truth, increasing values for the true label, while reducing them for the target label (*(bottom right)* of each panel). For (2) and (3), the layer where the respectively desired effect is best observable is marked bold (although it is not necessarily only observable there).

slight change in performance can be found for, e.g., class "laptop" at Layer 0 or the class "stole". However, the mean logit values never vary by more than 0.03 between the baseline and corrected model, with the ratio of true label logits and target label logits barely changing.

Since the projection step of P-ClArC relies on the computed CAV precisely representing the targeted CH concept, we next assert whether the CAV is meaningful wrt. the target class, i.e., whether it describes a feature specific to

the target class. We do this by observing how the mean logit values of the true and target class labels change when the model's inference process is poisoned by adding the computed CAV to the activations of the respective layer, thereby shifting these activations in the CH direction – as it is described by the CAV. The corresponding results are demonstrated in the *(top right)* of each panel of Figure 18. Here, we note a significant decrease in the mean logit values of the true class labels when poisoning the test data. At the same time, the mean logit values of the target labels mostly increase, e.g., class for "stole" in layer 4, where the true label logit mean value diminishes from 0.91 to 0.48 due to poisoning, while for the target label it increases from 0.11 to 0.53 at the same time. An exception is the class "laptop" at layer 0, where values decrease for both (sets of) classes. However, the *ratio* between true label and target label logits always changes in favor of the target label with poisoning. We thus deduce that since adding the computed CAV to the activations at the respective layer relatively increases the model's confidence of the target class over the true class, the CAV encodes for a feature that is specific to the target class. Note that this does not necessarily imply that the CAV describes the exact target CH concept, which is an observation that we investigate further in Figure 19 a few paragraphs further.

In the poisoned setting, the baseline model consistently assigns a larger logit value to the target class than in the unpoisoned setting, while the opposite occurs for the true class. Observed exceptions to this rule are layer 0 for class "laptop" and layer 10 for class "stole" (cf. the *(bottom right)* parts of the panels in Figure 18). This can be explained by the relative complexity of the respective artifacts and their (attempted) point of encoding in the network. Artifacts best expressed statically in pixel space (here, the laptop's lid) are more readily encoded by a CAV trained here, compared to later layers, where the model has developed invariances against pixel-specific encodings. Conversely, more complex and semantic concepts such as the mannequin head, which as a feature appear in multiple locations and poses over the dataset are more readily encoded in invariant latent representations later in the model.

When employing P-ClArC, the model manages to correct this skewed distribution successfully, and assigns larger logit values to the true class than to the target class. E.g., for layer 4 of class "stole", the baseline model infers a normalized logit mean value of 0.48 for the true class, but 0.53 for the target class. The corrected model, however, shifts this distribution in favor of the true class by projecting the activations beyond the CAV-predictor's hyperplane, assigning a normalized logit mean value of 0.87 to the true class and 0.11 to the target class. Note that in this example, the previous *unpoisoned* mean logit values (*(top right)* of each panel, *blue*) are almost perfectly restored. Thereby, it successfully counteracts the introduced poisoning.

An Indicator for Desensitized Features. P-ClArC projects samples beyond the hyperplane separating samples within a class that contain a target CH concept and samples that do not. Thus, its performance is entirely dependent on how well the learned CAV, i.e., the vector orthogonal to that hyperplane, describes the target CH. The quantitative analysis of Figure 18, however, is only able to assert that the CAV describes *some* feature specific to the target class of which the influence on the model's inference can be increased by adding it to the activation in feature space (and consecutively decreased again applying P-ClArC). Up to this point, however, we have not yet shown that the computed CAV describes *exactly* the target CH feature or that P-ClArC is able to successfully remove a CH concept that is not added artificially, but naturally occurs in the data.

For these two purposes, relevance maps computed via LRP are shown in Figure 19, with each panel dedicated to one specific CH concept. These CHs range (in the order of *(left)* to *(right)*, and *(top)* to *(bottom)*) from simple artifacts that are present in roughly the same pixels of each affected sample (e.g., laptop - "lid") to far more complex features (e.g., swimming trunks - "upper body") that present differently in each sample and can thus not be described in input space in a uniform manner. For three example images of the each CHs, and for the models's intermediate layers 4 and 10, (1) relevance maps of the CAV-predictor *(left relevance map for each layer)* and (2) the difference in relevance attribution *(right relevance map for each layer)* between the baseline model $R_{baseline}$ and the P-ClArC-employing corrected model $R_{P-ClArC}$ (Computed similarly to Eq. (32), as $R^{diff}_{P-ClArC} = R_{baseline} - R_{P-ClArC}$ are shown.

The former (1) visualizes which features are important to the decision hyperplane of the linear CAV-predictor in classifying whether a sample contains a CH or not, with *red* highlighting positive relevance, and *blue* negative relevance, and thereby offers an estimation of how well the computed CAV encodes the correct concept. The latter (2), on the other hand, is computed similarly to Eq. (32), as $R^{diff}_{P-ClArC} = R_{baseline} - R_{P-ClArC}$ shows how the importance of features for a certain decision changes when employing P-ClArC, with features that are more relevant to the prediction after applying P-ClArC in *teal*, and features whose relevance is reduced in *magenta*, and thus indicates how successfully the target concept's influence on the model's prediction is removed by P-ClArC.

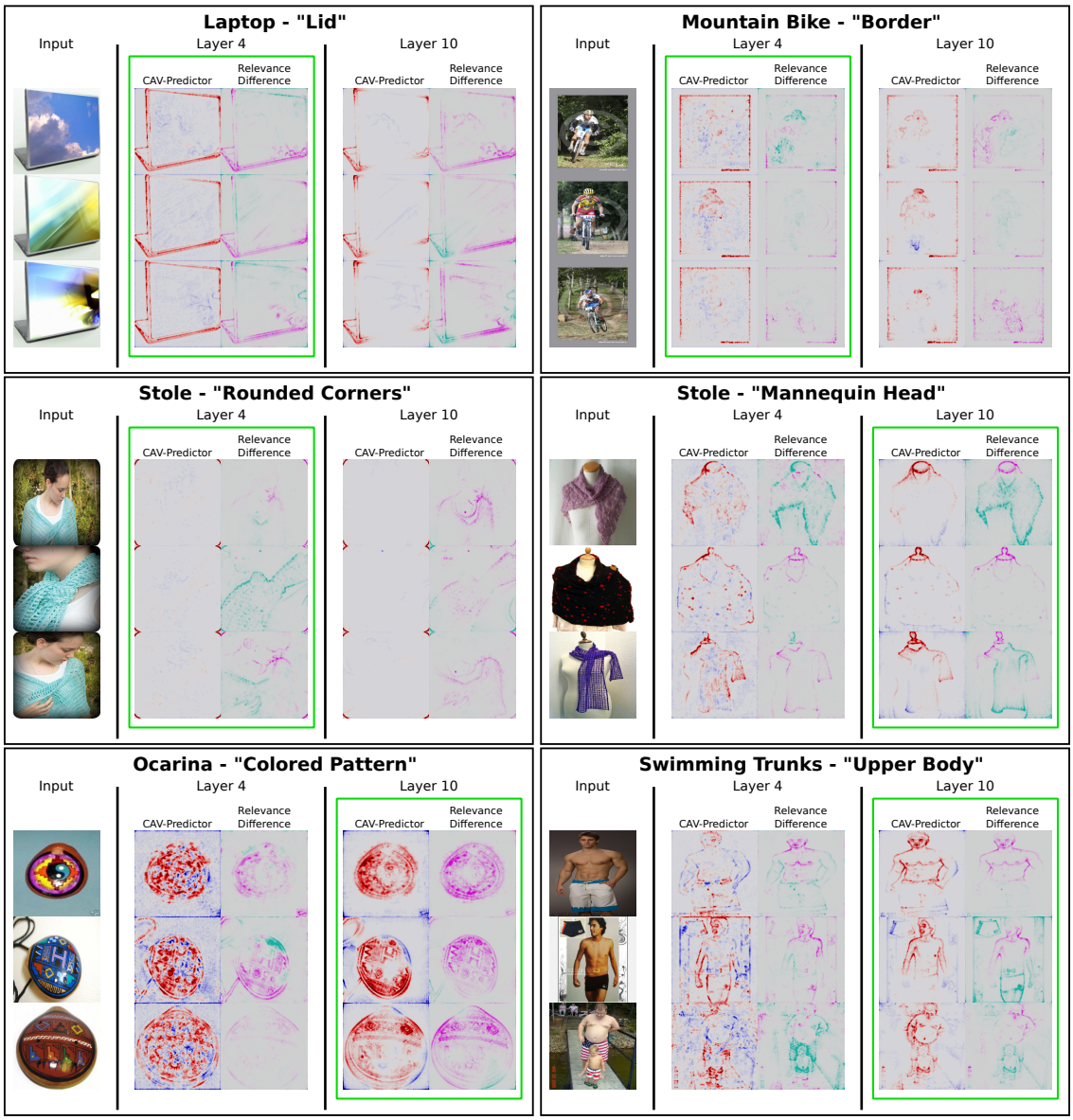


Figure 19: Effects of P-ClArC on ILSVRC2012. In every panel, P-ClArC was applied after layers 4 and 10. For each of these, the LRP relevances of the CAV-predictor *(left relevance map for each layer)* and the relevance difference *(right relevance map for each layer)* between the *baseline* and the *corrected* model is visualized. For the CAV-predictor heatmaps, *red* areas highlight the features that are indicative for the CAV-direction. In the relevance difference images, the corrected model focused less on the areas highlighted in *magenta* compared to the baseline model, but more on the parts highlighted *teal*. Refer to Supplementary Figure B.1 for a more detailed color description. The relevance differences are computed similarly to Eq. (32), as $R^{diff}_{P-ClArC} = R_{baseline} - R_{P-ClArC}$. The CAVs are each computed within a single class, between CH and non-CH samples, as clustered via SpRAy. While the first three CHs ("lid", "border", "rounded corners") occupy the same pixels between samples, the last three CHs ("mannequin head", "colored pattern", "upper body") consist of more complex features. In line with this complexity, P-ClArC seems to perform better on earlier layers of the feature extractor for the first group, with the heatmaps corresponding more to the target concept, as indicated by the *green* border. The opposite seems to be the case for the second group, concurring with the separability scores in Figure 11.

Qualitative Results. For all CH examples, the CAV seems to correctly encode the targeted artifact, as the correct features are used to identify them as containing the CH. However, there are notable variations in the precision of the CAV-predictor on the correct features between CHs and – for the same CH – between layers. E.g., for layer 4 of the laptop - "lid" artifact, the outline of a laptop backside, digitally rendered from a specific angle by the image creator,

is clearly visible. However, for layer 10, only the corners of the same outline seem to be relevant, indicating that the artifact encoded by the CAV is less precise and complete. A similar trend can be observed for other relatively simple CHs, i.e., mountain bike - "border". For stole - "rounded corners" the CAV seems to be on point for both layers. In contrast, the CAV-predictor for the more complex stole - "mannequin head", seems to focus on the mannequin head artifact as well as the correct class features of the stole itself in layer 4, however, in layer 10, it seems to single out the mannequin head artifact almost exclusively. A similar effect occurs with swimming trunks - "upper body", where the layer 4 heatmaps are relatively diffuse, while for layer 10 only the human upper bodies are assigned a large positive relevance.

In a similar manner, the relevance difference heatmaps show that the artifact is less impactful on the model's decision-making after applying P-ClArC. Again, the success of this artifact removal varies with the specific CH and layer, and this variation seems to behave in the same way as for the CAV-predictor heatmaps, as described above, although some differences exist. E.g., for layer 4 of the laptop - "lid" artifact, where the CAV-predictor seems to be most precisely learned, the corrected model assigns far less relevance to the outline of the laptop's lid. At layer 10, suddenly also parts of the image imprinted on the lid are removed. Mountain bike - "border" behaves in a similar manner, however, for stole - "rounded corners", while both CAV-predictor and relevance difference heatmaps somewhat coincide at layer 4, with the corners being removed correctly, at layer 10 mainly the blue stole itself receives less relevance, and relevance on the rounded corners actually increases. This makes sense, as the rounded corners are an extremely simple artifact – thereby being removed more successfully in earlier layers, in accordance with our previous findings. However, it also seems that just because the CAV seems to describe the artifact correctly, the unlearning result does not always exactly correspond to that. Note, that since these heatmaps are normalized wrt. the largest absolute relevance value, the small rounded corners may only be assigned an extremely large relevance, while the larger blue stole receives smaller relevance - therefore not being visible in the CAV-predictor relevance maps. In fact, this example further showcases another interesting problem: The samples of the class stole that contain the "rounded corners" artifact also always contain the same person and the same blue stole. The CH is thus ill defined here, since the "rounded corners" cannot be described by only using example images, making CAVs apparently not the ideal choice of representation for this specific artifact. Since the resulting CAV would encode both, in a way, this is thus both a simple and a complex CH, with P-ClArC removing the simple part ("rounded corners") at the earlier layer, and the more complex "blue stole" at the later layer.

Matching these interpretations, the complex "mannequin head", "colored pattern", and "upper body" artifacts are removed far more successfully at layer 10. Note especially the class "swimming trunks", where not only the relevance of the upper body decreases, but also relevance on the swimming trunks themselves is increased. The same effect is also visible for the "mannequin head" artifact.

Discussion. To summarize, there seems to be an intermediate layer where the computed CAVs not only encode the correct and intended CH concept – although this layer differs for each respective artifact. The CH correction is also more precise at the same layer, not only leading to a lessened impact of the targeted artifact on the model's prediction, but also often an *increase* of the correct non-CH class features. In fact, this layer largely coincides with the complexity of the targeted artifact, confirming expectations and our findings from Section 3.3. Although, we observe that in comparison to Section 3.3, the best performing layers are shifted backwards in the network, e.g., for the "lid" CH, this optimum seems to be at layer 4 instead of layer 0 when applying P-ClArC, possibly due to exploitation of the model's feature space representation at later layers being more invariant. Further taking the results of Figure 18 into account, where we showed how P-ClArC not only counteracts poisoning and shifts the prediction towards the true class, but also does not affect performance on unpoisoned data in a significant manner, we thus surmise that P-ClArC is an efficient but powerful tool for concept removal. Note, however, that P-ClArC will not lead to an increased generalization performance, since the model never has a chance to adapt its weights for learning other features and thus correct its faulty prediction reasoning. Nevertheless, its strengths lie in its ability to offer a fairer estimation of a model's generalization performance, untainted by features that should not contribute to the decision-making.

P-ClArC is able to successfully reduce the impact of CH artifacts on a model's prediction, and employing it on ILSVRC2012 is able to demonstrate that fact. Although, this dataset may not be sufficient for showcasing how powerful P-ClArC can be towards the solution of some pressing problems hindering the application of ML-methods in real-world scenarios. For this reason, the following paragraphs will offer two examples, where P-ClArC is employed to avoid predictions for the wrong reasons with dangerous consequences, and to increase classification fairness on

biased data.

5.3. Unlearning with Projective Class Artifact Compensation on ISIC 2019

In the previous section, we have confirmed the success of P-ClArC applications on toy examples and more complex settings on real photographic images, i.e., the ILSVRC2012 dataset. In this (and the following) section, we will apply P-ClArC to more domain specific datasets in order to solve practically relevant issues. Here, we demonstrate that P-ClArC can be used to increase the trustworthiness of models trained for skin lesion classification on the ISIC 2019 dataset. As it common practice, we fine-tune a neural network (here a VGG-16 model) pretrained on ILSVRC2012 on the ISIC 2019 [30, 31, 32] skin lesion classification dataset for 100 epochs, using the weights from the Pytorch model zoo for initialization. Due to ISIC 2019 not having a pre-defined labeled test set, 10% of the original training set were split off instead to evaluate its performance. Our model achieves a final test accuracy of 82.15%.

It is known, however, that the ISIC 2019 dataset contains several issues and confounders. First and foremost, a significant data artifact, that only occurs in the largest class, i.e. colorful band-aids next to the photographed skin alteration. Since this artifact is again limited to one class, it constitutes a CH-type artifact. For the purpose of skin lesion classification, aimed to be applied in the medical field to assist medical personnel or allow mobile diagnoses [93], CHs like these can have serious consequences, as they may easily lead to a misclassification, affecting the resulting diagnosis, and, as such, the life of a patient. Especially, since the affected class, "melanocytic nevus", is a benign form of skin alteration, possibly leading to fatal false negatives in terms of skin cancer diagnosis.

With this in mind, we aim to mitigate the effect that the "colorful band-aids" CH has on the model's prediction by employing P-ClArC. Only a singular CAV is computed for this experiment, from the CH-affected "melanocytic nevus" class, between all samples containing the colorful band-aids, and all clean samples. We again compare the model which P-ClArC is applied on and the original model in terms of predictions and LRP relevance maps.

Quantitative Results. Results are shown in Figure 20. As opposed to the corresponding evaluations for ILSVRC2012 (Figure 18) where normalized mean *logit* values were considered (due to the high number of classes), we here measure the more stabilized mean *softmax* probabilities, since ISIC 2019 only contains 9 distinct classes, whereas ILSVRC2012 contains 1000. Due to the missing test set labels, the (whole) training set is used for the quantitative evaluations in panels *(top left)* and *(bottom left)* of this figure. However, since the application of P-ClArC does not contain any further training, the model never has the opportunity to adapt to the performed alterations in any way, e.g., by shifting its inference strategy to features which prior to CH removal had a merely supporting function.

In Figure 20 *(top left)*, for layers 0 (i.e., the input layer), 4, and 10, the effect of adding the CAV computed (for later usage during P-ClArC) to the activations at the respective layer is measured. If the CAV encodes a feature that is specific to the target class "melanocytic nevus", one would expect the softmax probability of that class to increase when poisoning the samples in that way, while confidence in the actual true class label would decrease simultaneously. Note that due to the true class changing from sample to sample, the sum of (mean) true class and target class probabilities may exceed 1 in this figure. For layer 0, we observe a decrease both for true and target labels, indicating a generally confusing effect of the CAV-poisoning on the model, as could be expected to some degree: In input space, the encoding of CHs via CAVs may not be feasible, because the data is too complex in its raw form, and no invariant representation learned by the model has been applied yet. In contrast, for layer 4 and even more so for layer 10, the softmax probabilities exhibit precisely the expected effect. E.g., for layer 10, they change from 0.97 to 0.11 for the true class, but rise from 0.51 to 0.94 for class "melanocytic nevus". As indicated by the *green border*, this effect is most prominent in layer 10. We thus infer that the computed CAV indeed denotes a concept specific to the target class – at least for layers 4 and 10.

Building on that assertion, the next step is to validate whether the P-ClArC method is able to counteract said poisoning. Figure 20 *(right)* shows the corresponding results. The inference results of the *baseline* model and the model employing P-ClArC are compared in the form of *mean softmax probabilities*. With the data poisoned in the same manner as in Figure 20 *(top left)*, not only should confidence in the target class decrease with a successful removal of an artifact, but also confidence of the true class should increase, restoring the predicted probabilities of an unpoisoned setting as closely as possible. As visible throughout Figure 20 *(top left)* to *(bottom left)*, this is barely the case for layer 0, partly due to the probabilities already decreasing both for the target and the true class because of the poisoning. Even so, P-ClArC manages to almost restore the original confidences, with the true label probability growing from 0.38 to 0.83 (unpoisoned 0.97) and target label probability from 0.00 to 0.54 (unpoisoned

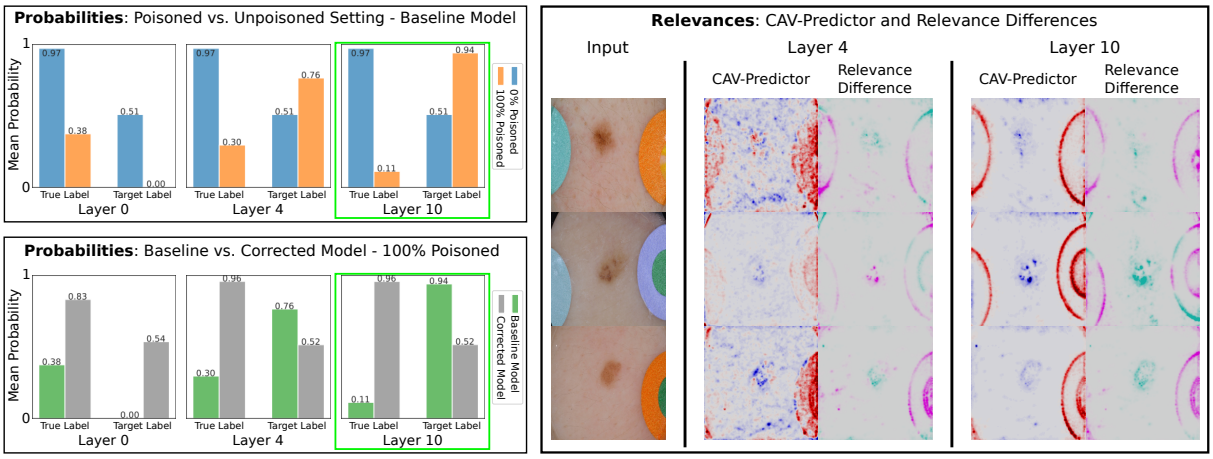


Figure 20: Employing P-ClArC on ISIC 2019 to suppress the "colored band-aid" CH within the "melanocytic nevus" class. *(Top left)*: effect of adding the computed CAV to activations at the respective layer. This CAV is computed between samples containing the "colored band-aid" artifact and those that don't, within the "melanocytic nevus" class. The mean softmax probability of the "melanocytic nevus" class increases, while true class probability decreases for layers 4 and 10, indicating that the CAV encodes a feature specific to the "melanocytic nevus" class. *Bottom left*: success of the concept suppression of P-ClArC. The poisoning of *(top left)* can be mitigated using P-ClArC, restoring confidence in the true class. For *(top left)* and *bottom left*, the layer where P-ClArC seems to perform best is marked by a green border. *(Right)*: example images and corresponding CAV-predictor- and LRP relevance difference heatmaps (similarly to Eq. (32), i.e., $R^{diff}_{P-ClArC} = R_{baseline} - R_{P-ClArC}$ for the layers 4 and 10, where the quantifications of *(top left)* and *bottom left* yielded positive results. In CAV predictor heatmaps, *red* areas indicate high relevance, i.e., highlight features indicative for the CAV direction. In the difference heatmaps, *magenta* areas were attributed less relevance, and thus used less by the model, after applying P-ClArC, while *teal* areas were used more. Refer to Supplementary Figure B.1 for a more detailed color description. The focus of the CAV-predictor seems to be relatively diffuse in layer 4, and only partly located on the targeted band-aids, supported by the only partial success of the concept mitigation, where sometimes even desired features are diminished. In contrast, both heatmap types are extremely precise at layer 10, and not only is the relevance of the CH reduced, but the melanoma itself becomes more important for the model's decision.

0.51). Although the CAV at layer 0 is not meaningful, P-ClArC can still mitigate the poisoning, showcasing again the need for our two-part quantitative evaluation, validating that not only the concept suppression is successful, but also that the CAV encodes a target-class-specific concept. For layers 4 and 10, P-ClArC restores predictions even more closely to the original values shown in Figure 20 *(top left)*, increasing confidence in the true class, while decreasing confidence for "melanocytic nevus" simultaneously. E.g., for layer 4, the former rises from 0.30 to 0.96 (unpoisoned 0.97), the latter from 0.94 to 0.52 (unpoisoned 0.51). In fact, the same result is obtained for layer 10: The poisoned probabilities deviate more extremely from an evaluation on unpoisoned data, which is also why we find an application of P-ClArC at layer 10 to be even more successful in counteracting poisoning (see *green border*).

Qualitative Results. In Figure 20 *(right)*, we aim to confirm above assertions for layers 4 and 10 by means of three sample images of the class "melanocytic nevus" that contain the targeted "colored band-aid" CH. The results here are obtained from the unperturbed data of the ISIC 2019 dataset, as opposed to the artificially poisoned setting of Figure 20 *(top left)* and *(bottom left)*. For each sample and layer, a heatmap computed for the CAV predictor is shown, highlighting areas which speak for the presence of the concept described by the CAV in *red* color, and areas speaking against it in *blue* color. Furthermore, to the *(right)* of the CAV-predictor heatmaps, the difference in relevances between the model to which P-ClArC is applied and the original model is visualized, with *magenta* areas denoting a *decreased* relevance after the application of P-ClArC. The relevance differences were computed similarly to Eq. (32), as $R^{diff}_{P-ClArC} = R_{baseline} - R_{P-ClArC}$. Conversely, *blue* areas identify features which are *increasingly* used by the model. For layer 4, the computed CAV seems to encode the "colored band-aid" concept only relatively diffusely, with some portion of the positive relevance being attributed to the nevus (i.e., the desired feature) itself, as can be seen for the middle example. Similarly, the heatmap for the CAV predictor also attributes negative relevance to the CH features. The then following CH correction results suffer from similar issues: While relevance is decreased on the "colored band-aids" themselves, often also the nevus receives less relevance, e.g., as observable with the first and second examples.

In contrast, the CAV-predictor heatmaps are far more precisely marking the confounding features in layer 10,

with not only the CH being extremely relevant, but the desired features also being a seemingly neutral (grey color in heatmap) or an even negative indicator (blue color in heatmap) for the presence of the encoded concept. The accompanying difference maps show a strong decrease in relevance for the CH areas, and a simultaneous increase in the relevance of the desired features, showing not only that P-ClArC in layer 10 successfully corrects the faulty usage of the "colored band-aid" as an important feature for the model to decide for the "melanocytic nevus" class, but also further shifts the model's focus to the actually desired features, i.e., the nevi themselves.

Since the computed CAV is not only meaningful wrt. the target class, but also exactly describes the targeted CH artifact (at least for layers 4 and 10), and since P-ClArC is able to unlearn that concept, we can thus surmise the – albeit layer-dependent – success of the P-ClArC method on the ISIC 2019 skin lesion classification dataset for mitigating the effects of training data containing CH artifacts. Due to the corrected model using desired features *preferably* to the CH features, its trustworthiness increases, reducing the risk of costly misclassifications caused by the CH.

5.4. Unlearning with Projective Class Artifact Compensation on the Adience Dataset of Unfiltered Faces

As opposed to the medical setting of ISIC 2019, we now apply P-ClArC to a gender classification task with the Adience dataset [35]. This dataset has various known problems, e.g., a relatively high class imbalance, as well as a multitude of biases within the data models tend to quickly overfit on, as in part identified in [94] via LRP.

Setup. In the gender classification setting, one of these bias-concepts is the presence of shirt collars in the class of male faces. Samples labelled as "male" with a shirt collar are a common occurence within the dataset, and samples labelled as "female" showing a shirt collar are quite rare. Thus, models trained on the Adience dataset often use use this confounding feature as a CH for the class defining the appearance of male faces, thereby short-cutting (the learning of) more complex features. This is also the case for the VGG-16 model we trained for gender classification. Using the pretrained ILSVRC2012 weights provided by Pytorch for initialization, the model was trained over 100 epochs on folds 1-4, keeping fold 0 for testing. The final accuracy achieved by this model was 94.02%.

However, the reliance of this model on CHs like the shirt collar concept may lead to unfair predictions, e.g., when a woman is predicted as "male" due to wearing clothes associated by the model with the class "male", i.e., here, a shirt collar. The impact of this is especially high in real-world applications, when stereotypes – that are apparently present in the available training data – are propagated into the inference of machine learning solutions. Here, we thus employ P-ClArC, with the aim of obtaining fairer gender predictions on the Adience dataset wrt. the "shirt collar" CH.

Quantitative Results. Figure 21 shows the results of this experiment on the fold 0 test set, comparing the original model to the model employing P-ClArC to suppress the targeted "shirt collar" CH. To compute the corresponding CAVs, two hand-selected subsets of the samples representing class "male" were used, one containing samples with shirt collars, and one without. Figure 21 *(top left)* and *(bottom left)* shows for intermediate layers 0, 4, and 10, similar to Figures 18 and 20, a quantitative evaluation of the change in mean *softmax* probabilities when using the computed CAV to poison activations at the respective layer (Figure 21 *(top left)* and when applying P-ClArC to mitigate that poisoning (Figure 21 *(bottom left)*). This change is measured for both class labels of the dataset, i.e., "female" and "male", with the latter being the target class. That is, the class for which the CH "shirt collar" is used by the model as an indicative feature. Similarly to the results obtained for the ISIC 2019 dataset, we find that for layer 0, the computed CAV does not seem to be able to concisely describe a feature specific to the target class, since activations poisoned with it lead to a decrease in the softmax probability of all classes, including class "male". To reiterate, a meaningful CAV direction, i.e., a CAV that encodes for a feature of the target class, would lead to an increase in the model's confidence on that class. However, this is not the case here with scores for class "male" dropping from 0.51 to 0.32 probably due to the raw input data that has not yet been affected by any learned invariant internal representation of the model, being too complex for the CAV to successfully describe. Note that the softmax scores for class "female" simultaneously increase in this setting (from 0.49 to 0.68). That is, however, a byproduct of the confidence decrease for class "male" due to the binary classification task. The poisoning counteraction of P-ClArC for layer 0 is comparatively successful (Figure 21 *(bottom left)*), with the original probabilities from Figure 21 *(top left)* only being within a margin of error of only 0.04. But since the computed CAV is not fully meaningful, the direction removed by P-ClArC cannot correspond to the target concept for layer 0. In contrast, for layer 4 and even more so for layer 10, as indicated by the *green* border, the poisoning in Figure 21 *(top left)* yields the expected results, increasing the predicted probability of class "male" on average for, e.g., layer 10, from 0.51 to 0.94, while decreasing it for class "female" from 0.49 to 0.06. In

Figure 21 *(bottom left)*, however, the removal of CAV-poisoning seems to overreach for layers 4 and 10: The original predicted probabilities of 0.49 for class "female" and 0.51 for class "male" are not exactly restored, instead, e.g., for layer 10 the confidence of class "female" rises from 0.06 to 0.60 (instead of 0.49 for a perfect recovery), while it drops from 0.94 to 0.40 (instead of 0.51) for class "male", with 11% discrepancy compared to the original values. Keeping in mind that Figure 21 *(top left)* shows that the CAV is meaningful wrt. the target class, we thus infer that either the CAVs for layers 4 and 10 encode the targeted concept – and removing it affects the prediction so much because the model strongly relies on that feature, or the CAV encodes *not only* the shirt collar, but additionally other (possibly valid) features for class "male" that appear alongside shirt collars with a relatively large correlation. In any case, layer 10 is marked with a *green* border, since the concept suppression effect is strongest there.

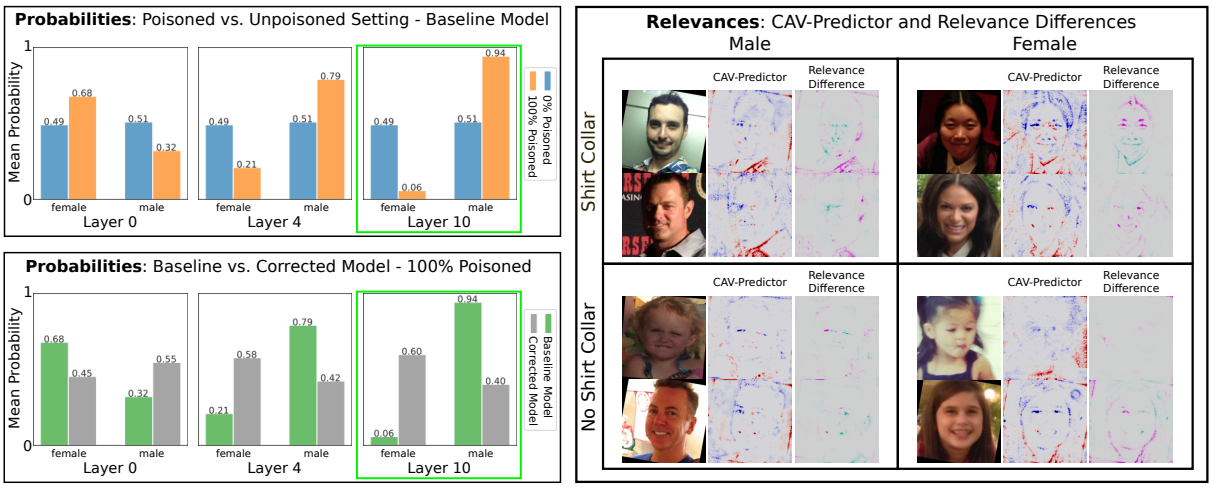
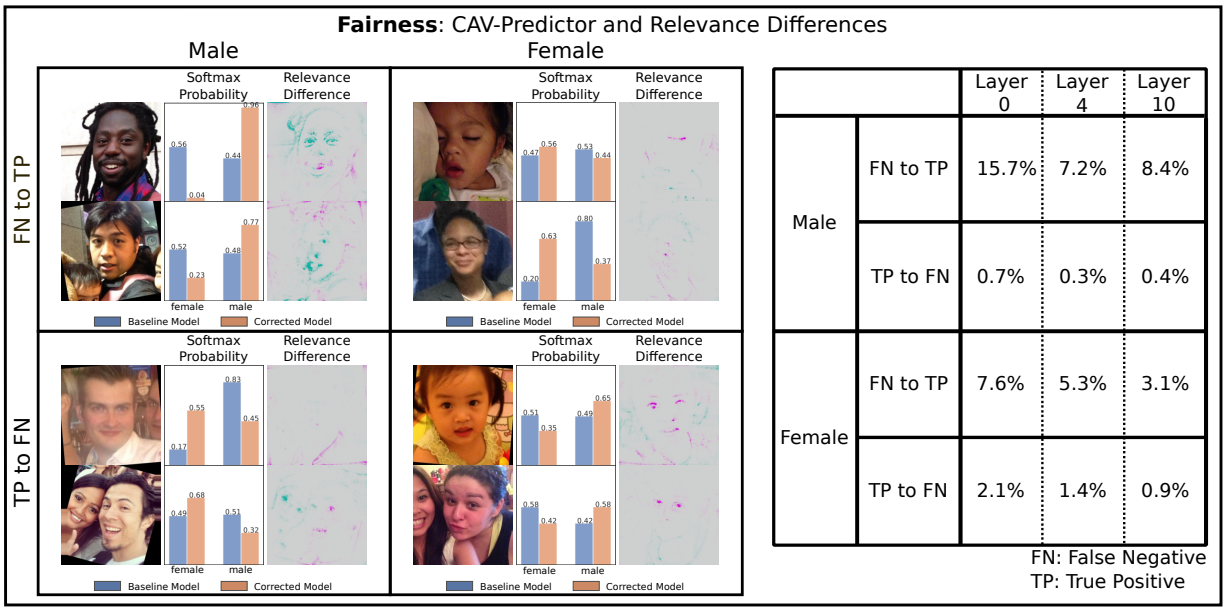


Figure 21: Application of P-ClArC on the Adience dataset, with the aim to obtain less stereotypical and fairer predictions. The target CH is the "shirt collar" concept used by the model to predict in favor of the class "male". The associated CAV is obtained using two hand-selected subsets of the samples representing class "male" were used, one containing samples with shirt collars, and one without. *(Top left)*: By adding the computed CAV to activations at the respective intermediate layer, the prediction can be affected in such a way that confidence on class "male" increases, showing that the CAV describes a concept specific to "male". The layer where this works best is marked by a green border. *(Bottom left)*: Using P-ClArC, poisoning via the computed CH is easily mitigated. As a result, the softmax probabilities on the class "female" increase, while they decrease for "male". *(Right)*: CAV-predictor and LRP relevance difference heatmaps (similarly to Eq. (32), i.e., $R^{diff}_{P-ClArC} = R_{baseline} - R_{P-ClArC}$) at layer 10 (the best performing layer according to *(top left)* and *(bottom left)*) for examples of both genders, with and without the target CH "shirt collar" each. For the former, *red* areas indicate high relevance, for the latter, *magenta* areas were attributed less relevance after applying P-ClArC. *blue* and *teal* areas indicate the respective opposite. Refer to Supplementary Figure B.1 for a more detailed color description. The artifact is predicted and suppressed successfully if present in class "male", however, in the class "female", this is not always the case.

Qualitative Results. For this layer 10, Figure 21 *(right)* shows samples for both classes "male" and "female", both with and without the target CH "shirt collar", respectively. Each sample is accompanied by two types of LRP relevance maps, the first on the *(left)* showing which features are important for the CAV-predictor in *red*, i.e., which features indicate the presence of the target concept as it is represented via the computed CAV, while features speaking against it are highlighted in *blue* color. The second relevance map (computation similarly to Eq. (32), i.e., $R^{diff}_{P-ClArC} = R_{baseline} - R_{P-ClArC}$ evaluates features of the respective sample that are used less by the model for its predictions after the application of P-ClArC in *magenta* color, and features that are used more in *teal* color. On images of the target class "male" that contain the target CH *(top left)*, at first glance positive relevances in the CAV-predictor heatmap seem to focus on the actual shirt collar, indicating that the computed CAV does encode for the target concept. In the relevance difference maps, however, while the relevance of the shirt collar decreases with an application of P-ClArC and that of the facial features (i.e., the features desired to be used by the model, naively summarized) increases some other features, e.g., visible and uncovered ears, seem to also be suppressed. In the CAV-predictor heatmap, these are assigned a small positive relevance. As found by [94], specifically the visible ears also tend to be learned by models as an indicator for class "male" and possibly even constitute a CH. Apparently, these features often appear alongside the positive examples for the "shirt collar" concept, thereby leading to the computed CAV not only encoding



		Layer 0	Layer 4	Layer 10
Male	FN to TP	15.7%	7.2%	8.4%
	TP to FN	0.7%	0.3%	0.4%
Female	FN to TP	7.6%	5.3%	3.1%
	TP to FN	2.1%	1.4%	0.9%

FN: False Negative
TP: True Positive

Figure 22: Analyses of transitions between *true positive* and *false negative* predictions when applying P-ClArC. Examples for layer 10 of which the predicted class is flipped are shown to the *(left)*, together with softmax probabilities of each sample before and after using P-ClArC and the corresponding change in relevances. Here, *magenta* areas were attributed less relevance after applying P-ClArC, and *teal* areas more. Refer to Supplementary Figure B.1 for a more detailed color description. The table to the *(right)* shows the percentage of original *true positives* that change to *false negatives*, and vice versa. Generally, a higher percentage of *false negatives* is corrected than *true positives* are confused. Due to the original model being 94% accurate, however, a larger absolute number of samples are changed from *true positives* to *false negatives*, leading to an overall decrease in accuracy.

for "shirt collar" features, but additionally for other – possibly CH features of the class "male", further confirming our suspicions regarding the large shift in mean softmax probabilities when P-ClArC is applied at layers 4 and 10 in Figure 21 *(bottom left)*. As the Adience dataset is an extremely complex dataset with highly biased data, a noisy CAV encoding is to be expected, especially, since the precision of the CAV is highly dependent on the samples chosen for its computation.

In contrast, when the target CH is not present (Figure 21 *(right) Male – No Shirt Collar*), correctly no collar is identified. Although, again, uncovered ears seem to receive partial positive relevance. For the "female" class, however, even though the shirt collar is identified by the CAV-predictor relevance maps ((Figure 21 *(right) Female – Shirt Collar*); albeit by far not as precisely as for class "male" – "collar"), it does not seem to diminish reliably after applying P-ClArC. Instead, e.g., in the top example, its relevance in the prediction process even increases, and the concept removal seems to focus mostly on the eyes and hairline. Contrary to class "male", an application of P-ClArC is as successful for samples of class "female". This brings up a possible issue with using CAVs to represent the target CH artifacts that we have previously only briefly touched upon: within the Adience dataset, the "shirt collar" CH only has a significant presence within class "male" – leading to positive and negative "shirt collar" examples for the CAV computation only reliably obtainable from samples of class "male". However, because the CAV is only computed using samples from one class, and because its ability to distinguish a concept relies entirely on the data used for fitting the corresponding linear classifier, it does not necessarily encode the target CH as precisely when faced with samples from another class since the domain changes for the CAV model. For the samples of class "female" without shirt collar features, no shirt collar is found and consecutively not removed (similar to the corresponding "male" samples). In the second example in Figure 21 *(right) Female - No Shirt Collar* the shape of the long hair seems to be identified as a shirt collar, showcasing another issue for this specific CH among samples belonging to class "female".

To summarize, while the concept suppression of P-ClArC seems to have a similar success on the "male" class as we previously found for CH in other datasets, albeit slightly more noisy due to the complex nature of the Adience dataset, applying it to the "female" class sheds light on various issues, e.g., a relatively strong domain dependence of

the computed CAVs.

Towards Fairer Predictions. Even though the previous results are relatively mixed, we evaluate the ability of P-ClArC to achieve *fairer* predictions in Figure 22. Here, the table to the *(right)* shows for layers 0, 4, and 10 and both classes the percentage of previously mispredicted (false negatives, i.e., FN) and correctly predicted samples (true positives,i.e., TP) of which the predicted class changed after an application of P-ClArC, turning them into true positives and false negatives, respectively. Relatively, more false negatives turn into true positives when P-ClArC is applied. Where we found the computed CAV for layer 0 to not be meaningful wrt. the target class, the FN to TP rate is comparatively high with 15.7% for class "male" and 7.6% for class "female". At the same time, however, the TP to FN rate is also significant, with 0.7% for class "male" and 2.1% for class "female". In layer 4, they decrease to 7.2%, 0.3%, 5.3% , and 1.4%, respectively. In layer 10, an interesting phenomenon occurs, with the rates growing to 8.4% (FN to TP) and 0.4% (TP to FN) for class "male", but still diminishing for class "female", to 3.1% and 0.9%. A large amount of samples changing from TP to FN and vice versa is not necessarily a sufficient measurement on its own, because many alterations to the model's inference process would have that effect, especially since with an accuracy of 94.02%, there are far more TP than FN absolutely. E.g., this seems to happen for P-ClArC with a badly encoded CH, as is the case for layer 0, according to our findings in Figure 21. However, both (TP to FN) and (FN to TP) rates seem to steadily diminish with higher layers, presumably due to alterations later in the network not being propagated as far and thus having a lessened effect, except – as noted above – for layer 10 of (only) the class "male", where a sudden increase occurs. This observation corresponds to our two previous assertions, that the layer 10 CAV and P-ClArC process for class "male" is quite precise wrt. the target concept "shirt collar" – although some other correlating distinct "male" features are also affected. For class "female", the same artifact does not seem to be as well defined.

In any case, a closer look at the affected samples is needed to come to a conclusion. For this purpose, Figure 22 *(left)* shows examples of which the prediction switched after applying P-ClArC in layer 4 are shown, along with the softmax probabilities of the respective samples before and after the attempted correction wrt. the CH concepts, together with the corresponding attribution difference maps, for classes "male" and "female" and both types of prediction change. For the class "male", samples seem to be predicted from TP to FN *(Male – TP to FN)* due to the target concept, i.e., "shirt collar" or correlating male features like uncovered ears, being suppressed successfully. The accompanying change in softmax probabilities is quite significant, especially for the first example. Interestingly, in the second example, the female face visible in the image gains in attributed relevance due to the removal of features corresponding to class "male". Furthermore, the change from FN to TP *(Male – FN to TP)* appears to happen due to more significance being attributed to facial features, and less to surrounding features. Interestingly, in the top example, part of a "shirt collar" is removed, but confidence for "male" is *increased*, perhaps due to the colorful expression of the visible clothing item. Again, we note significant changes in the predicted class probabilities. In contrast, for class "female" , probabilities often seem to only change slightly and due to the model having difficulties classifying the sample in the first place, as is the case, e.g., for small children (*Female – FN to TP and TP to FN*, first sample each). However, we also observe changes from FN to TP due to a shirt collar feature being withheld from the model (*Female – FN to TP,* second image), although the shirt collar removed here is a misinterpreted pearl necklace (Since the necklace is quite thin, this result is best viewed in digital format), and the corresponding alterations in relevance are by far not as distinct as for the examples labelled as "male". Even so, the accompanying discrepancies in softmax probabilities are notably higher for examples such as this, where the classification changes due to valid (wrt. the targeted CH) reasons.

Discussion. To summarize, on the Adience dataset – which is admittedly quite difficult to solve, due to its various inherent biases and imbalances – , we found that the influence of even highly complex CH, e.g., the "shirt collar" of class "male", can be successfully mitigated via P-ClArC, although not quite as precisely and significantly as achieved for, e.g., the ISIC 2019 dataset. Especially the issue of P-ClArC not being transferable between classes without losing in precision of the CH correction becomes clear if a concept is present within multiple classes but the CAV representation is only learned from samples of a single class. This, however, seems to be a problem of the representation only being computed from samples of one class – due to a sufficient number of examples expressing the CH sufficiently well only being available from the target class – , not the P-ClArC method itself. Finding more accurate and generalizing representations is subject to future work. In terms of fairness, we conclude that for the target class, the predictions after applying P-ClArC become more focused on the desired features, leading to classifications for the right reasons.

For the other class, this is not always the case due to the representation issue stated above, however, *if* the concept is detected and suppressed correctly, the resulting difference in predicted probabilities is far more significant.

6. Lessons Learned

This section first aims at putting the proposed ClArC method and its variants, A-ClArC and P-ClArC, into context, directly comparing them to similar approaches. We further offer a summarizing discussion of our findings, giving a clear overview over the potential of SpRAy and ClArC, any drawbacks or caveats, and indicate where additional research may be required. Lastly, we add a best practice manual, indicating how to successfully apply the workflow of methods presented in this paper when dealing with a new dataset.

6.1. Comparison to other Unhansing Methods

There already exist multiple methods that successfully correct a model's reasoning by alleviating CH effects or similar data-intrinsic biases. Table 1 offers an overview over these approaches, and compares them to the methods presented in this work, A-ClArC and P-ClArC, in terms of requirements, runtime, and how biases are mitigated.

Table 1: Comparison between existing methods for mitigating CH effects and the two methods presented in this work, A-ClArC and P-ClArC.

Method	*Requirements*	*Runtime*	*Type of Mitigation*
XIL	Regular synchronous interaction with a human expert during training	Increased training time	Updated Parameters
LNTL	Additional bias labels for the complete dataset	Normal training time	Updated Parameters
RRR	Ground truth explanations for the complete dataset	Normal training time	Updated Parameters
CDEP	Ground truth explanations for the complete dataset	Normal training time	Updated Parameters
A-ClArC	Sufficient artifact labels to model the artifact (e.g., obtained via SpRAy); Inductive artifact model	Normal training time	Updated Parameters
P-ClArC	Sufficient artifact labels to model the artifact (e.g., obtained via SpRAy); Suppressive artifact model	Inference time only	Concept Suppression

Finding CH-Artifacts. CH-like artifactual contamination is not only quite difficult to detect, as we demonstrated in Section 3.1, but also not definable without access to some ground truth information about desired and undesired features, which in practice can *only* be provided by a human expert. Using only the standard data and label information, CH artifacts and prominent but legitimate class features cannot be reliably distinguished. For this reason, to the best of our knowledge, all existing approaches, that aim to reduce the impact of CH-like data biases on the decisions a model makes, leverage additional ground truth information from human experts. However, this ground truth information is usually assigned in a sample-wise fashion, i.e., as bias labels [66], ground truth explanation masks [67, 64], or even reactive corrections during runtime [65, 50]. Obtaining these annotations can be infeasible for large datasets, or even slow down training time immensely, as is the case for XIL [65, 50]. By employing SpRAy, which bridges the gap between local and global XAI, we circumvent the issues of instance-wise ground truth annotations. Here, CH candidates are provided in a clustered and automated fashion, and only the decision whether a cluster of samples describes an actual CH is made by a human expert. This process speeds up annotation time dramatically in comparison to other methods, where the human expert would need to annotate every single sample, and naturally provides the relevant artifact labels from the clustering.

CH-Artifact Mitigation. When mitigating CH-effects, usually the goal is to improve a model's reasoning, so that it achieves a high performance without using certain undesired features for its decision-making. This is achieved by

updating the model's parameters via some training or finetuning scheme, as is done by [66, 67, 64, 65, 50], and for A-ClArC. While that process leads to a model that is more robust against biases and CHs, it is also expensive in terms of computation time. For this reason, we alternatively propose the extremely resource-efficient P-ClArC method, which does not adapt a model's parameters, and instead only suppresses a target concept. P-ClArC is thus extremely fast, since, after the targeted artifact is modeled once, no additional training is required. Instead, predictions that are less affected by a CH can directly be inferred, allowing for a more truthful generalization performance estimation.

Key Insights

Compared to existing CH and bias mitigating approaches,

- Employing **SpRAy** streamlines CH/bias annotation time and thus dramatically reduces the required amount of human involvement in the unlearning process.
- **A-ClArC** has a similar runtime to other training-based unlearning approaches (assuming all methods' requirements are met) and unlearns CHs by updating model parameters.
- **P-ClArC** does not require any finetuning and is therefore much faster and more efficient, but in turn only suppresses CHs concepts.

6.2. Summary of Experimental Results

In our experiments, we first demonstrated in a toy setting the difference between two types of dataset confounders, BDs and CHs, and illuminated the additional challenges when dealing with the latter, which appear alongside legitimate class features and are thus not only weaker, but far more difficult to detect. By employing local XAI-techniques, however, even the more elusive CH artifacts can be found, although this process is extremely tedious to perform manually, sample for sample, in practice.

For this reason, we employed and extended the SpRAy framework, which aims at offering global insights about a model's behavior based on local explanations. While investigating the capabilities of SpRAy for the purpose of CH detection, we found that SpRAy is successfully able to offer candidates that are extremely likely to be CHs, although a final human decision is still required. The artifact complexity strongly affects the layer where its separability score, informing about its likelihood of having a CH character, is highest. Simple artifacts, that can be expressed as affine transformations in input space, tend to best separate in lower layers. Highly complex artifact features, on the other hand, have a larger separability score in higher layers. However, we also observed that a high separability score does not necessarily mean that a CH is present, because whether a feature is a CH or legitimate is ultimately a subjective decision, even if the feature is prominent enough to separate well. Conversely, not every CH necessarily has a high separability score, as some CHs strategies may be non-trivial and thus not linearly separable in the embedded space. Further research is needed in this context, into whether objective criteria for CHs can be derived, and if and how the candidates offered by SpRAy can be determined to be actual CHs or not in a completely automated fashion.

SpRAy not only allows for uncomplicated identification of potential CHs by leveraging local XAI, but also assigns cluster labels that can easily be used to label whether a sample contains a specific CH. Given those labels, we estimated an inductive or suppressive artifact model, and mitigated specific CH effects using A-ClArC or P-ClArC, respectively. In a toy setting, A-ClArC was able to successfully adapt models to ignore artificially introduced CHs artifacts. On larger datasets, with naturally occurring CHs, we found that A-ClArC not only retained performance on unpoisoned data, but also increased performance on data systematically poisoned by the target CH. A-ClArC mitigated significant artifacts successfully, although, if the artifact was too small or not specific enough to the target class, A-ClArC did not always show a pronounced effect. Generally, the A-ClArC'ed model focused less on CH features and more on legitimate class features, compared to a model that did not employ A-ClArC, however, since this is a relative observation, complete CH *removal* cannot be guaranteed. We again uncovered a strong connection between artifact complexity and the layer where it can be unlearned best, similar to our observations for SpRAy in feature space. Further experiments are required in order to determine in a more complete fashion which types of artifact may be difficult to unlearn using A-ClArC, and how a complete invariance to CH concepts could be guaranteed.

A-ClArC adapts a model's parameters to grow more invariant against a target CH concept. However, this process involves tedious, resource-costly and time-consuming finetuning of the model. In an effort to drastically increase efficiency, we proposed P-ClArC, which is extremely fast and requires no training, but in turn only *suppresses* artifacts and does not adapt any model parameters. Neither variant of ClArC increases generalization performance of the model in terms of accuracy on the test set, but they instead (arguably) increase generalization in terms of being less affected by decisions that rely on CH concepts. A model that employs P-ClArC is more trustworthy, as wrong decisions due to CH confounders grow less likely. Here, we observed quantitatively and qualitatively that the suppressive artifact model (here CAVs) was generally able to correctly model and suppress a target CH concept on a toy and realistic data, while preserving performance on unpoisoned data and increasing performance on poisoned data. Again, a strong layer-dependence of the CH mitigation in connection with the artifact complexity was found. However, we also discovered that highly complex CHs (e.g., "collar" in the Adience dataset) were difficult to model successfully. If the artifact model is not sufficiently precise, P-ClArC may not mitigate artifacts as reliably. Furthermore, since CH artifacts per definition only have a strong effect on the model in conjunction with legitimate class features, they need to be modeled from samples within the affected class, because the artifact model may end up as a detector of legitimate class features otherwise. However, if that artifact model is applied to other classes during P-ClArC, performance may vary. The artifact model may therefore be highly domain dependent on the target class, and is not necessarily transferable to other classes, even if the same concept is present in both. Further research is required to analyze above issues in-depth and potentially solve them.

Key Insights

During our experiments, we observed the following:

- **CHs** are weaker and far more difficult to detect than BDs. However, local XAI techniques can aid in identifying even the comparatively more elusive CHs artifacts.
- **SpRAy** offers very likely CH candidates, but no definitive distinction. Therefore, a final human decision is still necessary for contextualizing suggested CH candidates wrt. the model's intended task, and further research is required into if and how CHs can be detected by SpRAy in an entirely automated fashion. Depending on the artifact complexity, SpRAy may be able to separate it better in higher layers.
- Both **A-ClArC** and **P-ClArC** successfully mitigate target CH concepts, while preserving performance on unpoisoned data. The layer where the concept mitigation performs best depends on the artifact complexity. For both methods, we only observed the mitigation relatively, and thus complete CH removal cannot be guaranteed, requiring further research. For P-ClArC, the artifact model is highly domain-dependent on the target class, and cannot necessarily be transferred to other classes.

6.3. How to Deal With a New Dataset?

Given the many observations and results, we may formulate a set of *best practices* when dealing with a new unknown dataset. A general step-by-step approach is shown in Table 2.

When nothing is known about a dataset, a first manual look at some samples can already give a rough idea of potential artifacts. *Steps 1 and 2* may already uncover potential CH behavior in the model when a lot of samples contain artifactual features. Depending on what we are looking for, we can group CH artifacts in 2 possible types: *Static* artifacts are simple and always appear in the same set of features, which may be a watermark in image data, or some background with a static frequency when dealing with audio data in frequency domain. *Conceptual* artifacts on the other hands are more complex, as they may be arbitrarily abstract and thus not be easily detectable in input space. They are generally only a problem if a model has successfully learned and encoded them within its intermediate representations.

In *Step 3* we use SpRAy to find *static* artifacts by using attributions wrt. the inputs of the model, or to find *conceptual* artifacts by using attributions wrt. intermediate features of the model, i.e., intermediate activations of a neural network. In the general case, it may be best to start with *static* artifacts and later investigate deeper by

Table 2: Detecting and dealing with CH behaviour on an unknown dataset.

Step	Action
1	Manually investigate a few selected Samples: Is there anything that stands out?
2	Use a local XAI technique (i.e. LRP) and visualize the attribution for the same samples. Does the attribution seem reasonable?
3	Apply SpRAy (Algorithm 1) on the whole data set in input space to find static artifacts. Compute the linear separability score τ (Eq. (7)) of the obtained clusterings.
4	Look at the 2d embedding (e.g. t-SNE) of the attributions each of the classes with the highest separability scores, color the points by their predicted clusters. Visualize the attributions grouped by their clusters: Is there a cluster in which unrelated features are predominantly attributed?
5	If there is such a cluster, label these samples and create an artifact model (using an additive model (Eq. (20)), a blending model for image data (Eq. (23)) or CAVs (Eq. (24)).
6	Given the artifact model, use A-ClArC or P-ClArC (Algorithm 2) to test for and to alleviate the artifact one at a time. The performance can be measured quantitatively by the classification accuracy on a poisoned test set using *inductive* artifact models, or qualitatively by investigating the attributions of affected samples before and after the alleviation.
7	Repeat steps 3-6 using attributions in feature space to find conceptual artifacts, which may not be detectable using only attributions in input space.

continuing with intermediate features to find *conceptual* artifacts. The output of SpRAy is a Spectral Embedding of each data point, grouped by classes, where each class has one or more proposed clusterings, each with its own separability score τ.

While we can potentially investigate all classes given an appropriate visualization of the embedding, clusters and samples in *Step 4*, classes with clusterings of high separability should be investigated first, as there is an increased possibility of CH behavior. Clusters that are are outliers in the 2d-embedding are often a very good indicator for CH.

The samples of suspicious clusters are labeled in *Step 5*. Samples do not need to be labeled too cautiously, since their actual impact can be measured by the behaviour of the model after their respective artifact model was used for A-ClArC or P-ClArC. These labels can then be used to construct either an *inductive* artifact model for fine-tuning with A-ClArC, or a *suppressive* artifact model to modify the model directly using P-ClArC.

Given our observations, it may be most straight-forward to use CAVs as a *supressive* artifact model in feature space to use with P-ClArC in *Step 6* first. Due to the cheap computational cost, we recommend to try CAVs at multiple layers, to find the representation which encodes the potential artifact best. Both the influence of the artifact, and the success of the alleviation can be obtained by comparing the model perfomance on a test set where each sample has been poisoned, and on another one where no samples were poisoned. In order to create poisoned samples, an *inductive* artifact model is necessary, which for example may be constructed with the same CAV used for P-ClArC. A qualitative assessment of the alleviation may be obtained by investigating the attributions of the affected samples before and after the alleviation process. For a successful alleviation, attribution on the features relevant for the CH artifact should decrease. A drop in model performance alongside a successful removal of the target artifact's influence with P-ClArC implies that the model has overfit significantly on the artifact, during initial training. In such a case, the model becomes unable to recognize the true class due to P-ClArC's suppression of its key associated features. Here, we recommend an application of A-ClArC instead, which is computationally more expensive, but allows the model to be corrected by shifting its predictive strategies to be based on an alternative feature set.

Finally, the process may be repeated from *Step 3* to *Step 6* on an intermediate representation to find conceptual artifacts.

7. Conclusion

Deep Learning models have gained high practical usability by pre-training on large corpora and then reusing the learned representation for transferring to novel related data. A prerequisite for this practice is the availability of large

sets of rather standardized and, most importantly, representative data. If artifacts or biases are present in data, then the representations formed are prone to inherit these flaws. This is clearly to be avoided, however, it requires either clean data or detection and subsequent removal of the influence of artifacts, biases etc. of databases that would cause dysfunctional representation learning.

In this paper we have used techniques from eXplainable Artificial Intelligence (e.g., LRP [17] and SpRAy [12] with several meaningful extensions), and introduced the Class Artifact Compensation framework to scalably and automatically detect, validate and alleviate Clever Hans behavior in multiple recent and large data corpora. Similar to how the absence of a CH behaviour could not be guaranteed, its complete removal could neither be guaranteed. However, model performance was retained and we observed the model's focus on the Clever Hans strategy to significantly reduce given the utilized XAI methods. While we mainly used LRP, the proposed ClArC framework is independent of the particular XAI method. We demonstrated how they can aid to detect and avoid their impact. In this context, we can can identify requirements for XAI methods that are necessary to improve the discovery and quantification of disruptive artifactual concepts for their elimination: (1) comparability of local explanations: it must be possible to compare explanations of different samples (2) smooth and robust attributions: same (local) features must produce the same explanation, and must be robust to noise (3) explanation of abstract concepts: it must be possible to attribute abstract concepts, which cannot be detected by attributing input space alone. Local attribution methods may be used in feature space of neural networks to identify concepts which cannot be represented in input space. (4) a combination of local and global must be used to understand prediction strategies of the model better. We have shown that current attribution methods, optionally applied to intermediate representations, in combination with SpRAy fulfill these requirements.

ClArC encompasses a first simple intuition based of how artifacts may harm generalization. As this intuitive model is based on logistic regression, it is rather crude, but it already shows the main effects caused by artifacts: deterioration of generalization ability. For neural networks it may, however, still serve as a reasonable guideline and indeed our large-scale experiments on various datasets show analogous effects, that can exhibit a dramatic drop of generalization for some classes.

Based on the ClArC model of artifactual features, we have introduced two concrete algorithms to implement the desensitization and unlearning of undesired features in a deep neural network: First, we proposed A-ClArC, an approach building on strategic augmentation of the data and subsequent fine-tuning of the model in order to remove the influence of artifactual confounders from inference. Second, we aim at P-ClArC suppressing the representation of an artifact as a feature to prevent its use in inference. While the latter approach is extremely efficient as it does not involve any training beyond the modeling of the artifact itself, the former can drive the model to adapt to a different, benign set of features. Both approaches can be applied on artifact representations obtained in input spaces, as well as latent space.

Let us discuss the main experimental findings. Based on an extended SpRAy technique we could in toy settings verify artificially created Clever Hans artifacts, and automatically detect some rather unexpected Clever Hans strategies of a popular pre-trained VGG-16 deep learning model on ILSVCR2012. These are caused by a zoo of artifacts and biases isolated by our framework in the corpus: encompassing copyright tags, unusual image formatting, specific co-occurrences of unrelated objects, cropping artifacts, just to name a few. Detecting this zoo gives not only insight but also the possibility for relieving models and datasets from their Clever Hans moments, i.e., based on our theoretical findings, we are now able, using ClArC, to implicitly un-Hans large reference datasets such as the ImageNet corpus and thus provide a more consistent basis for pre-trained models. We demonstrated this in *unlearning experiments* for several artifactual features on ImageNet, and in practical application scenarios, i.e., the ISIC 2019 dataset skin lesion prediction dataset and the Adience benchmark dataset of unfiltered faces, yielding more representative predictors for the tasks. In all scenarios, we observe that a precise modeling of the artifact, i.e., the availability and use of representative data distinguishing artifactual features from desired ones, will have a beneficial effect on the success of both ClArC variants.

Let us reiterate that without removing, or at least considering such data artifacts, learning models are prone to adopt Clever Hans strategies [12], thus, giving the correct prediction for an artifactual/wrong reason. Once these artifacts are absent or appear in unusal combination with other features in the wild, such Clever Hans models will experience significant loss in generalization (see, e.g., Figures 14, 20 and 21). This makes them especially vulnerable to adversarial attacks that can harvest all such artifactual issues in a data corpus [95].

Future work will therefore focus on the important intersection between security and functional cleaning of data

corpora, e.g., to lower the attack risk when building on top of pre-trained models.

Acknowledgements

We acknowledge Marina Höhne for valuable discussion. This work was supported in part by the German Ministry for Education and Research (BMBF) under grants 01IS14013A-E, 01GQ1115, 01GQ0850, 01IS18056A, 01IS18025A and 01IS18037A. This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 965221, and is also supported by the Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (No. 2017-0-001779), as well as by the Research Training Group "Differential Equation- and Data-driven Models in Life Sciences and Fluid Dynamics (DAEDALUS)" (GRK 2433) and Grant Math+, EXC 2046/1, Project ID 390685689 both funded by the German Research Foundation (DFG).

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagennet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems (NIPS), 2012, pp. 1097–1105.

[2] Y. LeCun, Y. Bengio, G. E. Hinton, Deep learning, Nature 521 (2015) 436–444.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (2016) 484–489.

[5] V. Firoiu, W. F. Whitney, J. B. Tenenbaum, Beating the world's best at super smash bros. with deep reinforcement learning, CoRR abs/1702.06230 (2017).

[6] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, Ç. Gülçehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. P. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, D. Silver, Grandmaster level in starcraft II using multi-agent reinforcement learning, Nature 575 (2019) 350–354.

[7] K. T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, A. Tkatchenko, Quantum-chemical insights from deep tensor neural networks, Nature Communications 8 (2017) 13890.

[8] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Zídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis, Improved protein structure prediction using potentials from deep learning, Nature 577 (2020) 706–710.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, F. Li, Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (2015) 211–252.

[10] P. Stock, M. Cissé, Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases, in: Proc. of European Conference on Computer Vision (ECCV), 2018, pp. 504–519.

[11] O. Pfungst, Clever Hans: (the horse of Mr. Von Osten.) a contribution to experimental animal and human psychology, Holt, Rinehart and Winston, 1911.

[12] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking clever hans predictors and assessing what machines really learn, Nature Communications 10 (2019) 1096.

[13] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digital Signal Processing 73 (2018) 1–15.

[14] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, K.-R. Müller (Eds.), Explainable AI: Interpreting, explaining and visualizing deep learning, Springer LNCS 11700 (2019).

[15] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, K.-R. Müller, Explaining deep neural networks and beyond: A review of methods and applications, Proceedings of the IEEE 109 (2021) 247–278.

[16] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, K.-R. Müller, How to explain individual classification decisions, The Journal of Machine Learning Research 11 (2010) 1803–1831.

[17] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS ONE 10 (2015) e0130140.

[18] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proc. of IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626.

[20] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proc. of International Conference on Machine Learning (ICML), 2017, pp. 3319–3328.

[21] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: Proc. of International Conference on Machine Learning (ICML), 2017, pp. 3145–3153.

[22] M. T. Ribeiro, S. Singh, C. Guestrin, 'why should I trust you?': Explaining the predictions of any classifier, in: Proc. of ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2016, pp. 1135–1144.
[23] L. M. Zintgraf, T. S. Cohen, T. Adel, M. Welling, Visualizing deep neural network decisions: Prediction difference analysis, in: Proc. of International Conference on Learning Representations (ICLR), 2017.
[24] R. C. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in: Proc. of IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3449–3457.
[25] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (2002) 389–422.
[26] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[27] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. B. Viégas, R. Sayres, Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV), in: Proc. of International Conference on Machine Learning (ICML), 2018, pp. 2673–2682.
[28] R. Rajalingham, E. B. Issa, P. Bashivan, K. Kar, K. Schmidt, J. J. DiCarlo, Large-scale, high-resolution comparison of the core visual object recognition behavior of humans, monkeys, and state-of-the-art deep artificial neural networks, Journal of Neuroscience 38 (2018) 7255–7269.
[29] S. M. Lundberg, G. G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S. Lee, Explainable AI for trees: From local explanations to global understanding, CoRR abs/1905.04610 (2019).
[30] P. Tschandl, C. Rosendahl, H. Kittler, The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, Scientific Data 5 (2018) 180161.
[31] N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. K. Mishra, H. Kittler, A. Halpern, Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC), in: Proc. of the IEEE International Symposium on Biomedical Imaging (ISBI), IEEE, 2018, pp. 168–172.
[32] M. Combalia, N. C. F. Codella, V. Rotemberg, B. Helba, V. Vilaplana, O. Reiter, A. C. Halpern, S. Puig, J. Malvehy, BCN20000: dermoscopic lesions in the wild, CoRR abs/1908.02288 (2019).
[33] Y. LeCun, The MNIST database of handwritten digits, `http://yann.lecun.com/exdb/mnist/`, 1998.
[34] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. of the IEEE 86 (1998) 2278–2324.
[35] E. Eidinger, R. Enbar, T. Hassner, Age and gender estimation of unfiltered faces, Proc. of the IEEE Transactions of Information Forensics Security 9 (2014) 2170–2179.
[36] Parliament and Council of the European Union, General data protection regulation (2016).
[37] B. Goodman, S. R. Flaxman, European union regulations on algorithmic decision-making and a "right to explanation", AI Magazine 38 (2017) 50–57.
[38] C. Soneson, S. Gerster, M. Delorenzi, Batch effect confounding leads to strong bias in performance estimates obtained by cross-validation, PLoS ONE 9 (2014) e100335.
[39] O. Z. Kraus, L. J. Ba, B. J. Frey, Classifying and segmenting microscopy images with deep multiple instance learning, Bioinformatics 32 (2016) 52–59.
[40] Y. Yang, V. Tresp, M. Wunderle, P. A. Fasching, Explaining therapy predictions with layer-wise relevance propagation in neural networks, in: Proc. of IEEE International Conference on Healthcare Informatics (ICHI), 2018, pp. 152–162.
[41] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, H. Müller, Causability and explainability of artificial intelligence in medicine, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9 (2019) e1312.
[42] M. Hägele, P. Seegerer, S. Lapuschkin, M. Bockmayr, W. Samek, F. Klauschen, K.-R. Müller, A. Binder, Resolving challenges in deep learning-based analyses of histopathological images using explanation methods, Scientific Reports 10 (2020) 6423.
[43] A. Binder, M. Bockmayr, M. Hägele, S. Wienert, D. Heim, K. Hellweg, M. Ishii, A. Stenzinger, A. Hocke, C. Denkert, et al., Morphological and molecular breast cancer profiling through explainable machine learning, Nature Machine Intelligence 3 (2021) 355–366.
[44] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, P. McDaniel, cleverhans v1.0.0: an adversarial machine learning library, CoRR abs/1610.00768 (2016).
[45] T. Gu, B. Dolan-Gavitt, S. Garg, BadNets: Identifying vulnerabilities in the machine learning model supply chain, CoRR abs/1708.06733 (2017).
[46] B. Tran, J. Li, A. Madry, Spectral signatures in backdoor attacks, in: Advances in Neural Information Processing Systems 31, 2018, pp. 8011–8021.
[47] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: Proc. of International Conference on Learning Representations (ICLR), 2014.
[48] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, B. Y. Zhao, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, in: Proc. of IEEE Symposium on Security and Privacy (SP), 2019, pp. 707–723.
[49] R. Geirhos, J. Jacobsen, C. Michaelis, R. S. Zemel, W. Brendel, M. Bethge, F. A. Wichmann, Shortcut learning in deep neural networks, CoRR abs/2004.07780 (2020).
[50] P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H.-G. Luigs, A.-K. Mahlein, K. Kersting, Making deep neural networks right for the right scientific reasons by interacting with their explanations, Nature Machine Intelligence 2 (2020) 476–486.
[51] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, W. Samek, Analyzing classifiers: Fisher vectors and deep neural networks, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2912–2920.
[52] J. Lehman, J. Clune, D. Misevic, C. Adami, L. Altenberg, J. Beaulieu, P. J. Bentley, S. Bernard, G. Beslon, D. M. Bryson, N. Cheney, P. Chrabaszcz, A. Cully, S. Doncieux, F. C. Dyer, K. O. Ellefsen, R. Feldt, S. Fischer, S. Forrest, A. Frénoy, C. Gagné, L. K. L. Goff, L. M. Grabowski, B. Hodjat, F. Hutter, L. Keller, C. Knibbe, P. Krcah, R. E. Lenski, H. Lipson, R. MacCurdy, C. Maestre, R. Miikkulainen, S. Mitri, D. E. Moriarty, J. Mouret, A. Nguyen, C. Ofria, M. Parizeau, D. P. Parsons, R. T. Pennock, W. F. Punch, T. S. Ray, M. Schoenauer, E. Schulte, K. Sims, K. O. Stanley, F. Taddei, D. Tarapore, S. Thibault, R. Watson, W. Weimer, J. Yosinski, The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities, Artificial Life 26 (2020) 274–306.
[53] D. S. Research, Specification gaming: the flip side of AI ingenuity, `"https://medium.com/@deepmindsafetyresearch/`

specification-gaming-the-flip-side-of-ai-ingenuity-c85bdb0deeb4", 2020.
[54] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, in: Y. Bengio, Y. LeCun (Eds.), Proc. of the International Conference on Learning Representations (ICLR), 2014.
[55] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.-R. Müller, Explaining nonlinear classification decisions with deep taylor decomposition, Pattern Recognition 65 (2017) 211–222.
[56] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, M. Wattenberg, Smoothgrad: removing noise by adding noise, CoRR abs/1706.03825 (2017).
[57] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems 30, 2017, pp. 4765–4774.
[58] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee, From local explanations to global understanding with explainable ai for trees, Nature Machine Intelligence 2 (2020) 2522–5839.
[59] F. Hohman, H. Park, C. Robinson, D. H. P. Chau, SUMMIT: Scaling deep learning interpretability by visualizing activation and attribution summarizations, IEEE Transactions on Visualization and Computer Graphics 26 (2019) 1096–1106.
[60] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing higher-layer features of a deep network, Technical Report, Univeristé de Montréal (2009).
[61] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, in: Proc. of Advances in Neural Information Processing Systems, 2016, pp. 3387–3395.
[62] C. Olah, A. Mordvintsev, L. Schubert, Feature visualization, Distill 2 (2017) e7.
[63] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, C. Olah, Activation atlas, Distill 4 (2019) e15.
[64] L. Rieger, C. Singh, W. J. Murdoch, B. Yu, Interpretations are useful: penalizing explanations to align neural networks with prior knowledge, CoRR abs/1909.13584 (2019).
[65] S. Teso, K. Kersting, Explanatory interactive machine learning, in: Proc. of the Conference on AI, Ethics and Society (AIES) 2019, 2019, pp. 239–245.
[66] B. Kim, H. Kim, K. Kim, S. Kim, J. Kim, Learning not to learn: Training deep neural networks with biased data, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9012–9020.
[67] A. S. Ross, M. C. Hughes, F. Doshi-Velez, Right for the right reasons: Training differentiable models by constraining their explanations, in: Proc. of Joint Conference on Artificial Intelligence (IJCAI), 2017, pp. 2662–2670.
[68] W. J. Murdoch, P. J. Liu, B. Yu, Beyond word importance: Contextual decomposition to extract interactions from lstms, in: Proc. of International Conference on Learning Representations (ICLR), 2018.
[69] C. J. Anders, P. Pasliev, A. Dombrowski, K. Müller, P. Kessel, Fairwashing explanations with off-manifold detergent, in: Proceedings of the 37th International Conference on Machine Learning, (ICML) 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 314–323.
[70] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, K.-R. Müller, Layer-wise relevance propagation: an overview, in: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Springer LNCS 11700, 2019, pp. 193–209.
[71] M. Kohlbrenner, A. Bauer, S. Nakajima, A. Binder, W. Samek, S. Lapuschkin, Towards best practice in explaining neural network decisions with lrp, in: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–7.
[72] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, Evaluating the visualization of what a deep neural network has learned, IEEE Transactions on Neural Networks and Learning Systems 28 (2017) 2660–2673.
[73] C. J. Anders, D. Neumann, W. Samek, K.-R. Müller, S. Lapuschkin, Software for dataset-wide xai: From local explanations to global insights with Zennit, CoRelAy, and ViRelAy, CoRR abs/2106.13200 (2021).
[74] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 8024–8035.
[75] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural computation 10 (1998) 1299–1319.
[76] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, science 290 (2000) 2323–2326.
[77] F. Wu, Z. Hu, The LLE and a linear mapping, Pattern Recognition 39 (2006) 1799–1804.
[78] J. Chen, Y. Liu, Locally linear embedding: a survey, Artificial Intelligence Review 36 (2011) 29–48.
[79] L. v. d. Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605.
[80] L. McInnes, J. Healy, UMAP: uniform manifold approximation and projection for dimension reduction, CoRR abs/1802.03426 (2018).
[81] M. Meila, J. Shi, A random walks view of spectral segmentation, in: Proc. of the International Workshop on Artificial Intelligence and Statistics (AISTATS), 2001.
[82] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2002, pp. 849–856.
[83] U. von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (2007) 395–416.
[84] S. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (1982) 129–137.
[85] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise., in: Proc. of the SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), volume 96, 1996, pp. 226–231.
[86] M. Everingham, L. Gool, C. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge results, "http://host.robots.ox.ac.uk/pascal/VOC/voc2007/workshop/everingham_cls.pdf" (2007).
[87] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of eugenics 7 (1936) 179–188.
[88] K. Fukunaga, Instruction to Statistical Pattern Recognition, Elsevier, 1972.
[89] B. Schölkopf, S. Mika, C. J. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, A. J. Smola, Input space versus feature space in kernel-based methods, IEEE transactions on neural networks 10 (1999) 1000–1017.
[90] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556 (2014).
[91] A. Krizhevsky, Learning multiple layers of features from tiny images, Master's thesis, University of Toronto, Department of Computer

Science, 2009.
[92] M. J. Kusner, J. R. Loftus, C. Russell, R. Silva, Counterfactual fairness, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 4066–4076.
[93] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (2017) 115–118.
[94] S. Lapuschkin, A. Binder, K.-R. Müller, W. Samek, Understanding and comparing deep neural networks for age and gender classification, in: Proc. of the IEEE International Conference on Computer Vision (ICCV) Workshops, 2017, pp. 1629–1638.
[95] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 39–57.

Appendix A. Neural Network Architecture and Training Setups

Appendix A.1. CIFAR-10 Training

The simple convolutional model used to train CIFAR-10 in 3.1 consists of two ReLU-activated convolutional-pooling blocks (filter sizes 16 and 32), followed by two dense layers (512 and 10 outputs, respectively). The model is trained for 5 epochs using SGD with a learning rate of 0.01 and a momentum of 0.9.

Appendix A.2. Colored MNIST Training

All models on colored MNIST in Sections 3.2 and 4.1 are trained using the AdaDelta algorithm with a learning rate of 1.0, which is multiplied by 0.7 after each epoch, for 10 epochs. The *a posteriori ClArC* is trained for 10 epochs on top of the *baseline model*, which has also been trained for 10 epochs. The network consists of 2 convolutional layers, followed by a max-pooling, and finally 2 fully connected layers. Dropout is used after the max pooling and after the first fully connected layer, with 25 percent and 50 percent dropout probabilities respectively. ReLU activations follow all linear layers except the final one.

The model used for 5.1 is trained with SGD, a learning rate of 0.001 for 5 epochs. The architecture, however, is the same as for the other colored MNIST models.

Appendix A.3. A-ClArC on ImageNet

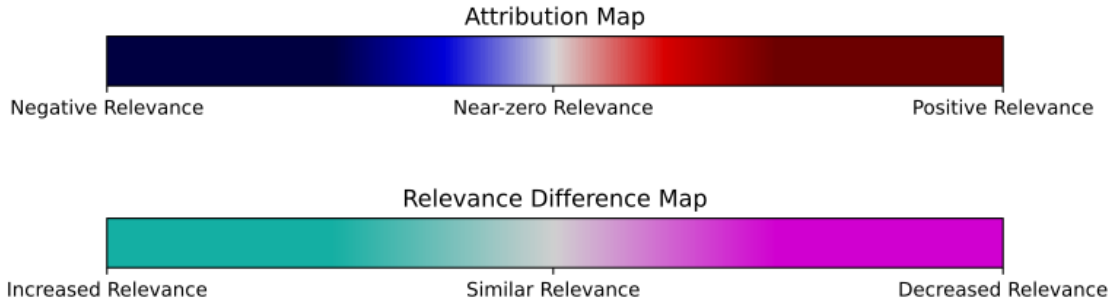
In Sections 4.2, 4.3 we employ A-ClArC using a VGG-16 model with the pretrained weights obtained from the Pytorch model zoo. For the input space A-ClArC experiment, we use an Adam optimizer with learning rate 0.0001 for fine-tuning. During feature spaceA-ClArC, an SGD optimizer with learning rate 0.001 and momentum 0.9 is applied. In both cases, we fine-tune over 10 epochs.

Appendix A.4. P-ClArC on ISIC 2019 and Adience Training

We again employ the VGG-16 model in Sections 5.3, 5.4 with the pretrained weights obtained from the Pytorch model zoo to train on both ISIC 2019 and Adience datasets, replacing the last fully connected layer of the classifier to fit the number of classes, i.e., 9 and 2, respectively. Both models are then trained over 100 epochs, using an SGD optimizer with learning rate 0.001 and momentum 0.9.

Appendix B. Supplementary Figures

Appendix B.1. Color Legend



Supplementary Figure B.1: Color Legend for the Attribution and Relevance Difference Maps employed in this work.

Acronyms

A-ClArC Augmentative Class Artifact Compensation. 3, 12, 13, 14, 15, 16, 21, 22, 23, 24, 25, 27, 28, 29, 31, 43, 44, 45, 46, 47, i

AI Artificial Intelligence. 4

AUC Area Under Curve. 16, 17, 19

BD Backdoor. 2, 4, 5, 6, 15, 16, 17, 44, 45

CAV Concept Activation Vector. 3, 6, 7, 13, 14, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 45, 46

CD Contextual Decomposition. 6

CDEP Contextual Decomposition Explanation Penalization. 6, 43

CH Clever Hans. 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47

ClArC Class Artifact Compensation. 2, 3, 6, 7, 10, 12, 14, 15, 23, 43, 45, 47

DNN Deep Neural Network. 1, 3, 4, 6, 7, 29

FDA Fisher Discriminant Analysis. 8, 9, 10, 16, 19, 20, 21

LLE Locally-linear Embedding. 7, 8

LNTL Learning not to Learn. 6, 43

LRP Layer-wise Relevance Propagation. 4, 7, 10, 15, 17, 20, 27, 28, 31, 32, 34, 35, 37, 38, 39, 40, 46, 47

ML Machine Learning. 2, 4, 36

P-ClArC Projective Class Artifact Compensation. 3, 13, 14, 15, 16, 21, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, i

ROC Receiver Operating Characteristic. 16, 17, 19

RRR Right for the Right Reasons. 6, 43

SC Spectral Clustering. 8, 9

SGD Stochastic Gradient Descent. 12, 13, i

SpeSig Spectral Signature. 6, 16, 17

SpRAy Spectral Relevance Analysis. 2, 3, 5, 7, 8, 9, 10, 15, 16, 17, 18, 19, 20, 21, 22, 27, 28, 31, 32, 33, 35, 43, 44, 45, 46, 47

t-SNE t-distributed Stochastic Neighbor Embedding. 8, 46

UMAP Uniform Manifold Approximation and Projection. 8, 10, 19, 20

XAI eXplainable Artificial Intelligence. 2, 3, 4, 5, 7, 16, 43, 44, 45, 46, 47

XIL eXplanatory Interactive Learning. 5, 6, 43