Information Fusion 103 (2024) 102094

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Preemptively pruning Clever-Hans strategies in deep neural networks

Lorenz Linhardt [a,b], Klaus-Robert Müller [a,b,c,d,e,*], Grégoire Montavon [f,b,a,*]

[a] Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany
[b] Berlin Institute for the Foundations of Learning and Data – BIFOLD, 10587 Berlin, Germany
[c] Department of Artificial Intelligence, Korea University, Seoul 136-713, South Korea
[d] Max Planck Institute for Informatics, 66123 Saarbrücken, Germany
[e] Google DeepMind, Berlin, Germany
[f] Department of Mathematics and Computer Science, Freie Universität Berlin, 14195 Berlin, Germany

ARTICLE INFO

Keywords:
Clever Hans effect
Model refinement
Pruning
Explainable AI
Deep neural networks

ABSTRACT

Robustness has become an important consideration in deep learning. With the help of explainable AI, mismatches between an explained model's decision strategy and the user's domain knowledge (e.g. Clever Hans effects) have been identified as a starting point for improving faulty models. However, it is less clear what to do when the *user and the explanation agree*. In this paper, we demonstrate that acceptance of explanations by the user is not a guarantee for a machine learning model to be robust against Clever Hans effects, which may remain undetected. Such hidden flaws of the model can nevertheless be mitigated, and we demonstrate this by contributing a new method, Explanation-Guided Exposure Minimization (EGEM), that *preemptively* prunes variations in the ML model that have not been the subject of positive explanation feedback. Experiments demonstrate that our approach leads to models that strongly reduce their reliance on hidden Clever Hans strategies, and consequently achieve higher accuracy on new data.

1. Introduction

Machine learning (ML) models such as deep neural networks have been shown to be capable of converting large datasets into highly nonlinear predictive models [5–10]. As ML systems are increasingly being considered for high-stakes decision making, such as autonomous driving [11] or medical diagnosis [12–16], building them in a way that they reliably maintain their prediction accuracy on new data is crucial.

Proper data splitting and evaluation on hold-out test sets have long been recognized as an essential part of the validation process (e.g. in [17]), but unfortunately, such techniques cannot detect all flaws of a model [18–21]. Misspecified loss functions, domain shifts, spurious correlations, or biased datasets can potentially compromise attempts to build well-generalizing models, without altering the measured accuracy. Spurious correlations – i.e. correlations that do not generalize – of input variables with the label are a common threat [22]: If a model learns to use a spurious correlation as part of its decision strategy, also known as the Clever Hans (CH) effect [19] or shortcut learning [21], its performance will drop on data where this fake correlation ceases to hold. For example, if a bird classification model learns to recognize a species based on the background, due to a spurious correlation in the training data, it will *not* be accurate when the bird is depicted in an untypical environment [23]. In real-world scenarios, e.g. medical applications, a failure to address these more elusive flaws might lead to catastrophic failures, as has been demonstrated numerous times (e.g. [14,19,24,25]). This has spurred efforts to find potential causes of such failures (e.g. [19,26–28]).

Explainable AI (XAI) [29–33] is a natural starting point for robustification beyond classical hold-out validation because it places human experts in the loop. As demonstrated e.g. in [19,34–36], an expert can scrutinize the model's decision strategy from the produced explanation, possibly identifying CH strategies, and remove them subsequently [37]. However, as discussed e.g. in [38], there is no guarantee that a model that passes the XAI test can be deployed safely. Specific data points where the CH strategy reveals itself may be missing at validation time, thereby leaving the expert with the false impression that the model is free of CH effects. This is likely to happen in practice, e.g. when the ML practitioner is not training their own model from scratch but relies instead on a model trained by a third-party (e.g. *foundation models*[1]) and does not have full access to the training data.

In this paper, we tackle for the first time the problem of undetected CH strategies (i.e. the case where the model remains flawed in

* Corresponding authors.
E-mail addresses: l.linhardt@campus.tu-berlin.de (L. Linhardt), klaus-robert.mueller@tu-berlin.de (K.-R. Müller), gregoire.montavon@fu-berlin.de (G. Montavon).

[1] Foundation models [1] are pretrained multi-purpose models typically made available by a third party (e.g. [2–4]).

https://doi.org/10.1016/j.inffus.2023.102094
Received 8 April 2023; Received in revised form 20 October 2023; Accepted 23 October 2023
Available online 4 November 2023

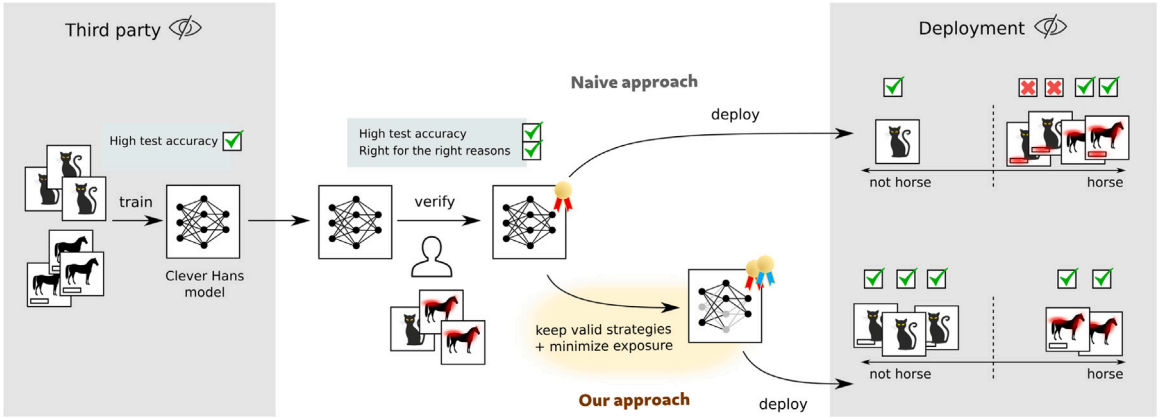

Fig. 1. Comparison of a naive XAI-based validation/deployment pipeline and our proposed approach incorporating an additional exposure minimization step. *Left:* A third party trains a flawed (Clever Hans) model which exploits a spurious correlation (images of the horse class have a copyright tag in the bottom-left corner). *Middle:* The user receives the model. Because the user has limited data (in particular, no data with copyright tag), the flaw of the model cannot be easily detected with XAI methods and the model appears to be both accurate and right for the right reasons. *Right:* Because of the undetected flaw, a naive deployment of the model is likely to result in prediction errors on new data (e.g. cats with copyright tags predicted to be horses). Our proposed approach preemptively reduces exposure to unseen features (the copyright tag), thereby avoiding these incorrect predictions.

spite of a full agreement of the human with the model's predictions and the generated explanations). We contribute a novel algorithm, called Explanation-Guided Exposure Minimization (EGEM), which distills from the original model a *refined model* with lower overall exposure to input features but that also preserves the few valid prediction strategies contained in the generated explanations. With mild approximations, the optimization problem embodied in our EGEM approach reduces to a simple soft-pruning rule, which can be easily implemented in a broad range of neural network architectures including convolutional networks and transformers. Our proposal, as well as the context in which it operates, are illustrated in Fig. 1.

To evaluate our approach, we simulate several scenarios of a user receiving a third-party model and possessing a subset of the data on which no CH strategies can be detected by classical XAI pipelines (e.g. LRP/SpRAy [19,39]). Results on image and text data demonstrate that our proposed EGEM approach (and our extension PCA-EGEM) delivers models with a much lower reliance on CH strategies, thereby achieving more stable prediction accuracy, especially when considering data with spurious features. Our approach furthermore outperforms a number of existing and contributed baselines.

2. Related work

In this section, we present related work on validating ML models that goes beyond classical validation techniques such as holdout or cross-validation [17] in order to address statistical artifacts such as domain shifts and spurious correlations. We make a distinction between methods relying on Explainable AI and users' explanatory feedback (Section 2.1), and a broader set of methods addressing domain shift and spurious correlations by statistical means (Section 2.2).

2.1. Explainable AI and Clever Hans

Explainable AI (XAI) [30,32,33,40–42] has been a major development in machine learning which has enabled insights into a broad range of black-box ML models. It has shown to be successful at explaining complex state-of-the-art neural networks classifiers [34,39,43–46], as well as regressors [47] and a broader set of ML techniques such as unsupervised learning (e.g. [48–50]). While most XAI methods generate an explanation for individual instances, solutions have been proposed to aggregate them into dataset-wide explanations that can be concisely delivered to the user [19]. In contrast to these works, we are not concerned with improving XAI techniques themselves, e.g. the interpretability of explanations, but rather with using these techniques to make models more robust against spurious correlations.

More closely related to our aims, several techniques have applied XAI for the purpose of revealing CH features in ML models [19,35,36,51]. Knowledge about the CH features can be used to desensitize the model to these features (e.g. via retraining [35] or layer-specific adaptations [37]). If ground-truth explanations are available (e.g. provided by a human expert), the model may be regularized to match these explanations [52–54], e.g. by minimizing the error on the explanation via gradient descent. Such adaptations to the users' expectations have also been shown to be effective in interactive settings [55,56]. Our approach differs from these works as we address the case where the available data does not contain CH features, hence making them indiscoverable by the techniques above.

A further approach is DORA [57], which attempts to find potential CH features in a data-agnostic way and subsequently uses the discovered candidate features to detect faulty decision strategies at deployment time by identifying samples that are outliers in the DNN's internal representation space. In contrast, we attempt to *robustify* the network with no need for further post-processing at deployment. Furthermore, we examine the scenario where a limited amount of clean data is available, allowing us to employ conceptually different criteria besides outlierness.

2.2. Robustness to spurious correlations

Our work is part of a larger body of literature concerned with *domain* or *covariate shift* and how to design models robust to it. Yet, it concerns itself with a decidedly specialized and rather recent part of this area: unlearning or avoiding the use of spurious features in deep neural networks. Previous work attempting to create models that are robust against spurious correlations approached the problem from the angle of optimizing worst-group loss [23,58–65]. This approach has been shown to be effective in reducing reliance on CH features. Yet, these methods require access to samples containing the CH features and a labeling of groups in the data induced by these features. In particular, as previously pointed out by Kirichenko et al. [63], Group-DRO (distributionally robust optimization) [59], subsampling approaches [60,62] and DFR (deep feature reweighting) [63] assume group labels on the training or validation data, and even methods that do away with these assumptions need to rely on group labels for hyper-parameter tuning [58,62,66]. Our setting is different from the ones above in that we assume that a pretrained model is to be robustified post hoc with

limited data and that data from the groups containing the CH feature are not available at all. We believe this is a highly relevant scenario, considering the increasing prevalence of pretrained third-party models that have been trained on datasets that are unavailable or too large to fully characterize.

3. Explanation-guided exposure minimization (EGEM)

Before presenting our main technical contribution, let us restate the key aspects of the application scheme studied in this paper (as depicted in Fig. 1):

(i) a pretrained model provided by a third party and potentially affected by a CH effect,
(ii) the unavailability of the original training data to the user, which prevents the discovery of CH features,
(iii) limited data which is available to the user to validate and refine the third-party model, and for which
(iv) the conclusion of the user is that the predictions and the associated decision strategies (as revealed by XAI) are correct. I.e. the data available to the user is free of CH features.

As argued before, in spite of the positive XAI-based validation outcome, there is no guarantee that the model's decision strategy remains correct in regions of the input space not covered by the available data (e.g. where the CH artifact could be expressed).

As a solution to the scenario above, we propose a preemptive model refinement approach, which we call *Explanation-Guided Exposure Minimization (EGEM)*. Technically, our approach is a particular form of knowledge distillation where the refined (or distilled) model should reproduce observed prediction strategies (i.e. predictions and explanations, hence *"explanation-guided"*) of the original model on the available data. At the same time, the refined model should *minimize* its overall sensitivity or *exposure* to variations in the input domain, so that uninspected (potentially flawed) decision strategies are not incorporated into the overall decision strategy.

Let the original and refined model have the same architecture but distinct parameters θ_{old} and θ. We denote by $f(\boldsymbol{x}, \theta_{\text{old}})$ and $f(\boldsymbol{x}, \theta)$ the predictions produced by the two models, and the explanations associated to their predictions as $\mathcal{R}(\boldsymbol{x}, \theta_{\text{old}})$ and $\mathcal{R}(\boldsymbol{x}, \theta)$ respectively. We then define the learning objective as

$$\min_{\theta} \ \mathbb{E}\left[\|\mathcal{R}(\boldsymbol{x}, \theta) - \mathcal{R}(\boldsymbol{x}, \theta_{\text{old}})\|^2 + \Omega_{\boldsymbol{x}}(\theta)\right] \tag{1}$$

where the expectation is computed over the available data, and where $\Omega_{\boldsymbol{x}}(\theta)$ is a function that quantifies the exposure of the model to the input variation in general, or in some neighborhood of $\boldsymbol{x}$. In the latter case, the neighborhood should be large enough to encompass instances outside the available data such as those encountered at deployment time.

Although the formulation of Eq. (1) is general, it is not practical, because it would require optimizing a highly nonlinear and non-convex objective. Moreover, the objective depends on explanation functions and on a complexity term that themselves may depend on multiple model evaluations, thereby making the optimization procedure intractable.

3.1. A practical formulation for EGEM

To make the concept of explanation-guided exposure minimization effective, we will restrict the class of XAI methods on which it depends to those (1) that can attribute onto any layer of the model and (2) whose produced scores have the following properties: the score assigned to a neuron i at a given layer should be decomposable in terms of the neurons j in the layer above, i.e. $R_i = \sum_j R_{ij}$ and terms of the decomposition should have the structure

$$R_{ij} = a_i \, \rho(w_{ij}) \, d_j \tag{2}$$

where a_i denotes the activation of neuron i, w_{ij} is the weight connecting neuron i to neuron j in the next layer, ρ is an increasing function satisfying $\rho(0) = 0$ (e.g. the identity function), and d_j is a term that only indirectly depends on the input activations and parameters in the given layer and that is reasonable to approximate as constant locally. Explanation techniques that produce explanation scores with such structure include propagation methods such as Layer-wise Relevance Propagation (LRP) [39,68] or its limit Gradient × Input (GI) [69] and Integrated Gradients (IG) [70]. (See Supplementary Note A for derivations.)

An advantage of expressing explanations in the form of Eq. (2) is that it gives access to higher-level abstractions (e.g. visual concepts) built by the network, which are often represented more naturally in these layers. If we further restrict the search for refined parameters to the weights of the same layer, one can then formulate an alternative to the optimization problem of Eq. (1) which is more expressive and also computationally more tractable:

$$\min_{w} \ \sum_{ij} \mathbb{E}\Big[\big(\overbrace{a_j \rho(w_{ij}) d_j}^{R_{ij}} - \overbrace{a_i \rho(w_{ij}^{\text{old}}) d_j}^{R_{ij}^{\text{old}}}\big)^2 + \overbrace{\lambda \cdot \big(\rho(w_{ij}) d_j\big)^2}^{\Omega_{ij}}\Big] \tag{3}$$

where, as for Eq. (1), the expectation is taken over the available data. Here, w_{ij}^{old} and w_{ij} denote the original and the refined weights. As in Eq. (1), the first squared term guides the explanations of the refined model to be close to that of the original model and the second squared term (Ω_{ij}) carries out the exposure minimization. The latter can be interpreted as the extent to which the refined model responds to the activation of neuron i through neuron j, in particular, if it becomes zero, the model becomes unresponsive to the activation of neuron i. The lack of direct dependence of Ω_{ij} on the input activations is instrumental for it to generalize outside the available data towards data encountered at deployment time. An advantage of the formulation of Eq. (3) is that it has the closed-form solution:

$$\forall_{ij} \ : \ w_{ij} = \frac{\mathbb{E}[a_i^2 d_j^2]}{\mathbb{E}[a_i^2 d_j^2] + \lambda \mathbb{E}[d_j^2]} w_{ij}^{\text{old}}. \tag{4}$$

See Supplementary Note B for a derivation. In other words, the refined model can be seen as a soft-pruned version of the original model where the pruning strength depends on how frequently and to what magnitude the input neuron is activated and how the model responds to the output neuron.

If we further assume that a_i and d_j are independent, then d_j vanishes from Eq. (4), leading to the simpler pruning rule

$$\forall_{ij} \ : \ w_{ij} = \frac{\mathbb{E}[a_i^2]}{\mathbb{E}[a_i^2] + \lambda} w_{ij}^{\text{old}}. \tag{5}$$

Lastly, because the pruning coefficients only depend on the input neuron, the same pruning can be achieved by keeping the weights intact and inserting a layer directly after the input activations that performs the scaling:

$$\forall_i \ : \ a_i \leftarrow a_i c_i \tag{6}$$

with $c_i = \mathbb{E}[a_i^2]/(\mathbb{E}[a_i^2] + \lambda)$. The pruning of the neural network architecture and the resulting loss of dependence on the CH feature are depicted in Fig. 2 (top). Due to the soft character of the pruning, i.e. $c_i \in [0, 1]$, EGEM is able to make more subtle changes to the network than explanation-based hard-pruning approaches, such as the one by Yeom et al. [71].

The same Eq. (6) can also be applied to convolutional layers. To calculate the scaling parameters c_i, the activations of each channel are summed up along the spatial dimensions. For refinement, the pruning coefficients are then applied to all activations of the corresponding feature map (cf. Eq. (6)). Such pruning strategy for convolutional layers can be derived exactly from Eq. (3) if assuming activation maps of infinite size (or circular convolutions) and stride 1. For the majority of convolution layers used in practice, Eq. (6) only derives from the objective formulation approximately.

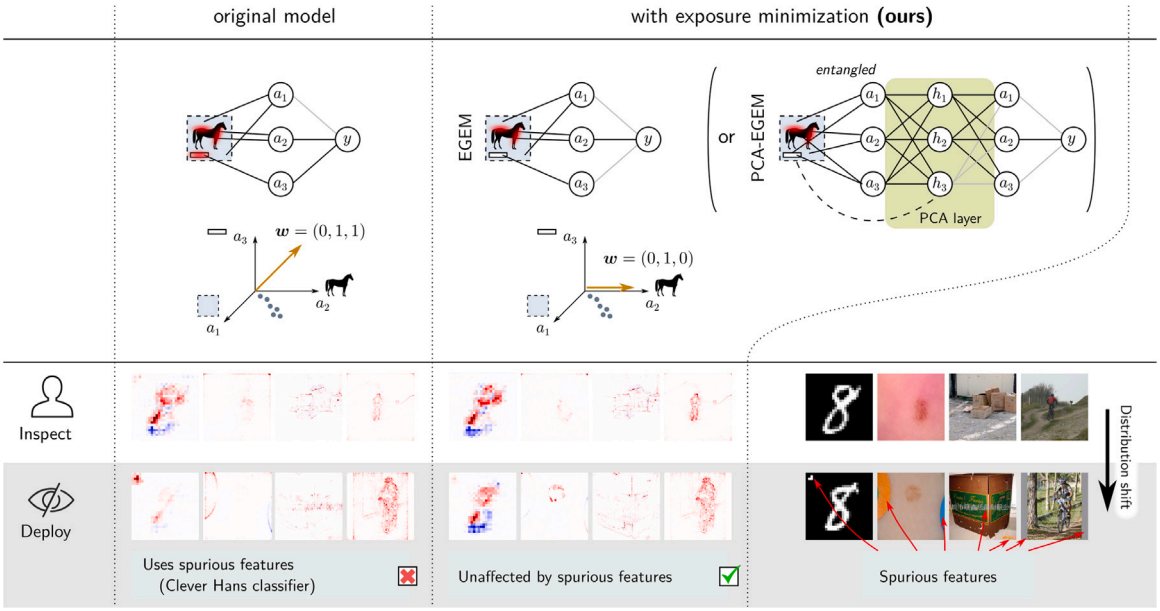

Fig. 2. *Top:* Cartoon depicting the removal of unseen Clever Hans strategies via our proposed exposure minimization approaches (EGEM and PCA-EGEM). The refined model only retains the dependence on a_2 (a neuron detecting the actual horse) and removes its reliance on a_3 (a neuron responsive to spurious copyright tags). *Bottom:* Qualitative behavior of PCA-EGEM on ML models trained on real datasets. The models produced by our approach become robust to spurious features not seen at inspection time but occurring at deployment time. (Pixel-wise explanations are computed using the `zennit` package [67]).

3.2. Pruning in PCA space

Within the EGEM soft-pruning strategy, each dimension of a layer is pruned individually. In practice, this only allows to eliminate undetected flawed strategies that use a set of neurons that is disjoint from neurons supporting the validated strategies. Because a given neuron may in practice contribute to both, the standard version of EGEM would not be able to carry out the exposure minimization task optimally. To address this limitation, we propose PCA-EGEM, which inserts a virtual layer, mapping activations to the PCA space (computed from the available data) and back (cf. Fig. 2). PCA-EGEM then applies soft-pruning as in Eq. (6), but in PCA space, that is:

$$\begin{aligned} h_k &= U_k^\top (\boldsymbol{a} - \bar{\boldsymbol{a}}) \\ h_k &\leftarrow h_k c_k \\ \boldsymbol{a} &\leftarrow \textstyle\sum_k U_k h_k + \bar{\boldsymbol{a}} \end{aligned} \tag{7}$$

with $c_k = \mathbb{E}[h_k^2]/(\mathbb{E}[h_k^2] + \lambda)$. Here, $\{U_k\}_{k=1}^K$ is the basis of PCA eigenvectors and $\bar{\boldsymbol{a}}$ is the mean of the activations over the available data. The motivation for such mapping to the PCA space is that activation patterns that support observed strategies will be represented in the top PCA components. PCA-EGEM can therefore separate them better from the unobserved strategies that are likely not spanned by the top principal components.

While using PCA to find principal directions of interpretable features has been proposed previously by Härkönen et al. [72] for GANs [73] and has found application beyond that [74,75], to our knowledge its use for the purpose of identifying a basis for exposure minimization is novel.

4. Experimental evaluation

In this section, we evaluate the efficacy of the approaches introduced in Section 3 on datasets that either naturally contain spurious correlations, giving rise to Clever Hans decision strategies, or have been modified to introduce such correlations. Each experiment involves three datasets for which we ensure the following structure:

(i) the **training data**, used for training the original model, containing CH features, and resulting in a model with a CH decision strategy. The training data is for its larger part not available to the user.
(ii) the **available data**, much smaller than the training data, and *not* containing any CH feature. It is the only data available to the user for validating and refining the model.
(iii) the **test data**, disjoint from (i) and (ii), and used to evaluate model refinement strategies. It is poisoned to contain the CH feature to 0% or 100% on all classes. Unlike in the training data, the CH feature is thus decorrelated from the target class.

After introducing the datasets, we demonstrate that the proposed approaches can mitigate the CH effect learned by pretrained models. Additionally, we empirically explore the effect of the number of samples used for refinement and discuss the challenges of hyper-parameter selection in Sections 4.5 and 4.6, as well as the impact of the type of CH feature in Section 5. A more qualitative evaluation on the CelebA dataset [76] follows in Section 6 and an application to sentiment classification is described in Section 7.

4.1. Datasets

We introduce here the datasets used to evaluate the proposed methods in Sections 4.4–4.6: a modified version of the MNIST dataset [77], the ImageNet dataset [78,79], and the ISIC dataset [80–82]. Details on the preprocessing and the neural networks used for each dataset can be found in the Supplementary Notes. Dataset statistics can be found in Table 1.

Modified MNIST. We create a variant of the original MNIST dataset [77] in which digits of the class '8' are superimposed with a small artifact in the top-left corner (see Fig. 2). In order to generate a natural yet biased split of the training data that separates an artifact-free set of data available for refinement, we train a variational autoencoder [83] on this modified dataset and manually chose a threshold along a latent dimension such that the samples affected with the artifact only fall on one side. This defines a subset of 39,942 samples from which clean datasets are sampled for refinement and leaves a systematically

Table 1
Overview of datasets, their class structure (class with CH feature in **bold**), a description of the CH feature, and the number of instances in each set of data. p(CH) is the fraction of training samples of the affected class containing the CH feature.

Dataset	# classes	Class names	CH feature	p(CH)	N_{train}	$N_{\text{available}}$	N_{test}
MNIST	10	{0, 1, 2, 3, 4, 5, 6, 7, **8**, 9}	3-pixel corner	0.700	60,000	700	10,000
ISIC2019	8	{**Melanocytic nevus**, ...}	colored patch	0.015	22,797	700	2,534
ImageNet subtask 1	2	{**carton**, crate}	watermark, www	0.438	1.2M	700	100
ImageNet subtask 2	2	{**carton**, envelope}	watermark, www	0.438	1.2M	700	100
ImageNet subtask 3	2	{**carton**, packet}	watermark, www	0.438	1.2M	700	100
ImageNet subtask 4	2	{**mountain bike**, bicycle-built-for-two}	gray frame	0.028	1.2M	700	100

biased subset (containing all modified '8' samples) only accessible during training. We train a neural network (2 convolutional and 2 fully connected layers) on this dataset using binary cross-entropy loss over all ten classes on the whole training data.

ISIC. The ISIC 2019 dataset [80–82] consists of images containing skin lesions that are associated with one of eight medical diagnoses. We fine-tune a neural network based on a VGG-16 pretrained on ImageNet for this classification task using a cross-entropy loss. Some images of the class 'Melanocytic nevus' are contaminated with colored patches (see Fig. 2), which have been recognized as potential CH feature [37,53,84,85]. We manually remove all contaminated images after training and use this clean dataset for refinement. Images in the test set are contaminated at the desired ratio by pasting one extracted colored patch onto other images.

ImageNet. We use the ILSVRC 2012 subset of the ImageNet dataset [78,79]. Previous work has identified multiple spurious correlations potentially affecting a model's output [37]. From these known spurious features we select two that lead to an easily reproducible CH effect in popular pretrained models. In particular, we use a watermark and web-address on images of the 'carton' class and a gray frame around images of the 'mountain bike' class as CH features (see Fig. 2). We vary their frequency in the test set by pasting these features on images (details in Supplementary Note C). The selected classes are evaluated in a binary classification setting against the most similar classes in terms of the output probabilities. Training set images used for refinement that do not contain the CH feature are manually selected for the 'carton' experiments and automatically for 'mountain bike' experiment. For experiments on this dataset, we make use of the pretrained ResNet50 [86] (for the 'carton' class) and VGG-16 [87] (for the 'mountain bike' class) networks available in pytorch.[2]

4.2. Methods

We evaluate both EGEM and PCA-EGEM and compare them to several baseline methods for the mitigation of the Clever Hans effect.

(i) **Original:** This is the original model without any modifications. Note that approaches for mitigating the Clever Hans effect, such as [35,37] reduce technically to this simple baseline because they only modify the model in presence of *detected* CH features, whereas in our scenario, no CH features are detectable in the available data.

(ii) **RGEM:** This baseline, which we contribute, is a modification of EGEM, where the exposure minimization is carried out under the constraint of preserving model *response* instead of model *explanation*. Specifically, RGEM optimizes the objective:

$$\min_{\theta} \quad \mathbb{E}[(f(\boldsymbol{x},\theta) - f(\boldsymbol{x},\theta_{\text{old}}))^2] + \lambda\|\theta\|^2 \tag{8}$$

with f being the neural network function and θ the parameters in the last layer. Compared to EGEM, RGEM can only refine the last layer weights.

(iii) **Ridge:** This baseline consists of replacing the last layer weights of the original model by weights learned via ridge regression on the available data. It is equivalent to linear probing or deep feature reweighting [63], which has been shown to be effective in mitigating accuracy loss due to subpopulation shifts [89] and the Clever Hans effect when hyper-parameter selection based on worst-group accuracy optimization is possible [63]. The formulation for Ridge can also be retrieved by replacing the output of the original model, $f(\boldsymbol{x},\theta_{\text{old}})$, in the formulation of RGEM (see Supplementary Note D) with the ground-truth labels.

(iv) **Retrain:** This baseline corresponds to a deeper retraining, where starting from the original model, layers are fine-tuned to improve the classification of the available data. Unlike the RGEM and Ridge baselines, all layers are updated to fit the available data. This makes fine-tuning the most flexible but also computationally costly approach of all the compared methods.

4.3. Evaluation setup

We evaluate all methods along two dimensions: in-distribution accuracy, which is measured as classification accuracy on CH-free (clean) data, and robustness against the CH effect. To evaluate the latter, we generate artificially a fully poisoned test set by adding the CH artifact to all test images of all classes. Artificially poisoned data allows us to isolate the CH effect, as it is identical to the clean test data, except for the CH feature. This setup allows us to attribute any difference in accuracy between these two (clean and manipulated) test sets to the presence of the CH feature. In this scenario, the correlation of the CH feature and the target class breaks. Such a distribution shift could, for example, happen in medical applications where a classifier might be trained on data in which the mode of data collection or the population characteristics of subjects are correlated with the outcome, but where this spurious correlation does not hold in the general case [14]. Note that while this poisoning scenario is an extreme case, it is not the worst case, as the class that was contaminated during training will also be modified with artifacts during testing.

For refinement, 700 correctly predicted instances per class are used, oversampling images if fewer than 700 correctly predicted samples are in the available data to maintain a balance between classes and ensure that all classes have equal contribution to the refinement processes. For the modified MNIST and the ISIC dataset, we use 1000 randomly chosen test samples for each run of the evaluation, for ImageNet we use all available validation samples. We evaluate the various models for all tasks under 0% and 100% uniform poisoning. Classification accuracy for intermediate levels of poisoning can be obtained by linear interpolation of these extremes.

4.4. Results

Fig. 3 shows the obtained accuracy under 0% and 100% poisoning. An ideal model would obtain high accuracy with only a very small difference between clean and poisoned data. It should be invariant to the spurious feature and at most react to possible interference with other features, e.g. the spurious feature being pasted on top of a relevant part of the image, while not losing accuracy on the clean data. As expected, across all datasets increased poisoning reduces the accuracy of the original model. *Importantly, this drop in accuracy cannot be detected without access to samples containing the CH feature.*

[2] www.pytorch.org [88].

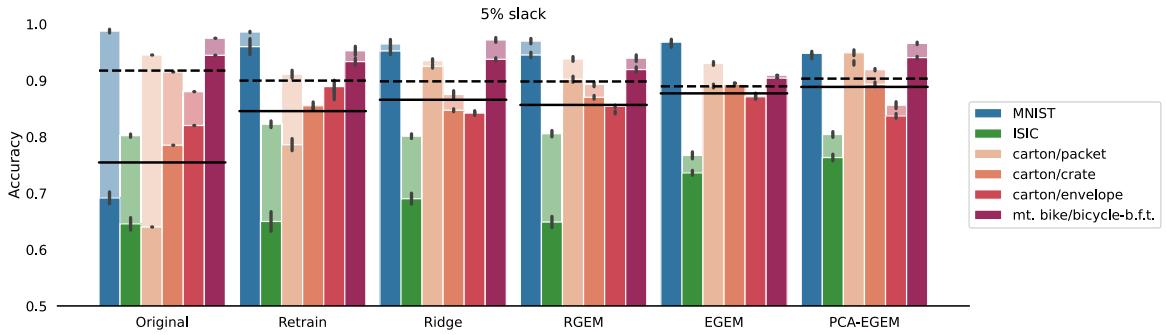

Fig. 3. The accuracy for 0% (lighter shade) and uniform 100% (darker shade) poisoning with the spurious feature. The last four bars for each method refer to the binary tasks constructed from the ImageNet dataset. Solid lines show average 100%-poisoned accuracy and dashed lines show average clean-data accuracy over all datasets. The results shown are the mean accuracy and standard deviation obtained using 700 refinement samples per class on the respective test sets over five runs.

Modified MNIST. On the modified MNIST dataset, the original model loses about 30% of its clean-data accuracy when evaluated at the 100% poisoning level. All other models achieve both clean-data and 100%-poisoned accuracy levels within 4% of the original model's clean-data accuracy. While explanation-based methods lose slightly more clean-data accuracy than the other baselines, they display virtually no gap between clean-data accuracy and 100%-poisoned accuracy, making them the most predictable when no poisoned data is available.

ISIC. On the more complex ISIC dataset, it can be observed that exposure to the CH feature cannot be completely removed by any of the methods. EGEM and PCA-EGEM still provide fairly robust models, with the highest poisoned-data accuracy and the smallest gap between 0%-poisoned and 100%-poisoned accuracy. PCA-EGEM retains clean-data accuracy while being the only method improving poisoned-data accuracy by more than 10 percentage points. The dataset provides a challenge for all other methods. Even though Retrain is the only method that improves clean-data accuracy in the refinement process, its poisoned-data accuracy is virtually the same as the original model's, indicating that the absence of a feature in the available data is not enough to remove it from the model, given only a limited amount of samples.

ImageNet. On the ImageNet tasks containing the 'carton' class, PCA-EGEM is the most robust refinement method. It is only outperformed in the 100%-poisoned setting of the 'carton/envelope' task, where Retrain achieves the highest clean-data and poisoned accuracy. As we will see in Section 4.5, the inferior 100% poisoning accuracy of PCA-EGEM is a result of the hyper-parameter selection procedure and not fundamentally due to the pruning-based nature of the method. On the 100%-poisoned setting 'mountain bike' task, no refinement method is able to achieve accuracy gains over the original model. This might be due to the small magnitude of the CH effect resulting in the clean-data loss due to refinement outweighing the robustness gain. This case also demonstrates that refinement is not beneficial in all scenarios and might not even lead to an improved 100%-poisoned accuracy. Whether or not to refine should be decided based on whether the loss of clean-data accuracy can be tolerated.

We hypothesize that the reason for Ridge and RGEM being sub-optimal for some of the datasets is that these approaches can only manipulate the last layer of the network, which may not be where the CH feature is expressed most clearly [37,50,57]. An analysis of the per-layer separability of clean and manipulated images supports this by showing that in particular for the ISIC dataset and the 'mountain bike' task, separability is highest at layers close to the input (see Supplementary Note J).

Another concern that should be taken into consideration when applying Ridge, Retrain, or RGEM is that those approaches may learn new spurious correlations on the limited available data. While we separated the available data into 80% training and 20% validation splits, newly emerging CH effects may still go unnoticed. Since (PCA-)EGEM can only *reduce* the sensitivity of the network, it precludes the learning of new, potentially CH, features.

Overall, the results in this section demonstrate that the proposed refinement methods can preemptively robustify a pretrained model against Clever Hans effects, even if the latter cannot be observed from the limited available data. We could clearly establish that the attempt to robustify against CH behavior in absence of the associated artifact or knowledge thereof is not a hopeless endeavor and can be addressed with relatively simple methods. Yet, the trade-off between clean-data accuracy and poisoned-data accuracy cannot be directly observed on real data and thus needs to be resolved heuristically. We explore this aspect in the next section.

4.5. Hyper-parameter selection

The hyper-parameters optimized in the experiments in this section are the number or epochs for 'Retrain' and the regularization factor λ for all other refinement methods. For the deep exposure-based approaches, EGEM and PCA-EGEM, we do not optimize λ for each layer directly, but we rather employ an approach inspired by the triangular method of Ashouri et al. [90] and earlier work [91] where pruning strength increases with the layer index, which allows us to reduce the number of parameters to optimize to one. In particular, we define thresholds τ_l that denote the desired average pruning factor per layer, where l is the layer index within the set of L layers to be refined:

$$\tau_l = 1 - (1 - \alpha) \times \frac{l - 1}{L - 1}$$

and optimize α. λ_l is then set such that the average pruning factor $\mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}} \mathbb{E}_j c_j$ from Eq. (4) for layer l is at most τ_l. The search for λ_l given τ_l can be easily implemented as exponential search.

Ideally, the hyper-parameters should be set such that classification loss is minimized while exposure to the spurious artifact is negligible. While classification loss on clean data can be readily approximated by evaluating the loss function on the available data, exposure to the spurious artifact is a more elusive quantity and cannot be measured without prior knowledge of the spurious artifact. In previous work (e.g. [23,58–64]) it is assumed that for each class a set of samples with and without the spurious artifact is given and in most cases that the *worst-group-accuracy* can be directly optimized or at least used for hyper-parameter selection, circumventing this problem. Since in our problem setting access to samples with the artifact is not given, this metric for parameter selection is not available and we need to establish a heuristic approach. Assuming that the classification loss on clean data can be approximated accurately, one option is to pick the strongest refinement hyper-parameter (i.e. highest number of epochs, largest λ or smallest α) from a pre-defined set (see Supplementary Note F) for

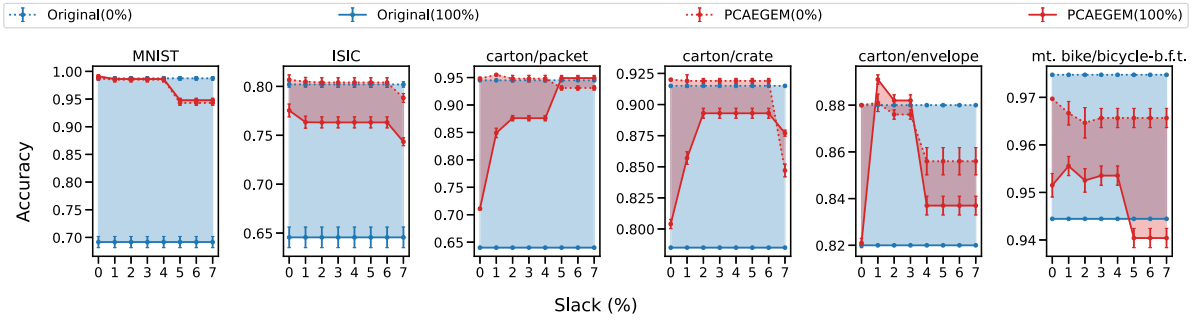

Fig. 4. Accuracy under variations of the slack parameter. Higher slack means higher refinement-data loss is accepted when selecting the refinement hyper-parameter. The dotted line indicates clean-data accuracy whereas the solid line indicates 100%-poisoned data accuracy. Mean and standard deviation are computed over 5 runs.

which the validation accuracy after refinement is at least as high as the one achieved by the original model.

As it is possible that strong refinement also impairs the use of generalizing features, there may be a trade-off between clean-data accuracy and robustness to spurious features. That optimizing overall clean-data accuracy is generally not the best approach to optimizing overall accuracy is highlighted by the fact that other works optimize *worst-group-accuracy*, as mentioned above. We explore the accuracy trade-off in Fig. 4 by introducing a 'slack' parameter s to the hyper-parameter selection for PCA-EGEM. We refer to Supplementary Note G for the results of all methods. The refinement hyper-parameter is then chosen as the strongest regularization, given that the validation accuracy is at most s% smaller than the one achieved by the original model. The idea is that minimizing loss of classification accuracy on the available data prevents removing too much exposure to useful features, yet, allowing for some slack counteracts the tendency to choose trivial least-refinement solutions.

We suspect that in the simple case of the modified MNIST dataset, the model only learned few important high-level features and that the CH feature is close to being disentangled in some layer of the network. This scenario is a natural fit for pruning methods which could simply remove the outgoing connections of the node corresponding to the CH feature. Stronger refinement risks pruning useful features as well, which is an effect that can be observed in Fig. 4. For most datasets, we can observe that the accuracy curves first converge to or maintain a minimal 0%–100% poisoning gap. In this regime, PCA-EGEM prunes unused or CH features. After crossing a certain level of slack, both accuracy values deteriorate as features necessary for correct classifications are being pruned as well.

The results previously presented in Fig. 3 are the outcomes for $s = 5\%$. This is a heuristic and it can be seen from Fig. 4 that different values of slack may be beneficial to increase robustness, depending on the dataset. We also show in Supplementary Note G and H that PCA-EGEM provides the most robust refinement over a large range of slack values. As slack cannot be optimized w.r.t. the true deployment-time accuracy, we propose to set s between 1% and 5% as a rule of thumb based on these experiments, where smaller data samples may permit lower slack values.

In principle, another hyper-parameter is the choice of layers to refine. Knowledge of the type of Clever Hans could potentially guide this choice [37,92] as the layer in which a concept is best represented may differ across concepts [93]. Since we do not assume such knowledge in our experiments, we simply refine the activations after every ResNet50 or VGG-16 block for the parts of the models that are derived from those architectures and additionally after every ReLU following a fully connected or convolutional layer that is not contained in a ResNet or VGG block. For 'Retrain' we fine-tune the whole network and RGEM and Ridge are restricted to the last layer.

4.6. The effect of the sample size

As the number of instances available for refinement is limited, a natural question is what impact the number of samples has on the efficacy of refinement and if refining with too few instances can be detrimental. In this section, we repeat the experiment from Section 4.4 for datasets containing 25, 50, 200, 500, and 700 instances per class for refinement, under 0% and 100% uniform poisoning. Slack is again set to 5%. If for some classes fewer correctly classified instances are available for refinement, these are over-sampled to achieve the desired number.

The effect of varying sample size is shown in Fig. 5. It can be seen that especially in the low-sample regime, the positive effect of refinement is modulated by the number of instances. See Supplementary Note H for all other methods. While refinement with a small sample size appears in most cases to be remarkably effective for increasing 100%-poisoned accuracy, clean-data accuracy tends to suffer as the sample does not cover all of the features necessary to generalize, some of which are thus pruned away. For this reason, a larger refinement sample is in most cases beneficial, in particular for preserving clean-data accuracy. Yet, there are two cases that stand out as breaking this rule: The modified MNIST dataset and the 'carton/envelope' task. In both cases, the gap between 0% and 100%-poisoned accuracy is close to constant, suggesting that the drop in accuracy stems from a loss of generalizing features rather than a loss of robustness, as could be induced e.g. by samples contaminated by CH features. Considering the effect of slack, displayed in Fig. 4, we can see that those two scenarios are also cases for which 5% slack leads to stronger than optimal refinement. We hypothesize that here, the negative effect of increasing sample size stems from the interrelation between sample size and refinement strength. In particular, for EGEM and PCA-EGEM, using fewer instances generally means less coverage of the feature space, which leads to fewer active neurons and hence more zero coefficients in the neuron-wise pruning factor described in Eq. (6). By this mechanism, larger sample sizes potentially lead to *weaker* refinement. This effect may be exacerbated if a larger refinement sample also introduces images containing features that share parts of the latent representation with the CH feature.

Since clean-data accuracy can be evaluated on held-out data, we can observe that applying PCA-EGEM results in fairly predictable 100%-poisoning performance across a wide range of sample sizes, i.e. the spread between clean-data and poisoned-data accuracy is small, as is demonstrated by the relatively small shaded area in Fig. 5.

A comparison to all other baselines can be found in the Supplementary Notes. What becomes particularly apparent is that Retrain benefits the most from large sample sizes and is among the weakest methods in the small-sample regime. On the modified MNIST dataset, its 100%-poisoned accuracy only reaches 90% with a refinement sample size of more than 200, whereas all other methods already achieve this with only 25 images.

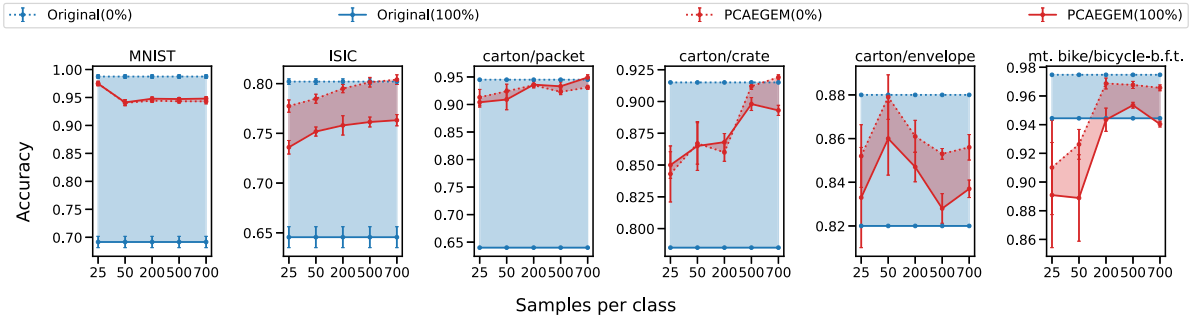

Fig. 5. Effect of the number of instances used for refinement. The dotted line indicates clean-data accuracy whereas the solid line indicates 100%-poisoned data accuracy. Mean and standard deviation are computed over 5 runs.

5. MNIST revisited: Varying the Clever-Hans feature

In this section, we are revisiting the MNIST task from Section 4, using different CH features. While all previous tasks contained localized additive artifacts, we will here investigate the effect of blur, removal of the lower part of the digits, and color shift as CH feature. We run the experiment with 5% slack and 50 samples per class for refinement for each of these CH features.

Fig. 6 shows the classification accuracy under 0% and 100% poisoning and permits some interesting observations.

Spatially localized artifacts have a stronger effect. The original model sees the strongest decline in accuracy due to spatially localized CH features, specifically the additive artifact and removal. It appears that these features can be modeled well by the given architecture, e.g. the artifact can be captured easily by a convolutional filter or by the spatially disentangled features that the convolutional layers pass on to the fully-connected layers. This observation demonstrates that not all CH features are equal in their effect on a particular architecture, even if their rate of occurrence is equal.

EGEM underperforms on non-additive artifacts. The localized additive artifact is most amenable to removal via pruning whereas accuracy in all other scenarios drops when applying EGEM. We hypothesize that the pixel artifact has a very localized representation in the network (e.g. few particular convolutional filters, or neurons) and thus the neural basis, which EGEM uses for pruning, and the CH feature are more aligned.

In order to substantiate this hypothesis, we measure the sparsity of the change in representation induced by the CH feature. To this end, we sample 100 images of the digit 8 and measure the activations a^l at every layer l that is to be refined. We quantify the sparsity of the difference of activations before and after adding the CH artifact to the input by the ratio of the ℓ_2 and the ℓ_1 norm:

$$\frac{\|\mathbf{a}^l_{CH} - \mathbf{a}^l\|_2}{\|\mathbf{a}^l_{CH} - \mathbf{a}^l\|_1}, \tag{9}$$

which is 1 if the difference is localized at one dimension and less otherwise. The average sparsity of the effect of the CH feature is shown in Fig. 7. The representation change induced by the spatially localized CH feature has the highest sparsity, which may explain why EGEM is effective in this scenario but not in others.

The representation realignment performed by PCA-EGEM via the PCA virtual layer resolves the problem, leading to consistently high accuracy across all types of CH features. This higher flexibility of PCA-EGEM w.r.t. the nature of the CH feature further justifies its use over basic EGEM.

6. Use case on CelebA: Reducing bias

In this section, we will take a closer look at the effect of applying PCA-EGEM to a model trained on the CelebA dataset [76]. In contrast to the previous experiments, we do not evaluate based on a specific known CH feature, but rather conduct the analysis in an exploratory manner, uncovering subpopulations for which a learned CH behavior is leading to biased classifications. In practice, such an analysis could be done in hindsight, e.g. by proceeding along the following steps: (1) a third-party model is acquired, (2) it is refined using PCA-EGEM and available data, (3) it is deployed, i.e. applied to test data, (4) the effect of PCA-EGEM is examined on the test data using XAI and recall statistics.

The CelebA dataset contains 202,599 portrait images of celebrities, each associated with 40 binary attributes. The existence of spurious correlations in the CelebA dataset has been documented previously [23,60,94,95] and it can be seen in Supplementary Note C.2 that the attributes in the training set are correlated to various degrees. We train a convolutional neural network (details in Supplementary Note E) on the 'train' split of the CelebA dataset using cross-entropy loss on a 'Blond_Hair'-vs-not classification task. The training data is stratified and we achieve a binary test accuracy of 93%, which is comparable to accuracy reported in other works, e.g. Sagawa et al. [23]. We regard this classifier as a model given to the user by a third party.

In the following, we will assume a scenario where the user seeks to use the third-party classifier to retrieve blond people from a set of images available during deployment. They wish this retrieval process to be accurate and not biased against subgroups in the population. In order to analyze the impact of applying PCA-EGEM on such retrieval task, we simulate a validation set where the user has a limited subset of 'clean' examples, specifically, 200 examples of both classes, that are correctly predicted by the model and whose explanation highlight the actual blond hair as determined by LRP scores falling dominantly within the area of the image where the hair is located (see Supplementary Note F). These explanations (considered by the user to be all valid) are then fed to PCA-EGEM to produce a model that is more robust to potential unobserved Clever Hans effects. As for the previous experiments, we use 5% slack, which translates here to $\alpha = 0.01$.

6.1. PCA-EGEM reduces exposure to shirt collars

After the model is deployed, the analysis of the decision strategy (of the original and refined model) can be reexamined in light of the new data now available. Fig. 8 shows explanations for some retrieved images, specifically, evidence for them being predicted to be blond. We can observe that pixels displaying hair are considered to be relevant and remain so after refinement.

However, one can also identify a significant change of strategy before and after refinement in the lower part of the image: The original model appears to make heavy use of shirt and suit collars as a feature inhibiting the detection of blond hair, whereas such inhibiting effect is much milder in the refined model. This observation suggests that PCA-EGEM has effectively mitigated a previously unobserved Clever Hans strategy present in the original model, and as a result, effectively aided the retrieval of images with collars on them.

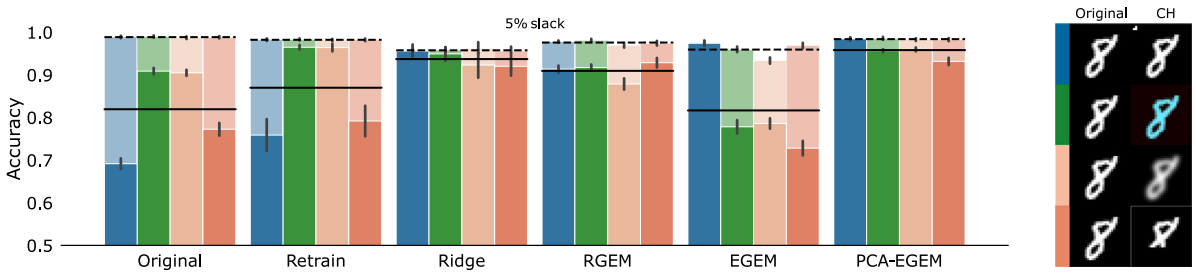

Fig. 6. The accuracy for 0% (lighter shade) and uniform 100% (darker shade) poisoning with the spurious feature. Solid lines show average 100%-poisoned accuracy and dashed lines show average clean-data accuracy over all datasets. The results shown are the mean accuracy and standard deviation obtained using 50 refinement samples per class on the respective test sets over 10 runs.

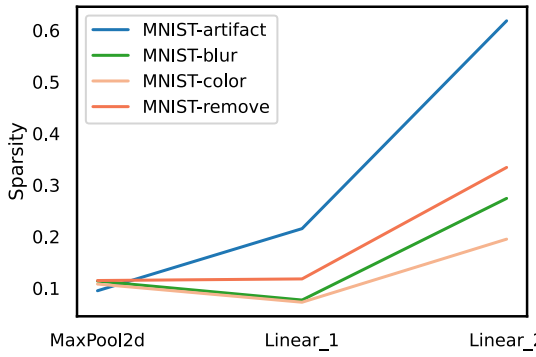

Fig. 7. Sparsity of the change in representations of MNIST images of the class 8, induced by different CH artifacts and computed with Eq. (9). Results are averaged over 100 images.

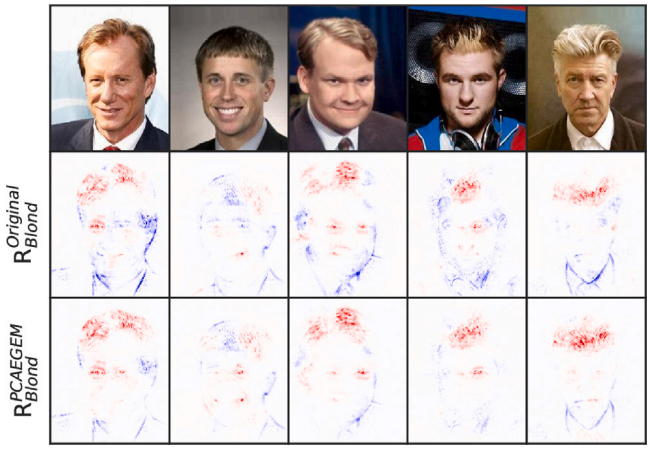

Fig. 8. Test set images that exhibit strong changes in the detection of blond hair and corresponding LRP explanations, before and after refinement. Red indicates positive and blue negative contribution to the detection of blond hair. Shirt collars and similar features appear to inhibit the prediction of blond hair in the original model but less so in the refined one.

6.2. PCA-EGEM balances recall across subgroups

We will now analyze the implication of the Clever Hans effect reduction by PCA-EGEM on specific subgroups, specifically, whether certain subgroups benefit from the model refinement in terms of recalling members with the attribute 'Blond_Hair'.

To this end, we randomly sample for every attribute in the dataset, a subset of 5000 images from the test data that only contains samples exhibiting this attribute. If fewer images are available for some attribute, we use all of them. We evaluate the classifier before and after the application of PCA-EGEM on each of these subgroups.

Fig. 9 shows recall scores for each subgroup. We observe a substantial increase of recall on low-recall subgroups, such as 'Wearing_Necktie', 'Goatee', and 'Male'. Most high-recall groups see only minuscule negative effects. Overall, while having almost no effect on the dataset-wide recall, we can observe that the application of PCA-EGEM rebalances recall in favor of under-recalled subsets. Our investigation thus demonstrates that a model bias responsible for under-detecting blond hair in these subgroups has been mitigated by applying PCA-EGEM. This consequently leads to a set of retrieved images that is more representative of the different subgroups and more diverse.

It is of theoretical interest to ask whether such rebalancing effect would generalize to other scenarios. An argument is that the underrepresentation of certain subgroups in the retrieved set is mainly caused by subgroups with low prevalence of the class of interest and those subgroups being actively suppressed by the model in order to optimize its accuracy. In practice, such suppression can be achieved by identifying features specific to the subgroup and, although causally unrelated to the task, making these features contribute negatively to the output score. Our PCA-EGEM technique, by removing such task-irrelevant Clever Hans features, remodels the decision function in favor of these low-prevalence subgroups, thereby leading to a more balanced set of retrieved instances.

Two outliers to the overall rebalancing effect can however be noted in Fig. 9: 'Wearing_Hat', and 'Blurry'. Interestingly, these are two subgroups in which the feature of interest (the hair) is occluded or made less visible. In other words, in these two subgroups, only weakly correlated features are available for detection, and their removal by PCA-EGEM consequently reduces the recall. An underlying assumption behind the rebalancing effect is therefore that the concept of interest (blond hair) is detectable in the input image without resorting to weakly or spuriously correlated features which may be refined away due to being underrepresented in the available data. A more detailed discussion can be found in Supplementary Note I.

Overall, we have demonstrated in our CelebA use case, that PCA-EGEM can be useful beyond raising accuracy on disadvantageous test-set distributions. Specifically, we have shown that our PCA-EGEM approach enables the retrieval of a more diverse set of positive instances from a large heterogeneous dataset.

7. Use case: Removing a CH effect in movie reviews

In this section, we demonstrate the use of PCA-EGEM for unlearning CH features in the context of binary sentiment classification of movie reviews, employing a transformer model.

Existing works have underscored the susceptibility of pretrained sentiment classifiers to features spuriously correlated with the class, such as actor names [46,96] or the occurrence of stop words [97], even if these words should not be taken into account for the classification process.

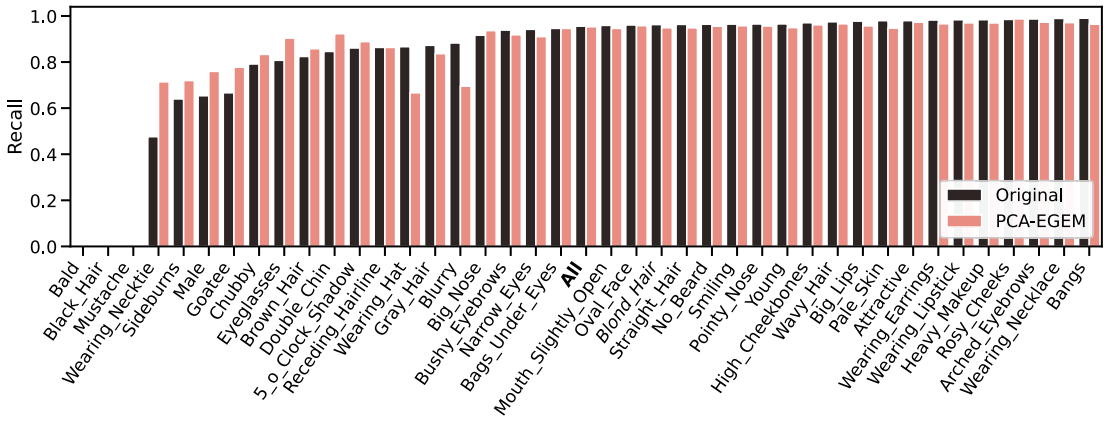

Fig. 9. Comparison of recall of the pretrained model before (Original) and after refinement (PCA-EGEM). Recall is calculated on subsets of CelebA containing only samples exhibiting the attribute on the x-axis. 'All' is sampled from the whole test set.

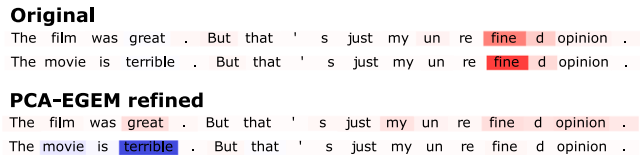

Fig. 10. Effect of each word, estimated by replacing them by the UNK token. Red: positive, blue: negative sentiment. Normalized on the sentence level.

To demonstrate the potential of PCA-EGEM to reduce such biases, we fine-tune a pretrained DistilBert [98] model[3] on the training set of the IMDB dataset [99]. A CH effect is artificially induced by appending the sentence "*But that's just my unrefined opinion.*" to 25% of positive reviews within the training set. This sentence was chosen as it is plausible in the context of movie reviews and may be a formulation linked to specific reviewers rather than sentiment, thereby potentially giving rise to a CH effect. For refinement with PCA-EGEM we use 1000 positively and 1000 negatively labeled reviews, randomly selected from the validation set. Employing a slack of 5%, we refine the linear output layer of each multi-head self-attention block, using the same hyper-parameter search procedure described in Section 4.5.

The unrefined fine-tuned classifier achieves a test accuracy of 92.4% whereas the refined version achieves 90.1%. Upon appending the CH sentence to all test-samples (100% uniform poisoning), the unrefined model only achieves the base-rate accuracy of 50% whereas the refined model maintains high accuracy of 88.8%. Evidently, the refinement procedure was able to mitigate sensitivity to the introduced CH feature. We create explanations for two synthetic, poisoned review examples in Fig. 10 by measuring the effect of replacing each token in the sentence with the UNK token. The original model focuses primarily on the appended sentence, specifically, the token "fine" which DistilBert seems to encode saliently for its relevance to sentiment classification.[4] On the other hand, this effect is starkly reduced in the refined model. This experiment demonstrates that the proposed PCA-EGEM refinement approach can also be applied insightfully to other neural network architectures, such as transformer models, and to data modalities beyond images, such as text.

[3] https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english

[4] When repeating the same experiment without the token "fine" (i.e. by appending the shorter statement "*But that's just my opinion.*"), the CH effect does not develop.

8. Open questions

We could demonstrate the efficacy of our proposed EGEM and PCA-EGEM refinement methods for mitigation of Clever Hans effects in Sections 4–7, however, it could also be observed that (1) a complete removal of the model's response to the spurious (CH) feature is usually not fully achieved, and (2) classification accuracy on clean data may suffer slightly. We suggest that there are multiple reasons for these undesired effects.

8.1. Entangled feature representations

In deep neural networks, CH features are not generally neatly disentangled from generalizing features. This means that either entangled well-generalizing features might suffer from pruning, reducing clean-data accuracy, or CH features might not be pruned due to being entangled with a feature present in the clean dataset. The latter would inhibit robustification against the CH feature. In this case, a trade-off between clean-data accuracy and robustness exists, which can be navigated via the slack parameter introduced in Section 4.5.

A way to avoid this conflict altogether could be a feature disentanglement strategy, separating the CH feature from others. While the PCA-EGEM extension achieves a basic form of disentanglement, more refined disentanglement methods based, for example, on finding independent components, could be considered in future work. Chormai et al. [75] have demonstrated the efficacy of explanation-based disentanglement into independent subspaces. In principle, our proposed soft-pruning strategy could be applied in such spaces. An alternative direction may be the disentanglement into factors derived from human behavioral data [100], which have been shown to be represented in different pretrained neural networks to various degrees [101].

8.2. Limited available data

Another open question is how the proposed exposure minimization approach can be effective in the case where the available data is scarce or unrepresentative of the true input distribution.

The disentanglement methods discussed above, beyond achieving the desired separation between the CH and correct strategies, may also be useful for addressing data scarcity. In particular, they could enable the generation of low-dimensional subspaces in which only a few data points are sufficient to characterize the correct decision strategy.

Furthermore, methods that extend the data in an informed manner could also be useful. This includes methods for synthesizing class prototypes (via activation-maximization [102,103], or class-conditioned sampling [104]). The generated prototypes are sometimes referred to as '*global explanation*'. These generated instances may be appended to the available data and subsequently included in the dataset available for refinement.

8.3. Imprecise explanations or user assessment

Beyond the need for sufficient data to validate the model, it is also crucial that the XAI method upon which our exposure minimization technique builds is able to accurately convey the presence or absence of CH effects. While there have been concerns about the effectiveness of XAI methods in some scenarios [38,105–107], especially their ability to reliably detect CH effects, these concerns could be addressed in future work by moving beyond pixel-wise explanations e.g. using concept-based or counterfactual explanations [108–110].

Furthermore, assuming a well-working XAI technique, the user should be able to make a decision (e.g. no CH effects) in an error-free manner. Lack of attention, limited knowledge on how to interpret explanations, or an inability to process the potentially large amount of produced explanations in reasonable time may cause the user to falsely conclude that the ML model at hand is CH-free. One overall direction to address these issues is to develop improved interaction tools between the XAI technique and the user. For example, the set of explanations could be rendered to the user in a more intuitive way so that more examples can be inspected and outliers can be paid special attention – a strategy already implemented e.g. in SpRAy [19]. Beyond that, the amount of explanations to be inspected by the user could be decreased by extracting a representative subset of instances from the available data. This may be done by summarizing techniques such as coresets [111] or pool-based sampling approaches as are used in active learning [112] and may also be based on explanations [113].

8.4. Performance after removing CH features

As pointed out in previous work [95], removing spuriously correlated (CH) features can hurt performance on data where this correlation holds. One such effect was observed in our experiment on CelebA in Section 6, where a refined model would underperform on the 'Blurry' and 'Wearing_Hat' subgroups at test time due to not being able to rely on the weakly correlated CH features. If the CH features were present in the data available for refinement, a proper reweighting of features (inhibiting CH features and enhancing correct ones) would in principle be achievable before deployment (e.g. [37,63]). In the more difficult scenario studied in this paper, this is not possible. Still, one could potentially monitor the discrepancy between the original model and PCA-EGEM at test time, in order to quickly mitigate unresolved biases in the deployed model.

9. Conclusion

Sensitivity to distribution shifts, such as the ones induced by spurious correlations (so-called Clever Hans effects) has long been an Achilles heel of machine learning approaches, such as deep learning. The problem becomes even more pronounced with the increasing adoption of foundation models for which the training data may not be public and is thus closed to scrutiny. Explanation techniques have the potential to uncover such deficiencies by putting a human in the loop [19,30,114,115]. Previous work in XAI has mainly focused on improving explanations or fixing flaws in a model that have been identified by the user from such explanations. In contrast, we have considered the under-explored case where the human and the explanation *agree* but where there are possibly unobserved spurious features that the model is sensitive to. While recent work has shown that XAI-based validation techniques may fail to detect some of these Clever Hans strategies employed by a model [38], we have argued that one can nevertheless still reduce the exposure of a model to some of these hidden strategies and demonstrated this via our proposed Explanation-Guided Exposure Minimization approach.

Our approach, while formulated as an optimization problem, reduces to simple pruning rules applied in intermediate layers, thereby making our method easily applicable, without retraining, to complex deep neural network models such as those used in computer vision. Our method was capable of systematically improving prediction performance on a variety of complex classification problems, outperforming existing and contributed baselines.

Concluding this paper, we would like to emphasize the novelty of our approach, which constitutes an early attempt to leverage correct explanations for producing refined ML models and attempts to tackle the realistic scenario where Clever Hans features are not accessible. We believe that in future work, the utility derived from explanations via refinement can still be expanded, e.g. by letting the user specify what is correct and what is incorrect in an explanation so that the two components can be treated separately, or by identifying sets of examples to present to the user that are the most useful to achieve model refinement, for example, by ensuring that they cover the feature space adequately or by active learning schemes.

CRediT authorship contribution statement

Lorenz Linhardt: Methodology, Software, Investigation, Writing – original draft, Visualization. **Klaus-Robert Müller:** Conceptualization, Writing – review & editing, Resources, Funding acquisition. **Grégoire Montavon:** Conceptualization, Methodology, Writing – original draft, Visualization, Supervision, Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partly funded by the German Ministry for Education and Research (under refs 01IS14013A-E, 01GQ1115, 01GQ0850, 01IS18056A, 01IS18025A and 01IS18037A), the German Research Foundation (DFG) as Math+: Berlin Mathematics Research Center (EXC 2046/1, project-ID: 390685689). Furthermore KRM was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program, Korea University and No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation). We thank Pattarawat Chormai and Christopher Anders for the valuable discussion and the extracted watermark artifact.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.inffus.2023.102094.

References

[1] R. Bommasani, D.A. Hudson, E. Adeli, R. Altman, S. Arora, et al., On the opportunities and risks of foundation models, 2021, arXiv preprint arXiv:2108.07258.

[2] J. Devlin, M.-W. Chang, K. Lee, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in: Advances in Neural Information Processing Systems. Vol. 33, 2020, pp. 1877–1901.

[4] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al., Dinov2: Learning robust visual features without supervision, 2023, arXiv preprint arXiv:2304.07193.

[5] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[6] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Proceedings of the 3rd International Conference on Learning Representations, 2015.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.

[8] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 5998–6008.

[10] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: Proceedings of the 37th International Conference on Machine Learning. Vol. 119, 2020, pp. 1597–1607.

[11] S. Aradi, Survey of deep reinforcement learning for motion planning of autonomous vehicles, IEEE Trans. Intell. Transp. Syst. 23 (2) (2022) 740–759.

[12] D. Capper, D.T.W. Jones, M. Sill, V. Hovestadt, D. Schrimpf, et al., DNA methylation-based classification of central nervous system tumours, Nature 555 (7697) (2018) 469–474.

[13] L. Wynants, B. Van Calster, G.S. Collins, R.D. Riley, G. Heinze, et al., Prediction models for diagnosis and prognosis of Covid-19: Systematic review and critical appraisal, BMJ 369 (8242) (2020) m1328.

[14] M. Roberts, D. Driggs, M. Thorpe, J. Gilbey, M. Yeung, et al., Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans, Nat. Mach. Intell. 3 (3) (2021) 199–217.

[15] P. Jurmeister, S. Glöß, R. Roller, M. Leitheiser, S. Schmid, L.H. Mochmann, et al., DNA methylation-based classification of sinonasal tumors, Nature Commun. 13 (1) (2022) 7148.

[16] E. Sorantin, M.G. Grasser, A. Hemmelmayr, S. Tschauner, F. Hrzic, V. Weiss, J. Lacekova, A. Holzinger, The augmented radiologist: Artificial intelligence in the practice of radiology, Pediatr. Radiol. 52 (11) (2022) 2074–2086.

[17] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[18] T. Gebru, J. Buolamwini, Gender shades: Intersectional accuracy disparities in commercial gender classification, in: Proceedings of the 1st Conference on Fairness, Accountability and Transparency. Vol. 81, 2018, pp. 77–91.

[19] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking clever hans predictors and assessing what machines really learn, Nature Commun. 10 (1) (2019) 1096.

[20] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (5) (2019) 206–215.

[21] R. Geirhos, J.H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F.A. Wichmann, Shortcut learning in deep neural networks, Nat. Mach. Intell. 2 (11) (2020) 665–673.

[22] C.S. Calude, G. Longo, The deluge of spurious correlations in big data, Found. Sci. 22 (3) (2016) 595–612.

[23] S. Sagawa, P.W. Koh, T.B. Hashimoto, P. Liang, Distributionally robust neural networks, in: Proceedings of the International Conference on Learning Representations, 2020.

[24] J.K. Winkler, C. Fink, F. Toberer, A. Enk, T. Deinlein, R. Hofmann-Wellenhof, L. Thomas, A. Lallas, A. Blum, W. Stolz, H.A. Haenssle, Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition, JAMA Dermatol. 155 (10) (2019) 1135.

[25] 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, Sci. Rep. 10 (2020) 6423.

[26] W. Wu, H. Xu, S. Zhong, M.R. Lyu, I. King, Deep validation: Toward detecting real-world corner cases for deep neural networks, in: Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2019, pp. 125–137.

[27] S. Booth, Y. Zhou, A. Shah, J. Shah, Bayes-TrEx: A Bayesian sampling approach to model transparency by example, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2020.

[28] Y. Tian, Z. Zhong, V. Ordonez, G. Kaiser, B. Ray, Testing DNN image classifiers for confusion & bias errors, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, 2020, pp. 1122–1134.

[29] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, G.-Z. Yang, XAI-explainable artificial intelligence, Science Robotics 4 (37) (2019) eaay7120.

[30] W. Samek, G. Montavon, A. Vedaldi, L.K. Hansen, K.-R. Müller (Eds.), Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, in: Lecture Notes in Computer Science, vol. 11700, Springer International Publishing, Cham, 2019.

[31] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[32] W. Samek, G. Montavon, S. Lapuschkin, C.J. Anders, K.-R. Müller, Explaining deep neural networks and beyond: A review of methods and applications, Proc. IEEE 109 (3) (2021) 247–278.

[33] A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, W. Samek (Eds.), xxAI - Beyond Explainable AI - International Workshop, Held in Conjunction with ICML 2020, in: Lecture Notes in Computer Science, vol.13200, Springer, 2022.

[34] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?" explaining the predictions of any classifier, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[35] G. Plumb, M.T. Ribeiro, A. Talwalkar, Finding and fixing spurious patterns with explanations, 2021, arXiv preprint, arXiv:2106.02112.

[36] A. Ghandeharioun, B. Kim, C.L. Li, B. Jou, B. Eoff, R.W. Picard, Dissect: Disentangled simultaneous explanations via concept traversals, in: Proceedings of the 10th International Conference on Learning Representations, 2022.

[37] C.J. Anders, L. Weber, D. Neumann, W. Samek, K.-R. Müller, S. Lapuschkin, Finding and removing Clever Hans: Using explanation methods to debug and improve deep models, Inf. Fusion 77 (2022) 261–295.

[38] J. Adebayo, M. Muelly, H. Abelson, B. Kim, Post hoc explanations may be ineffective for detecting unknown spurious correlation, in: Proceedings of the 10th International Conference on Learning Representations, 2022.

[39] 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 (7) (2015) e0130140.

[40] D. Gunning, D.W. Aha, DARPA's explainable artificial intelligence (XAI) program, AI Mag. 40 (2) (2019) 44–58.

[41] B. Mittelstadt, C. Russell, S. Wachter, Explaining explanations in AI, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, ACM, 2019, pp. 279–288.

[42] Y. Zhang, P. Tino, A. Leonardis, K. Tang, A survey on neural network interpretability, IEEE Trans. Emerg. Top. Comput. Intell. 5 (5) (2021) 726–742.

[43] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, Int. J. Comput. Vis. 128 (2) (2020) 336–359.

[44] Q. Zhang, X. Wang, R. Cao, Y.N. Wu, F. Shi, S.-C. Zhu, Extraction of an explanatory graph to interpret a CNN, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (2021) 3863–3877.

[45] G. Jeon, H. Jeong, J. Choi, Distilled gradient aggregation: Purify features for input attribution in the deep neural network, in: Advances in Neural Information Processing Systems. Vol. 35, Curran Associates, Inc., 2022, pp. 26478–26491.

[46] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K.T. Schütt, K.-R. Müller, G. Montavon, Higher-order explanations of graph neural networks via relevant walks, IEEE Trans. Pattern Anal. Mach. Intell. 44 (11) (2022) 7581–7596.

[47] S. Letzgus, P. Wagner, J. Lederer, W. Samek, K.-R. Müller, G. Montavon, Toward explainable artificial intelligence for regression models: A methodological perspective, IEEE Signal Process. Mag. 39 (4) (2022) 40–58.

[48] O. Eberle, J. Büttner, F. Kräutli, K. Müller, M. Valleriani, G. Montavon, Building and interpreting deep similarity models, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2022) 1149–1161.

[49] P. Liznerski, L. Ruff, R.A. Vandermeulen, B.J. Franks, M. Kloft, K.-R. Müller, Explainable deep one-class classification, 2020, arXiv preprint, arXiv:2007.01760.

[50] J. Kauffmann, M. Esders, L. Ruff, G. Montavon, W. Samek, K.-R. Müller, From clustering to cluster explanations via neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–15.

[51] S. Wu, M. Yuksekgonul, L. Zhang, J. Zou, Discover and cure: Concept-aware mitigation of spurious correlation, in: Proceedings of the 40th International Conference on Machine Learning, 2023.

[52] A.S. Ross, M.C. Hughes, F. Doshi-Velez, Right for the right reasons: Training differentiable models by constraining their explanations, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 2662–2670.

[53] L. Rieger, C. Singh, W.J. Murdoch, B. Yu, Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge, in: Proceedings of the 37th International Conference on Machine Learning, 2019, pp. 8116–8126.

[54] B. Simpson, F. Dutil, Y. Bengio, J.P. Cohen, GradMask: Reduce overfitting by regularizing saliency, 2019, arXiv preprint, arXiv:1904.07478.

[55] S. Teso, K. Kersting, Explanatory interactive machine learning, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 239–245.

[56] 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, Nat. Mach. Intell. 2 (8) (2020) 476–486.

[57] K. Bykov, M. Deb, D. Grinwald, K.-R. Müller, M.M.C. Höhne, DORA: Exploring outlier representations in deep neural networks, 2022, arXiv preprint, arXiv:2206.04530.

[58] E. Creager, J.-H. Jacobsen, R. Zemel, Environment inference for invariant learning, in: Proceedings of the 38th International Conference on Machine Learning. Vol. 139, 2021, pp. 2189–2200.

[59] W. Hu, G. Niu, I. Sato, M. Sugiyama, Does distributionally robust supervised learning give robust classifiers? in: Proceedings of the 35th International Conference on Machine Learning. Vol. 80, 2018, pp. 2029–2037.

[60] S. Sagawa, A. Raghunathan, P.W. Koh, P. Liang, An investigation of why overparameterization exacerbates spurious correlations, in: Proceedings of the 37th International Conference on Machine Learning. Vol. 119, 2020, pp. 8346–8356.

[61] N.S. Sohoni, M. Sanjabi, N. Ballas, A. Grover, S. Nie, H. Firooz, C. Ré, BARACK: Partially supervised group robustness with guarantees, 2021, arXiv preprint, arXiv:2201.00072.

[62] B.Y. Idrissi, M. Arjovsky, M. Pezeshki, D. Lopez-Paz, Simple data balancing achieves competitive worst-group-accuracy, in: Proceedings of the First Conference on Causal Learning and Reasoning, 2022, pp. 177:336–351.

[63] P. Kirichenko, P. Izmailov, A.G. Wilson, Last layer re-training is sufficient for robustness to spurious correlations, in: Proceedings of the 11th International Conference on Learning Representations, 2023.

[64] J. Nam, J. Kim, J. Lee, J. Shin, Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation, 2022, arXiv preprint, arXiv:2204.02070.

[65] D.J. Fremont, X. Yue, T. Dreossi, A.L. Sangiovanni-Vincentelli, S. Ghosh, S.A. Seshia, Scenic: A language for scenario specification and scene generation, in: Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI, Association for Computing Machinery, 2019, pp. 63–78.

[66] E.Z. Liu, B. Haghgoo, A.S. Chen, A. Raghunathan, P.W. Koh, S. Sagawa, P. Liang, C. Finn, Just train twice: Improving group robustness without training group information, in: Proceedings of the 38th International Conference on Machine Learning. Vol. 139, 2021, pp. 6781–6792.

[67] 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, 2021, arXiv preprint arXiv:2106.13200.

[68] 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 International Publishing, 2019, pp. 193–209.

[69] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: Proceedings of the 34th International Conference on Machine Learning. Vol. 70, 2017, pp. 3145–3153.

[70] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning. Vol. 70, 2017, pp. 3319–3328.

[71] S.-K. Yeom, P. Seegerer, S. Lapuschkin, A. Binder, S. Wiedemann, K.-R. Müller, W. Samek, Pruning by explaining: A novel criterion for deep neural network pruning, Pattern Recognit. 115 (2021) 107899.

[72] E. Härkönen, A. Hertzmann, J. Lehtinen, S. Paris, GANSpace: Discovering interpretable GAN controls, in: Advances in Neural Information Processing Systems. Vol. 33, 2020, pp. 9841–9850.

[73] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems. Vol. 27, 2014, pp. 2672–2680.

[74] Y. Shen, B. Zhou, Closed-form factorization of latent semantics in GANs, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 1532–1540.

[75] P. Chormai, J. Herrmann, K.-R. Müller, G. Montavon, Disentangled explanations of neural network predictions by finding relevant subspaces, 2022, arXiv preprint, arXiv:2212.14855.

[76] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision, ICCV, 2015.

[77] Y. LeCun, C. Cortes, MNIST handwritten digit database, 1998, http://yann.lecun.com/exdb/mnist/.

[78] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[79] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. (IJCV) 115 (3) (2015) 211–252.

[80] N.C.F. Codella, D. Gutman, M.E. Celebi, B. Helba, M.A. Marchetti, S.W. Dusza, A. Kalloo, K. Liopyris, N. 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), 2018, arXiv preprint, arXiv:1710.05006.

[81] P. Tschandl, C. Rosendahl, H. Kittler, The HAM10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions, Sci. Data 5 (2018) 18016.

[82] M. Combalia, N.C.F. Codella, V. Rotemberg, B. Helba, V. Vilaplana, O. Reiter, C. Carrera, A. Barreiro, A.C. Halpern, S. Puig, J. Malvehy, BCN20000: Dermoscopic lesions in the wild, 2019, arXiv preprint, arXiv:1908.02288.

[83] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, 2014, arXiv preprint, arXiv:1312.6114.

[84] N.K. Mishra, M.E. Celebi, An overview of melanoma detection in dermoscopy images using image processing and machine learning, 2016, arXiv preprint, arXiv:1601.07843.

[85] A. Bissoto, E. Valle, S. Avila, Debiasing skin lesion datasets and models? Not so fast, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020.

[86] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 770–778.

[87] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the 3rd International Conference on Learning Representations, 2015.

[88] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems 32, 2019, pp. 8024–8035.

[89] S. Santurkar, D. Tsipras, A. Madry, BREEDS: Benchmarks for subpopulation shift, in: Proceedings of the 9th International Conference on Learning Representations, 2021.

[90] A.H. Ashouri, T.S. Abdelrahman, A. Dos Remedios, Retraining-free methods for fast on-the-fly pruning of convolutional neural networks, Neurocomputing 370 (C) (2019) 56–69.

[91] A. Polyak, L. Wolf, Channel-level acceleration of deep face representations, IEEE Access 3 (2015) 2163–2175.

[92] Y. Lee, A.S. Chen, F. Tajwar, A. Kumar, H. Yao, P. Liang, C. Finn, Surgical fine-tuning improves adaptation to distribution shifts, in: Proceedings of the 10th International Conference on Learning Representations, 2023.

[93] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, R. Sayres, Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV), in: Proceedings of the 35th International Conference on Machine Learning. Vol. 80, 2018, pp. 2668–2677.

[94] T. Xu, J. White, S. Kalkan, H. Gunes, Investigating bias and fairness in facial expression recognition, in: Computer Vision – ECCV 2020 Workshops, 2020, pp. 506–523.

[95] F. Khani, P. Liang, Removing spurious features can hurt accuracy and affect groups disproportionately, in: Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, ACM, 2021, pp. 196–205.

[96] A. Ali, T. Schnake, O. Eberle, G. Montavon, K.-R. Müller, L. Wolf, XAI for transformers: Better explanations through conservative propagation, in: Proceedings of the 39th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol.162, PMLR, 2022, pp. 435–451.

[97] A. Liusie, V. Raina, V. Raina, M. Gales, Analyzing biases to spurious correlations in text classification tasks, in: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Association for Computational Linguistics, 2022, pp. 78–84.

[98] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of BERT: Smaller, faster, cheaper and lighter, 2019, arXiv e-prints, arXiv:1910.01108.

[99] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, Association for Computational Linguistics, 2011, pp. 142–150.

[100] L. Muttenthaler, C.Y. Zheng, P. McClure, R.A. Vandermeulen, M.N. Hebart, F. Pereira, VICE: Variational interpretable concept embeddings, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems. Vol. 35, Curran Associates, Inc., 2022, pp. 33661–33675.

[101] L. Muttenthaler, J. Dippel, L. Linhardt, V. Robert A, S. Kornblith, Human alignment of neural network representations, in: Proceedings of the 11th International Conference on Learning Representations, 2023.

[102] K. Hansen, D. Baehrens, T. Schroeter, M. Rupp, K.-R. Müller, Visual interpretation of kernel-based prediction models, Mol. Inform. 30 (9) (2011) 817–826.

[103] A. Nguyen, J. Yosinski, J. Clune, Understanding neural networks via feature visualization: A survey, in: Lecture Notes in Computer Science, vol. 11700 LNCS, Springer Verlag, 2019, pp. 55–76.

[104] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, J. Clune, Plug & play generative networks: Conditional iterative generation of images in latent space, 2016, arXiv e-print, arXiv:1612.00005.

[105] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim, Sanity checks for saliency maps, in: Advances in Neural Information Processing Systems, 2018, pp. 9505–9515.

[106] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, P. Kessel, Explanations can be manipulated and geometry is to blame, in: Advances in Neural Information Processing Systems. Vol. 32, 2019, pp. 13589–13600.

[107] A. Binder, L. Weber, S. Lapuschkin, G. Montavon, K.-R. Müller, W. Samek, Shortcomings of top-down randomization-based sanity checks for evaluations of deep neural network explanations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023, pp. 16143–16152.

[108] I. Stepin, J.M. Alonso, A. Catala, M. Pereira-Farina, A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence, IEEE Access 9 (2021) 11974–12001.

[109] S. Verma, J. Dickerson, K. Hines, Counterfactual explanations for machine learning: Challenges revisited, 2021, arXiv preprint, arXiv:2106.07756.

[110] A.-K. Dombrowski, J.E. Gerken, K.-R. Müller, P. Kessel, Diffeomorphic counterfactuals with generative models, 2022, arXiv preprint, arXiv:2206.05075.

[111] J.M. Phillips, Coresets and sketches, Handbook of Discrete and Computational Geometry, Chapman and Hall/CRC, 2017, pp. 1269–1288.

[112] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B.B. Gupta, X. Chen, X. Wang, A survey of deep active learning, ACM Comput. Surv. 54 (9) (2021).

[113] D. Mahapatra, A. Poellinger, L. Shao, M. Reyes, Interpretability-driven sample selection using self supervised learning for disease classification and segmentation, IEEE Trans. Med. Imaging 40 (10) (2021) 2548–2562.

[114] Z.C. Lipton, The Mythos of model interpretability, Commun. ACM 61 (10) (2016) 35–43.

[115] W.J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, B. Yu, Definitions, methods, and applications in interpretable machine learning, Proc. Natl. Acad. Sci. USA 116 (44) (2019) 22071–22080.