nature machine intelligence

Article

https://doi.org/10.1038/s42256-025-01000-2

Explainable AI reveals Clever Hans effects in unsupervised learning models

Received: 27 July 2024

Accepted: 19 December 2024

Published online: 17 March 2025

Check for updates

Jacob Kauffmann[1,2], Jonas Dippel[1,2,3], Lukas Ruff[1,3], Wojciech Samek[1,2,4], Klaus-Robert Müller[1,2,5,6,7] & Grégoire Montavon[1,2,8]

Unsupervised learning has become an essential building block of artifical intelligence systems. The representations it produces, for example, in foundation models, are critical to a wide variety of downstream applications. It is therefore important to carefully examine unsupervised models to ensure not only that they produce accurate predictions on the available data but also that these accurate predictions do not arise from a Clever Hans (CH) effect. Here, using specially developed explainable artifical intelligence techniques and applying them to popular representation learning and anomaly detection models for image data, we show that CH effects are widespread in unsupervised learning. In particular, through use cases on medical and industrial inspection data, we demonstrate that CH effects systematically lead to significant performance loss of downstream models under plausible dataset shifts or reweighting of different data subgroups. Our empirical findings are enriched by theoretical insights, which point to inductive biases in the unsupervised learning machine as a primary source of CH effects. Overall, our work sheds light on unexplored risks associated with practical applications of unsupervised learning and suggests ways to systematically mitigate CH effects, thereby making unsupervised learning more robust.

Unsupervised learning is a subfield of machine learning (ML) that has gained prominence in recent years[1–3]. It addresses fundamental limitations of supervised learning, such as the lack of labels in the data or the high cost of acquiring them. Unsupervised learning has achieved successes in modelling the unknown, such as uncovering new cancer subtypes[4,5] or extracting novel insights from large historical corpora[6]. Furthermore, the fact that unsupervised learning does not rely on task-specific labels makes it a good candidate for core artifical intelligence (AI) infrastructure: unsupervised anomaly detection provides the basis for various quality or integrity checks on the input data[7–10]. Unsupervised learning is also a key technology behind 'foundation models'[1,11–15], which extract representations upon which various downstream models (for example, classification, regression, 'generative AI' and so on) can be built.

The growing popularity of unsupervised learning models creates an urgent need to carefully examine how they arrive at their predictions. This is essential to ensure that potential flaws in the way these models process and represent the input data are not propagated to the many downstream supervised models that build upon them.

In this study, through conducting multiple investigations of popular unsupervised ML models of image data, we show that unsupervised learning models largely suffer from Clever Hans (CH) effects[16]. Specifically, we find that unsupervised learning models often produce representations from which instances can be correctly predicted to

[1]Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Berlin, Germany. [2]BIFOLD—Berlin Institute for the Foundations of Learning and Data, Berlin, Germany. [3]Aignostics, Berlin, Germany. [4]Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, Berlin, Germany. [5]Department of Artificial Intelligence, Korea University, Seoul, Korea. [6]Max-Planck Institute for Informatics, Saarbrücken, Germany. [7]Google Deepmind, Berlin, Germany. [8]Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany. e-mail: klaus-robert.mueller@tu-berlin.de; gregoire.montavon@fu-berlin.de

https://doi.org/10.1038/s42256-025-01000-2

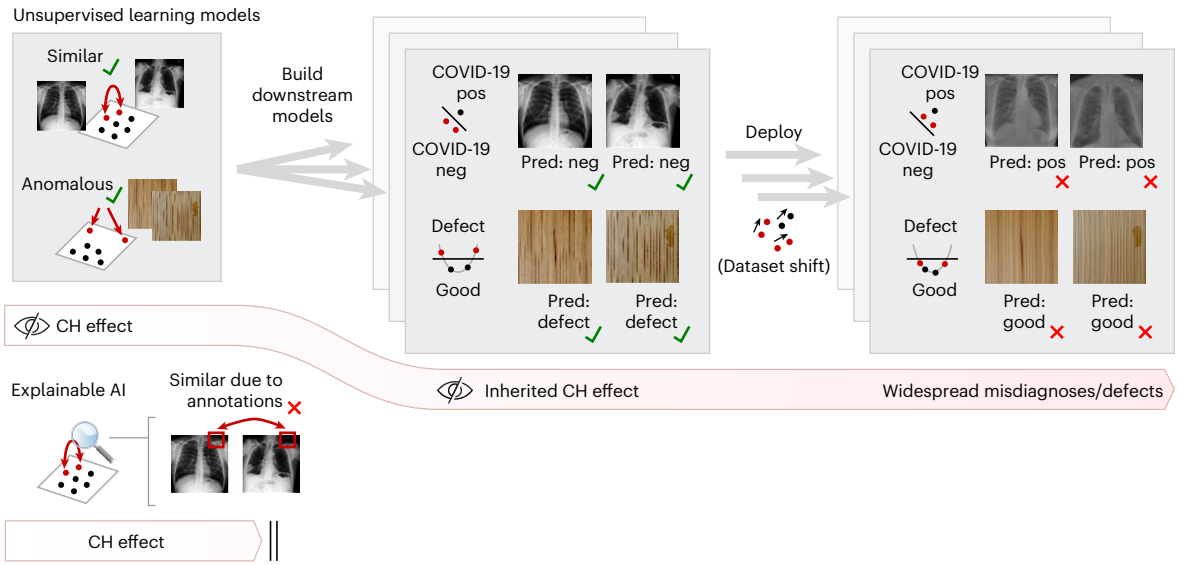

Fig. 1 | The CH effect in unsupervised learning. The unsupervised model correctly predicts data instances as similar or anomalous, but does so using features that do not generalize well outside the available data. The CH effect typically goes undetected in a classical validation scheme and manifests itself in the form of prediction errors only after deployment. The problem is critical because the flaw can be inherited by potentially many downstream tasks. Our explainable AI approach allows CH effects to be detected directly in the unsupervised model and, in some cases, corrected. Pos, positive; pred, predicted; neg, negative. X-ray images reproduced from: left, middle, ref. 79 under a Creative Commons licence CC 1.0; right, ref. 90 under a Creative Commons licence CC BY-3.0.

be, for example, similar or anomalous, although largely supported by data quality artefacts. The flawed prediction strategy is not detectable by common evaluation benchmarks such as cross-validation, but may manifest itself much later in 'downstream' applications in the form of unexpected errors, for example, if subtle changes in the input data occur after deployment (Fig. 1). While CH effects have been studied quite extensively for supervised learning[16–22], the lack of similar studies in the context of unsupervised learning, together with the fact that unsupervised models supply many downstream applications, is a cause for concern.

For example, in image-based industrial inspection, which often relies on unsupervised anomaly detection[9,10], we find that a CH decision strategy can systematically miss a wide range of manufacturing defects, resulting in potentially high costs. As another example, unsupervised foundation models of image data, advocated in the medical domain to provide robust features for various specialized diagnostic tasks, can potentially introduce CH effects into many of these tasks, with the prominent risk of large-scale misdiagnosis. These scenarios (illustrated in Fig. 1) highlight the practical implications of an unsupervised CH effect, which, unlike its supervised counterpart, may not be limited to malfunctioning in a single specific task, but potentially in all downstream tasks.

To uncover and understand unsupervised CH effects, we propose to use explainable AI[23–27] (here techniques that build on the layer-wise relevance propagation (LRP) explanation framework[28–30]). Our proposed use of these techniques allows us to identify at scale which input features are used (or misused) by the unsupervised ML model, without having to formulate specific downstream tasks. We use an extension of LRP called BiLRP[31] to reveal input patterns that are jointly responsible for similarity in the representation space. We also combine LRP with 'virtual layers'[32,33] to reveal pixel and frequency components that are jointly responsible for predicted anomalies.

Furthermore, our explainable AI-based analysis allows us to pinpoint more formal causes for the emergence of unsupervised CH effects. In particular, they are due not so much to the data, but to the unsupervised learning machine, which hinders the integration of the true task-supporting features into the model, even though vast amounts of data points are available. Our findings provide a novel direction for developing targeted strategies to mitigate CH effects and increase model robustness.

Overall, our work sheds light on the presence, prominence and distinctiveness of CH effects in unsupervised learning, calling for increased scrutiny of this essential component of modern AI systems.

Results

The CH effect can be defined as the property of a model to rely on features that are predictive in a particular setting (due to a spurious correlation between them and the true signal), but fail to remain so on new data, causing a significant drop in performance. (See also Supplementary Note D for a formal characterization and distinction from related concepts such as shortcut learning[17] or human–AI alignment[34,35].) Through experiments on two representative families of unsupervised models, representation learning and anomaly detection, and using explainable AI as our main analysis tool, we demonstrate the widespread presence of CH effects in unsupervised learning models, their adverse consequences and possible strategies to mitigate them.

CH effects in representation learning

We first investigate the CH effect in the context of using a recent medical foundation model to solve a COVID-19 detection task. Simulating an early pandemic phase characterized by data scarcity, we aggregate, similar to ref. 19, a large, well-established non-COVID-19 dataset with a more recent and smaller COVID-19 dataset. Specifically, we aggregate 2,597 instances of the National Institute of Health (NIH) CXR8 dataset[36], collected between 1992 and 2015, with the 535 instances of the GitHub-hosted 'COVID-19 image data collection'[37], which contains COVID-19 instances from multiple sources. We refer to them as the 'NIH' and 'GitHub' subsets, respectively.

Further motivated by the need to accommodate the critically small number of COVID-19 instances and to avoid overfitting, we choose to rely on the representations provided by unsupervised foundation models[15,38–40]. Specifically, we feed our data into a pretrained PubMedCLIP model[39], which has built its representation in an unsupervised manner from a very large collection of X-ray scans. On top of the

https://doi.org/10.1038/s42256-025-01000-2

Table 1 | Performance of unsupervised models on various downstream tasks, evaluated on different data subgroups

	COVID-19		ImageNet: truck		ImageNet: fish	
	Original	GitHub	Original	Logo	Original	Human
PubMedCLIP	87.5 (FPR 18%)	81.7 ↓ (FPR 51%)	–	–	–	–
CLIP	–	–	84.7	80.3 ↓	85.4	84.1
CLIP + CH mitigation	–	–	84.4	83.6 ↑	–	–
SimCLR	–	–	74.8	74.8	81.4	74.8 ↓↓
Barlow Twins	–	–	79.2	78.7	83.2	75.8 ↓↓
Supervised	–	–	82.4	82.3	85.9	81.0 ↓

We report PubMedCLIP's accuracy scores and FPRs on the aggregate COVID-19 dataset (original) and the more difficult GitHub subgroup. We repeat the analysis for generic unsupervised models on two ImageNet superclasses, both on the original data and on the difficult subgroups (logo and human). ↓/↓↓ denote a substantial accuracy decrease (exceeding 3/6 percentage points) on the difficult subgroups and ↑ denotes a substantial accuracy increase (of 3 percentage points or more) after CH mitigation. Upward and downward effects are statistically significant under a two-sided *t*-test ($P < 0.001$).

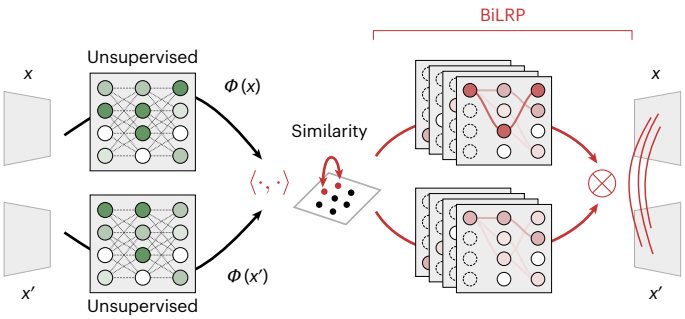

Fig. 2 | Illustration of the BiLRP method for explaining similarity predictions of a representation learning model. The output of BiLRP is a decomposition of the predicted similarity onto pairs of features from the two input images. It is typically displayed as a weighted bipartite graph connecting the contributing feature pairs.

PubMedCLIP model, we train a downstream classifier that separates COVID-19 from non-COVID-19 instances. It achieves a class-balanced accuracy of 87.5% on the test set (Table 1). However, a closer look at the structure of this performance score reveals a strong disparity between the NIH and GitHub subgroups, with all NIH instances being correctly classified and the GitHub instances having a lower class-balanced accuracy of 81.7%, and, more strikingly, a false positive rate (FPR) of 51%, as presented in Table 1. Considering that the higher heterogeneity of instances in the GitHub dataset is more characteristic of real-world conditions, this higher error estimate is more realistic. In particular, the high FPR of 51% precludes any practical use of the model in a hospitalization setting, where the model's prediction should reliably and with low risk assist in the selection of appropriate medical treatment. We emphasize that this flaw in the model could have been easily overlooked if one had not paid close attention to (or known about) the data sources and instead relied only on the overall accuracy score.

To proactively detect this heterogeneous, non-robust prediction behaviour, we propose to use explainable AI. Specifically, to test whether the flaw has its sources in the unsupervised PubMedCLIP component, we use the BiLRP explanation technique[31]. BiLRP operates directly on similarity in the representation space without the need to formulate a specific downstream task. It is illustrated in Fig. 2 and its mathematical formulation is given in Methods. The output of BiLRP for two exemplary pairs of COVID-19-positive instances is shown in Fig. 3 (left). It shows that the modelled similarity comes from text-like annotations that appear in both images. This allows us to attribute the observed heterogeneity in performance to a CH effect and in turn to highlight broad risks for downstream applications (see Supplementary Note A for further analysis). We note that, unlike the per-group accuracy analysis above, our explainable AI analysis based on BiLRP did not require provenance metadata (GitHub or NIH) nor did it focus on a specific downstream task with its specific labels.

To test whether representation learning has a general tendency to evolve CH strategies beyond the above use case, we downloaded three generic foundation models, namely the original CLIP model[13], SimCLR[12,41] and Barlow Twins[42]. CLIP consists of an image encoder and a text encoder, and it aligns images to their associated text in representation space by minimizing a contrastive loss. SimCLR and Barlow Twins generate augmented views of the input image through random resized crops and colour augmentation, and maximize the similarity of these two views in representation space. As a downstream task, we consider the classification, using linear-softmax classifiers, of the 8 classes from ImageNet[43] that share the WordNet ID 'truck' and of the 16 ImageNet classes that share the WordNet ID 'fish' (see Methods for details). The test accuracy of each model on these two tasks is given in Table 1 (columns 'original'). On the truck classification task, the CLIP model performs best, with an accuracy of 84.7%. On the fish classification task, the CLIP and supervised models perform best, with accuracies of 85.4% and 85.9%, respectively.

We use BiLRP to examine the representations of these unsupervised models. In Fig. 3 (centre), we observe that CLIP-based similarities, as in PubMedCLIP, also rely on text. Here, a textual logo in the lower-left corner of two garbage truck images is used to support the similarity, suggesting a CH effect (see ref. 20 for a similar finding in supervised learning). SimCLR and Barlow Twins ignore the text and rely instead on the actual garbage truck. In the fish classification task (Fig. 3, right), we observe that all unsupervised models amplify humans over fish features, again suggesting a CH effect.

To establish the CH nature of the logo and human detection strategies identified by BiLRP, we proceed to test the models on specific data subgroups that may be more prevalent under operational conditions. The results are presented in Table 1. We observe a systematic degradation in performance when moving from the original data to some of these data subsets. For example, when we break the spurious correlation between logo and truck class by inserting a logo on each truck image, we observe a drop in the accuracy of the CLIP model from 84.7% to 80.3% (column 'logo' in Table 1). Sharper drops in performance can be observed when looking at individual classes, such as tow trucks, which are generally difficult to separate from garbage trucks (Supplementary Note B). For the fish case, a similar drop in accuracy is observed for SimCLR and Barlow Twins from 81.4% and 83.2% to 74.8% and 75.8%, respectively, when only images containing humans are retained and class rebalancing is performed. In the case of CLIP, its lack of focus on fish is surprisingly not associated with a similar drop in performance, leaving open the question of what exact strategy allows CLIP to generalize well on this data. A detailed analysis of the structure of the prediction errors for each model and classification task, supported by confusion matrices, is given in Supplementary Note B.

To better assess the risk of CH effects in unsupervised learning, it is necessary to reflect on the more abstract factors that contribute to their occurrence. The heterogeneity of strategies revealed by BiLRP for models otherwise trained on similar large datasets suggests that the unsupervised learning machine, more than the data, is crucial in shaping the data representation strategy. In the case of SimCLR and Barlow Twins, the systematic amplification of humans in the centre of the image can be attributed to their random crop matching objective, where those features in the centre of the image carry the most mutual information across random crops (for further studies of amplification/suppression effects in these models, we refer to refs. 44–47). When

https://doi.org/10.1038/s42256-025-01000-2

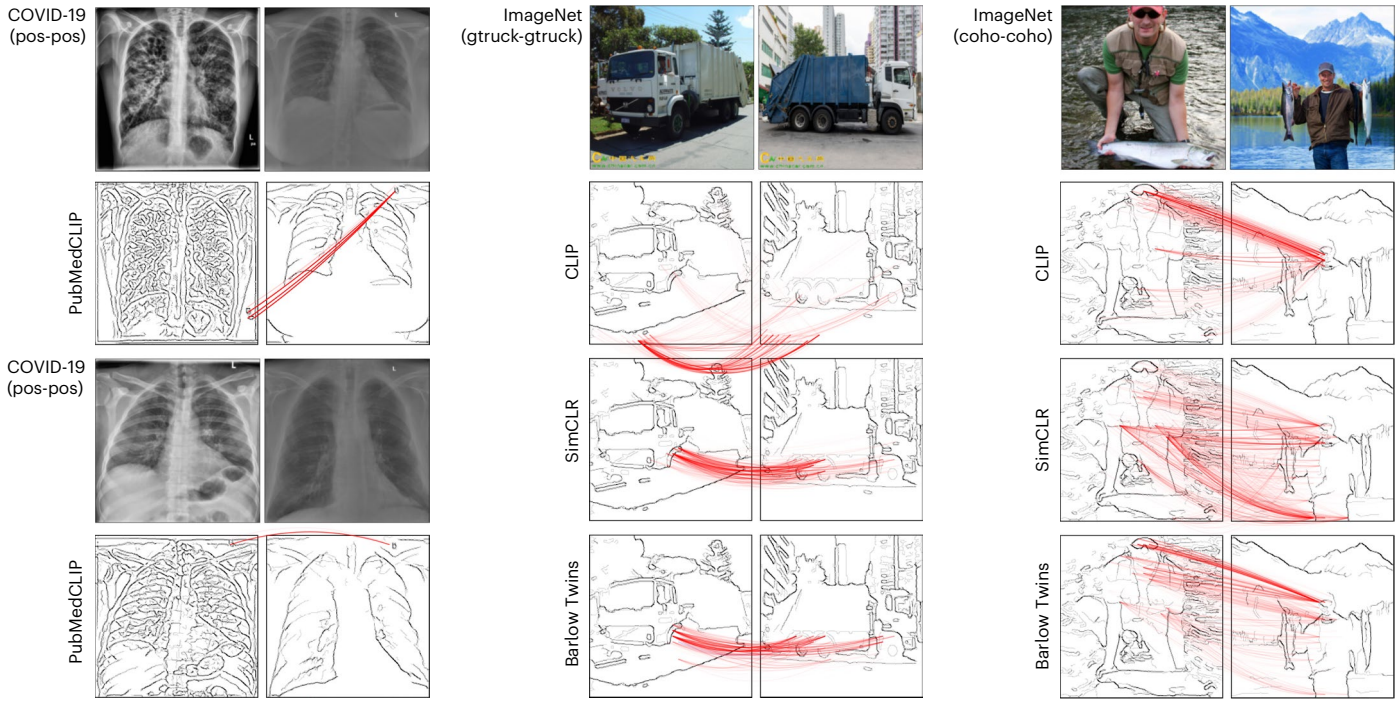

Fig. 3 | Explainable AI analysis of the predictions of the PubMedCLIP unsupervised model and the general-purpose CLIP, SimCLR and Barlow Twins unsupervised models. We show pairs of X-ray images from the GitHub subset, and pairs of natural images resembling ImageNet images from the classes garbage truck (gtruck) and coho, respectively. Explanations are generated using BiLRP. They highlight unexpected strategies used by the unsupervised models: for example, for X-ray data, similarity between instances arises from shared spurious textual annotations. For ImageNet data, similarity arises from logo artefacts or the presence of humans in the background. X-ray images reproduced from ref. 90 under a Creative Commons license CC BY-3.0. Credit: truck (left), Pixnio under a Creative Commons licence CC 1.0; truck (right), Pexels under a Creative Commons licence CC 1.0; fish (left), iStock.com/christiannafzger; fish (right), iStock.com/BrandyTaylor.

considering the CLIP and PubMedCLIP models, the systematic amplification of textual logos, faces or other identifying features can be attributed to their image–text matching objective, which tends to amplify any features from the two modalities that carry mutual information.

In summary, while the matching tasks defined in, for example, CLIP, SimCLR and Barlow Twins intuitively aim to introduce useful prior knowledge and invariance into the representation, they can, on certain data subsets, lead to strong imbalances in the expression of different features. These imbalances are prone to cause CH effects and, in turn, loss of accuracy in downstream tasks.

CH effects in anomaly detection

Extending our investigation of the CH effect to another area of unsupervised learning, namely anomaly detection, we consider an industrial inspection use case based on the popular MVTec-AD dataset[9]. The dataset consists of 15 product categories, each consisting of a training set of images without manufacturing defects and a test set of images with and without defects. Since manufacturing defects are infrequent and heterogeneous in nature, the problem is typically approached using unsupervised anomaly detection[2,9]. These models map each instance to an anomaly score, from which threshold-based downstream models can be built to classify between instances with and without manufacturing defects. Unsupervised anomaly detection has received considerable attention, with sophisticated approaches based on deep neural networks such as PatchCore[48] or EfficientAD[49] showing excellent performance in detecting a wide range of industrial defects.

Somewhat surprisingly, simpler approaches based on distances in pixel space show competitive performance for selected tasks[2]. We consider one such approach, which we call 'D2Neighbors', where anomalies are predicted according to the distance to neighbours in the training data. Specifically, the anomaly score of a new instance $\mathbf{x}$ is computed as $f(\mathbf{x}) = \mathrm{softmin}_j\{\|\mathbf{x}-\mathbf{u}_j\|^2\}$ where $(\mathbf{u}_j)_{i=1}^N$ is the set of available inlier instances (see Methods for details on the model and data preprocessing). This anomaly model belongs to the broader class of distance-based models[50–52], and connections can be made to kernel density estimation[53,54] and one-class support vector machines[55]. Using D2Neighbors, we are able to build downstream models that classify industrial defects of the MVTec data with F1 scores above 0.9 for five categories (bottle, capsule, pill, toothbrush and wood).

To shed light on the prediction strategy associated with these unexpectedly high F1 scores, we make use of explainable AI. Specifically, we consider an extension of LRP for anomaly detection[30,56] and further equip the explanation technique with 'virtual layers'[32,33]. The technique of 'virtual layers' (Fig. 4) is to map the input to an abstract domain and back, leaving the prediction function unchanged, but providing a new representation in terms of which the prediction can be explained. We construct such a layer by applying the discrete cosine transform (DCT)[57], shown in Fig. 4 (bottom right), followed by its inverse. This allows us to explain the predictions jointly in terms of pixels and frequencies.

The result of our proposed analysis is shown in Fig. 4 for two wood instances (see Supplementary Note C for instances of different categories). Explanations at the pixel level show that D2Neighbors supports its anomaly predictions largely based on pixels containing the actual industrial defect. The squared difference in its distance function ($\|\Delta\|^2 = \sum_i \Delta_i^2$) encourages a sparse pixel-wise response of the model, efficiently discarding regions of the image where the new instance shows no difference from instances in the training data. However, we also see in the pixel-wise explanation that a non-negligible part of the anomaly prediction comes from irrelevant background pixels. Joint pixel-frequency explanations shed light on these unresolved contributions, showing that they arise mostly from the high-frequency part of the model's decision strategy (Fig. 4b,c).

https://doi.org/10.1038/s42256-025-01000-2

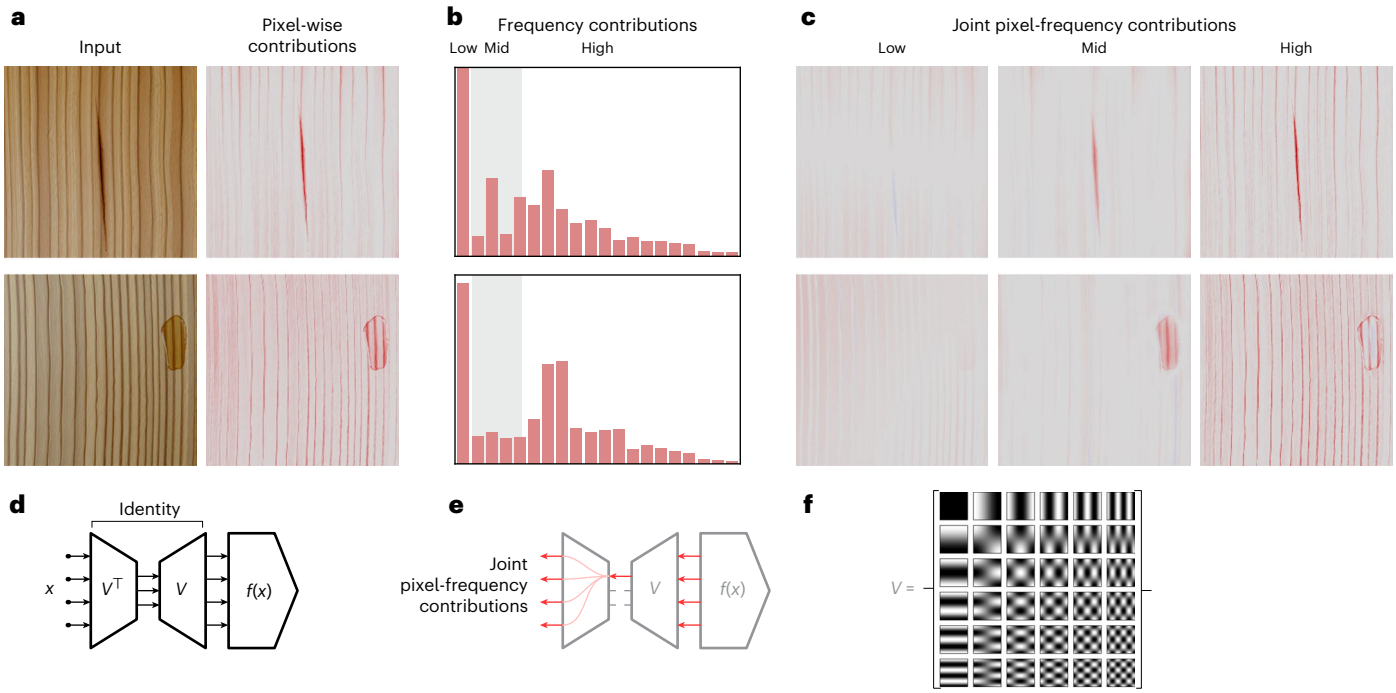

Fig. 4 | Explainable AI analysis of D2Neighbors anomaly predictions. a, Images synthesized to resemble MVTec-AD (class wood) and pixel-wise LRP explanations of the anomaly predictions. The explanations for other MVTec-AD categories are given in Supplementary Note C. **b**, Frequency domain explanations. The *x* axis represents the frequencies (on a power scale) and the *y* axis is the contribution of the corresponding frequencies to the anomaly prediction. **c**, Pixel-wise contributions filtered by frequency band. **d**, A schematic of the virtual inspection layer used to explain anomalies in the joint pixel-frequency domain. **e**, Pixel-wise contributions are filtered by blocking frequency contributions within the virtual layer. **f**, The basis elements of the DCT, which we use to map pixels to frequencies and back.

The high exposure of the D2Neighbors model to these irrelevant high-frequency features, as detected by our LRP analysis, raises the suspicion that we are again in the presence of a CH effect. We simulate an innocuous postdeployment perturbation of the data preprocessing by changing the image resizing algorithm from OpenCV's nearest neighbour resizing to a more sophisticated resizing method that includes antialiasing, a procedure that cuts high frequencies to eliminate resizing artefacts. In practice, such a change may result from a software update, for example. Resizing techniques have been shown in some cases to substantially affect image quality and image generation metrics[58], but their effect on general ML models, especially unsupervised ones, has been little studied. The performance of the D2Neighbors model before and after changing the resizing algorithm is presented in Table 2 (columns 'original' and 'deployed', respectively). The F1 score performance of D2Neighbors degrades by almost 10 percentage points. This performance degradation, along with D2Neighbors' reliance on high frequencies revealed by LRP, exposes the CH nature of the model: when antialiasing is introduced into the resizing procedure, the high frequencies that the D2Neighbors model uses to support its prediction disappear from the data, significantly reducing each instance's anomaly score and causing the performance degradation. This performance degradation of D2Neighbors under postdeployment conditions is particularly surprising given that the data quality has actually improved. Looking more closely at the structure of the performance degradation, we see that the false negative rate (FNR) rises sharply from 4% to 23% (Table 2), which can be explained by the absence of anomaly-contributing high frequencies after deployment. In an industrial inspection setting, an increase in FNR can have serious consequences, in particular, many defective instances may be missed and propagated through the production chain. This can result in wasted resources in subsequent production stages and high recall costs.

Table 2 | Performance of different anomaly detection models on simulated original and postdeployment data conditions

	Original	Deployed
D2Neighbors	0.91 (FNR 4%)	0.82 ↓ (FNR 23%)
D2Neighbors + CH mitigation	0.92	0.92 ↑
D2Neighbors (ℓ_1)	0.92	0.83 ↓
D2Neighbors (ℓ_4)	0.91	0.84 ↓
PatchCore	0.92	0.86 ↓

These conditions correspond to standard and antialiased resizing, respectively. Performance is reported in terms of F1 score and FNR, averaged over the five MVTec-AD categories retained for analysis. ↓ Shows a substantial F1 score decrease (of 3 percentage points or more) after deployment and ↑ shows a substantial F1 score increase (of 3 percentage points or more) after CH mitigation. Upwards and downwards effects are statistically significant under a two-sided *t*-test ($P<0.001$).

As in the case of representation learning, it is useful to ask what factors contribute to the CH effect. We trace the D2Neighbors CH strategy to the distance functions it relies on. Unlike the linear layers commonly used in supervised learning, distances cannot inherently build invariance to specific directions in the input space, exposing models to a manifold of irrelevant data perturbations. Despite this tendency to overexposure, distance functions (including the usual Euclidean distance as well as ℓ_p variants) are a building block of many popular unsupervised anomaly models, their primary advantage being that they generate a decision boundary without requiring a representative set of anomalous instances to contrast against. Distance functions also appear in the more advanced PatchCore model[48] where they are computed on top of more abstract visual features (Methods). As presented in Table 2, they also suffer a significant drop in performance after

deployment, suggesting that they are affected by a similar CH effect. Overall, our analysis highlights the challenge of creating anomaly models that are both general enough not to miss unexpected anomalies, but also not overexposed so as not to increase the risk of CH effects.

Alleviating CH in unsupervised learning

Leveraging the explainable AI analysis above, we aim to build models that are more robust across different data subgroups and in postdeployment conditions. Unlike previously proposed CH removal techniques[20,21], we aim to operate on the unsupervised model rather than the downstream tasks. This allows us to potentially achieve broad robustness improvements while leaving the downstream learning machines (training supervised classifiers or adjusting detection thresholds) untouched. We first consider the CLIP model, which our explainable AI analysis has shown to incorrectly rely on text logos, and proceed by removing CLIP activations whose response differs most between images of the logo and non-logo subgroups (details in Methods). We also experiment with a CH mitigation approach for anomaly detection, where we prune the high frequencies spuriously used by the model by inserting a blur layer at the input of the model (details in Methods). In both cases, the proposed CH mitigation technique improves model robustness, largely reversing the performance degradation observed in simulated postdeployment conditions (Tables 1 and 2, rows 'CH mitigation'). Our CH mitigation experiments, which effectively modify the structure of the model, again underscore the primary role of the learning machine in allowing or preventing CH effects.

Discussion

Unsupervised learning is an essential category of ML that is increasingly being used in core AI infrastructure to power a variety of downstream tasks, including classification, regression and also 'generative AI'. Much research so far has focused on improving the performance of unsupervised learning algorithms, for example, to maximize downstream classification accuracy. These evaluations often pay little attention to the exact strategy used by the unsupervised model to achieve the reported high performance, in particular whether these models rely on CH strategies.

Using advanced explainable AI techniques such as BiLRP or LRP in the frequency domain, we have shown that CH strategies are widespread in unsupervised learning. These strategies can take several forms, such as predicting correctly but based on features such as text that are spuriously amplified in the unsupervised representation, or based on high-frequency features to which unsupervised anomaly models are overexposed. These flawed prediction strategies no longer work well when the data distribution changes after deployment. As shown in two use cases, this can have important practical consequences such as widespread misdiagnosis of patients or systematic failure to recall manufacturing defects. Importantly, the same flawed unsupervised representation can produce CH effects in any of its potentially many downstream models.

Addressing these CH effects is therefore crucial to apply unsupervised learning more reliably. However, compared with CH effects in supervised learning, another dimension of complexity is added to the problem: one has to decide whether to handle CH effects in the downstream models or directly in the unsupervised model part. Revising downstream models (for example, with human feedback[20,21,59] or in response to changing conditions[60–63]) may help to maintain high accuracy on the given task. However, it is not sustainable if we consider that the procedure would have to be repeated for every single downstream task. This may be necessary even after a flaw in the foundation model becomes known (for example, refs. 64,65) since building a new unsupervised model is computationally expensive and requires extensive testing. Instead, we have proposed in this paper to address CH effects directly in the design of the unsupervised model, with the goal of achieving persistent robustness that benefits all existing and future downstream applications.

However, this requires a better formal understanding of the reasons for CH effects in unsupervised learning. We found that they differ substantially from those in supervised learning in that they arise less from data quality issues and more from flaws in the design of the unsupervised learning machine. For example, our study showed that unsupervised anomaly detection is structurally unable to reduce its exposure to high frequencies and thus also fails to reproduce common filtering mechanisms found in supervised learning[66–68], with D2Neighbors being a prominent example. The high risk of generalization error caused by feature overexposure led us to ask the more fundamental question of 'what are appropriate model selection criteria for unsupervised learning'. D2Neighbors, with its apparent simplicity, would probably fare well under Occam's razor or other classical model selection criteria, although our experiments have shown that it clearly lacks generalizability and robustness. Thus, it seems essential to refine these criteria to include overexposure or feature balancing as additional factors.

Having shed light on reasons for the emergence of CH effects in unsupervised learning, we have experimented with CH mitigation strategies based on feature rebalancing or exposure reduction, and have been able to achieve performance improvements on difficult data subgroups or in simulated postdeployment conditions. In doing so, we have demonstrated the actionability of our analysis, showing that it can guide the process of identifying and subsequently correcting the faulty components of an unsupervised learning model.

While our investigation of unsupervised CH effects and their consequences has focused on image data, extension to other data modalities seems straightforward. Explainable AI techniques such as LRP operate independently of the type of input data. LRP has recently been extended to recurrent neural networks[69], graph neural networks[70], transformers[71] and state space models[72], which represent the state of the art for large language models and other models of structured data. Thus, our analysis could be extended in the future to analyse other instances of unsupervised learning, such as anomaly detection in time series or the representations learned by large language models (for example, refs. 73,74).

Overall, through the application of recent explainable AI techniques, our work has contributed to highlighting the pervasiveness of CH effects in unsupervised learning, the multiple factors that lead to them, the resulting loss of accuracy on new data and possible ways to mitigate these CH effects. We believe that the CH effect in unsupervised learning, and the uncontrolled risks associated with it, is a question of general importance, and that explainable AI and its recent developments provide an effective way to tackle it.

Methods

This section first introduces the unsupervised ML models studied in this work, the datasets on which they are applied and the considered CH mitigation techniques. It then presents the LRP method for explaining predictions, its BiLRP extension for explaining similarity and the technique of 'virtual layers' for generating joint pixel-frequency explanations.

ML models and data for representation learning

Representation learning experiments were performed on the PubMedCLIP[39], CLIP[13], SimCLR[12,41] and Barlow Twins[42] models. PubMedCLIP is a representation learning model specialized for X-ray data. It is based on a pretrained CLIP model (described below) and fine tuned on the ROCO dataset[75], a collection of radiology and image caption pairs. In our experiments, we chose the variant based on the ResNet-50 architecture and downloaded the weights from ref. 76. CLIP learns representations using a large collection of image–text pairs from the internet. Images are given to an image encoder and the corresponding texts are given to a text encoder. The similarity of the two resulting embeddings is then maximized with a contrastive loss. In our experiments, we again chose

the ResNet-50 variant with weights from ref. 77. SimCLR augments the input images with resized crops, colour jitter and Gaussian blur to create two different views of the same image. These views are then used to create positive and negative pairs, where the positive pairs represent the same image from two different perspectives and the negative pairs are created by pairing different images. The contrastive loss objective maximizes the similarity between the representations of the positive pairs while minimizing the similarity between the representations of the negative pairs. In our experiments, we used the ResNet-50 architecture and weights from the vissl library (https://vissl.ai/). Barlow Twins is similar to SimCLR in that it also generates augmented views of the input image through randomly resized crops and colour augmentation, and maximizes their similarity in representation space. However, it differs from SimCLR in the exact mechanisms used to prevent representation collapse. In our experiments, we again used the ResNet-50 architecture and took the weights from ref. 78. For our representation learning experiments, we also considered a supervised baseline, with the same ResNet-50 architecture, but trained in a purely supervised fashion using backpropagation. We used the default model weights from the torchvision library.

Downstream classifiers. To establish the CH effect in these unsupervised models, specifically its manifestation in downstream tasks, we built linear classifiers (readouts) on top of the unsupervised representations. For binary detection tasks, specifically detection of COVID-19 instances, we trained a linear support vector machine classifier (details in Supplementary Note A), with the slack parameter C set to 0.01 through a hold-out validation procedure. For multiclass classification problems (classifying among the 8 types of trucks and among the 16 types of fishes), we instead used a logistic regression classifier (sklearn) with the lbfgs solver, l2 regularization ($C = 1.0$), no bias term, a maximum of 1,000 iteration steps and class-balanced sampling.

Datasets. The analysis and training of these models were performed on different datasets. For the X-ray experiments, we combined the NIH ChestX-ray8 (CXR8) dataset[36,79] and the GitHub-hosted 'COVID-19 image data collection'[37,80]. The GitHub dataset contains 342 COVID-19-positive and 193 COVID-19-negative images. We split the data 80:20 into training and test sets. This resulted in 272 positive and 168 negative images in the training set and 70 positive and 25 negative images in the test set. The training split was consolidated by adding 2,552 randomly selected negative images from the NIH dataset. We also expanded the test set by adding another 45 randomly selected negative images from NIH to obtain a class-balanced test set. The selection was made so that the same patient IDs did not appear in both the training and test sets. All images were resized and centre-cropped to 224 × 224 pixels. The ImageNet experiments were performed on two ImageNet subsets. First, the 'truck' subset, consisting of the eight classes sharing the WordNet ID 'truck' (minivan, moving van, police van, fire engine, garbage truck, pickup, tow truck and trailer truck), resulting in a dataset of 10,259 training and 400 test examples. Then the 'fish' subset, consisting of the 16 classes sharing the WordNet ID 'fish' (tench, barracouta, coho, sturgeon, gar, stingray, great white shark, hammerhead, tiger shark, puffer, electric ray, goldfish, eel, anemone fish, rock beauty and lionfish), resulting in 20,334 training and 800 test examples.

ML models and data for anomaly detection

The D2Neighbors model used in our experiments is an instance of the family of distance-based anomaly detectors, which encompasses a variety of methods from the literature[2,50–52,81,82]. The D2Neighbors model computes anomaly scores as $o(\mathbf{x}) = \mathbb{M}_j^{\gamma}\{\| \mathbf{x} - \mathbf{u}_j \|_p^p\}$ where $\mathbf{x}$ is the input, $(\mathbf{u}_j)_{j=1}^N$ are the training data and $\mathbb{M}^{\gamma}$ is a generalized f-mean, with $f(t) = \exp(-\gamma t)$. The predicted anomaly scores can be interpreted as a soft minimum over distances to data points, that is, a distance to the nearest neighbours. In our experiments, the data received as input are images of size 224 × 224 with pixel values encoded between −1 and 1, downsized from their original high resolution using OpenCV's fast nearest neighbour interpolation. We set γ so that the average perplexity[83] equals 25% of the training set size for each model.

We also considered the PatchCore[48] anomaly detection model, which uses mid-level patch features from a fixed pretrained network. It constructs a memory bank of these features from nominal example images during training. Anomaly scores for test images are computed by finding the maximum distance between each test patch feature and its nearest neighbour in the memory bank. Distances are computed between patch features $\phi_p(\mathbf{x})$ and a memory bank of location-independent prototypes $(\mathbf{u}_j)_{j=1}^N$. The overall outlier scoring function of PatchCore can be written as $o(\mathbf{x}) = \max_k \min_j \| \phi_k(\mathbf{x}) - \mathbf{u}_j \|$. The function ϕ_k is the feature representation aggregated from two consecutive layers at spatial patch location k, extracted from a pretrained WideResNet50. The features from consecutive layers are aggregated by rescaling and concatenating the feature maps. The difference between our reported F1 scores and those in ref. 48 is mainly due to the method used to resize the images. We used the authors' reference implementation[84] as the basis for our experiments.

Datasets. All models above were trained on the MVTec-AD dataset. The MVTec-AD dataset consists of 15 image categories ('bottle', 'cable', 'capsule', 'carpet', 'grid', 'hazelnut', 'leather', 'metal nut', 'pill', 'screw', 'tile', 'toothbrush', 'transistor', 'wood' and 'zipper') of industrial objects and textures, with good and defective instances for each category. For the experiments based on D2Neighbors, we simulated different data preprocessing conditions before and after deployment by changing the way images are resized from their original high resolution to 224 × 224 pixels. We first used a resizing algorithm found in OpenCV v.4.9.0 (ref. 85) that is based on nearest neighbour interpolation. We then simulated postdeployment conditions using an improved resizing method, specifically a bilinear interpolation implemented in Pillow v.10.3.0 and used by default in torchvision v.0.17.2 (ref. 86). This improved resizing method includes antialiasing, which has the effect of smoothing the transitions between adjacent pixels of the resized image.

Details of CH mitigation techniques

We describe in detail the CH mitigation techniques we use to mitigate the reliance of ML models on spurious features. To prune textual logos in the CLIP model, we computed responsiveness by measuring the difference in activation between a set of randomly selected truck images with and without a watermark logo, and then pruning (that is, setting to zero) the top k filters in the bottom of the image (we pruned five such filters in the main paper and experimented with different values of k in the Supplementary Information). We looked at multiple layers, and chose an early layer of the CLIP model (encoder.relu3) as it showed a large difference on just a few filters compared to more abstract layers later in the network. In our anomaly detection experiments, where our analysis revealed a spurious use of high frequencies, we proposed to address the CH effect by pruning those high frequencies, specifically by adding a low-pass filter at the input of the model, which convolves the red, green and blue channels individually with Gaussian filters of size 11 × 11.

Explanations for representation learning

Our experiments examined dot product similarities in representation space, that is, $y = \langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{x}') \rangle$, where $\mathbf{\Phi}$ denotes the function that maps the input features to the representation, typically a deep neural network. To explain similarity scores in terms of input features, we used the BiLRP technique[31] which extends the LRP technique[26,28,29,87] for this specific purpose. The conceptual starting point of BiLRP is the observation that a dot product is a bilinear function of its input. BiLRP then

proceeds by reverse propagating the terms of the bilinear function to pairs of activations from the layer below and iterating down to the input. Denoting by $R_{kk'}$ the contribution of neurons k and k' to the similarity score in some intermediate layer in the network, BiLRP extracts the contributions of pairs of neurons j and j' in the layer below via the propagation rule

$$R_{jj'} = \sum_{kk'} \frac{z_{jk} z_{j'k'}}{\sum_{jj'} z_{jk} z_{j'k'}} R_{kk'}. \quad (1)$$

In this formula, z_{jk} denotes the contribution of neuron j to the activation of neuron k. In practice, the reverse propagation procedure above can be implemented equivalently, but more efficiently and easily, by computing a collection of standard LRP explanations (one for each neuron in the representation layer) and recombining them in a multiplicative manner

$$\text{BiLRP}(y) = \sum_k \text{LRP}(\Phi_k(\mathbf{x})) \otimes \text{LRP}(\Phi_k(\mathbf{x}')). \quad (2)$$

Overall, assuming the input consists of d features, BiLRP produces an explanation of size $d \times d$, which is typically represented as a weighted bipartite graph between the set of features of the two input images. Due to the large number of terms, pixel-to-pixel contributions are aggregated into patch-to-patch contributions, and elements of the BiLRP explanations that are close to zero are omitted in the final explanation rendering. In our experiments, we computed BiLRP explanations using the Zennit implementation of LRP[88], which handles the ResNet-50 architecture, and set Zennit's LRP parameters to their default values.

Explanations for the D2Neighbors model

The D2Neighbors model we investigate for anomaly detection is a composition of a distance layer and a soft min-pooling layer. To handle these layers, we use the purposely designed LRP rules of refs. 30,56. Propagation in the softmin layer ($\mathbb{M}_j^y$) is given by the formula

$$R_j = \frac{f(\|\mathbf{x} - \mathbf{u}_j\|_p^p)}{\sum_j f(\|\mathbf{x} - \mathbf{u}_j\|_p^p)} o(\mathbf{x}), \quad (3)$$

a 'min-take-most' redistribution, where f is the same function as in $\mathbb{M}_j^y$. Each score R_j can be interpreted as the contribution of the training point $\mathbf{u}_j$ to the anomaly of $\mathbf{x}$. To further propagate these scores into the pixel-frequency domain, we adopt the framework of 'virtual layers'[32,33] and adapt it to the D2Neighbors model. As a frequency basis, we use the DCT[57], shown in Fig. 4 (bottom right), which we denote by its collection of basis elements $(\mathbf{v}_k)_k$. Since the DCT forms an orthogonal basis, we have the property $\sum_k \mathbf{v}_k \mathbf{v}_k^\top = I$, and multiplication by the identity matrix can be interpreted as a mapping to the frequencies and back. For the special case where $p = 2$, the distance terms in D2Neighbors reduce to the squared Euclidean norm $\|\mathbf{x} - \mathbf{u}_j\|^2$. These terms can be developed to identify pixel–pixel-frequency interactions: $\|\mathbf{x} - \mathbf{u}_j\|^2 = (\mathbf{x} - \mathbf{u}_j)^\top (\sum_k \mathbf{v}_k \mathbf{v}_k^\top)(\mathbf{x} - \mathbf{u}_j) = \sum_k \sum_{ii'} [\mathbf{x} - \mathbf{u}_j]_i [\mathbf{x} - \mathbf{u}_j]_{i'} [\mathbf{v}_k]_i [\mathbf{v}_k]_{i'}$. From there, one can construct an LRP rule that propagates the instance-wise relevance R_j to the pixel–pixel-frequency features:

$$R_{ii'k} = \sum_j \frac{[\mathbf{x} - \mathbf{u}_j]_i [\mathbf{x} - \mathbf{u}_j]_{i'} [\mathbf{v}_k]_i [\mathbf{v}_k]_{i'}}{\epsilon + \|\mathbf{x} - \mathbf{u}_j\|^2} R_j, \quad (4)$$

where the variable ϵ is a small positive term that handles the case where $\mathbf{x}$ and $\mathbf{u}_j$ overlap. A reduction of this propagation rule can be obtained by marginalizing over interacting pixels ($R_{ik} = \sum_{i'} R_{ii'k}$). Further reductions can be obtained by marginalizing over pixels ($R_k = \sum_i R_{ik}$) or frequencies ($R_i = \sum_k R_{ik}$). These reductions are used to generate the heat maps in Fig. 4.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All data used in this paper, in particular the NIH CXR8 (ref. 36), 'COVID-19 image data collection'[37], ImageNet[43] and MVTec-AD[9] datasets, as well as the pretrained models, are publicly available. The URLs for these datasets and models are given in Methods.

Code availability

The full code for reproducing our results is available via Zenodo at https://doi.org/10.5281/zenodo.14186119 (ref. 89).

References

1. Brown, T. B. et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems, NeurIPS* Vol. 33 (eds Larochelle, H. et al.) 1877–1901 (Curran Associates, 2020).
2. Ruff, L. et al. A unifying review of deep and shallow anomaly detection. *Proc. IEEE* **109**, 756–795 (2021).
3. Krishnan, R., Rajpurkar, P. & Topol, E. J. Self-supervised learning in medicine and healthcare. *Nat. Biomed. Eng.* **6**, 1346–1352 (2022).
4. Li, A. et al. Unsupervised analysis of transcriptomic profiles reveals six glioma subtypes. *Cancer Res.* **69**, 2091–2099 (2009).
5. Jiang, L., Xiao, Y., Ding, Y., Tang, J. & Guo, F. Discovering cancer subtypes via an accurate fusion strategy on multiple profile data. *Front. Genet.* **10**, 20 (2019).
6. Eberle, O. et al. Historical insights at scale: a corpus-wide machine learning analysis of early modern astronomic tables. *Sci. Adv.* **10**, eadj1719 (2024).
7. Rettig, L., Khayati, M., Cudré-Mauroux, P. & Piórkowski, M. in *Applied Data Science* 289–312 (Springer, 2019).
8. Eskin, E., Arnold, A., Prerau, M. J., Portnoy, L. & Stolfo, S. J. in *Applications of Data Mining in Computer Security, Advances in Information Security* 77–101 (Springer, 2002).
9. Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D. & Steger, C. The MVTec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *Int. J. Comput. Vis.* **129**, 1038–1059 (2021).
10. Zipfel, J. et al. Anomaly detection for industrial quality assurance: a comparative evaluation of unsupervised deep learning models. *Comput. Ind. Eng.* **177**, 109045 (2023).
11. Bommasani, R. et al. On the opportunities and risks of foundation models. Preprint at https://doi.org/10.48550/arXiv.2108.07258 (2021).
12. Chen, T., Kornblith, S., Swersky, K., Norouzi, M. & Hinton, G. E. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems, NeurIPS* Vol. 33 (eds Larochelle, H. et al.) 22243–22255 (Curran Associates, 2020).
13. Radford, A. et al. Learning transferable visual models from natural language supervision. In *ICML Proc. Machine Learning Research* Vol. 139, 8748–8763 (2021).
14. Moor, M. et al. Foundation models for generalist medical artificial intelligence. *Nature* **616**, 259–265 (2023).
15. Dippel, J. et al. RudolfV: a foundation model by pathologists for pathologists. Preprint at https://doi.org/10.48550/arXiv.2401.04079 (2024).
16. Lapuschkin, S. et al. Unmasking Clever Hans predictors and assessing what machines really learn. *Nat. Commun.* **10**, 1096 (2019).
17. Geirhos, R. et al. Shortcut learning in deep neural networks. *Nat. Mach. Intell.* **2**, 665–673 (2020).
18. Schramowski, P. et al. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nat. Mach. Intell.* **2**, 476–486 (2020).

19. DeGrave, A. J., Janizek, J. D. & Lee, S.-I. Ai for radiographic COVID-19 detection selects shortcuts over signal. *Nat. Mach. Intell.* **3**, 610–619 (2021).
20. Anders, C. J. et al. Finding and removing Clever Hans: using explanation methods to debug and improve deep models. *Inf. Fusion* **77**, 261–295 (2022).
21. Linhardt, L., Müller, K.-R. & Montavon, G. Preemptively pruning Clever-Hans strategies in deep neural networks. *Inf. Fusion* **103**, 102094 (2024).
22. Winkler, J. K. et al. Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition. *JAMA Dermatol.* **155**, 1135–1141 (2019).
23. Montavon, G., Samek, W. & Müller, K.-R. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* **73**, 1–15 (2018).
24. Gunning, D. et al. XAI—explainable artificial intelligence. *Sci. Robot.* **4**, eaay7120 (2019).
25. Arrieta, A. B. et al. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **58**, 82–115 (2020).
26. Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J. & Müller, K.-R. Explaining deep neural networks and beyond: a review of methods and applications. *Proc. IEEE* **109**, 247–278 (2021).
27. Klauschen, F. et al. Toward explainable artificial intelligence for precision pathology. *Annu. Rev. Pathol.* **19**, 541–570 (2024).
28. Bach, S. et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **10**, e0130140 (2015).
29. Montavon, G., Binder, A., Lapuschkin, S., Samek, W. & Müller, K.-R. in *Lecture Notes in Computer Science* Vol. 11700 (eds Samek, W. et al.) 193–209 (Springer, 2019).
30. Kauffmann, J., Müller, K.-R. & Montavon, G. Towards explaining anomalies: a deep Taylor decomposition of one-class models. *Pattern Recognit.* **101**, 107198 (2020).
31. Eberle, O. et al. Building and interpreting deep similarity models. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 1149–1161 (2022).
32. Vielhaben, J., Lapuschkin, S., Montavon, G. & Samek, W. Explainable AI for time series via virtual inspection layers. *Pattern Recognit.* **150**, 110309 (2024).
33. Chormai, P., Herrmann, J., Müller, K.-R. & Montavon, G. Disentangled explanations of neural network predictions by finding relevant subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 7283–7299 (2024).
34. Zhou, C. et al. LIMA: less is more for alignment. In *Advances in Neural Information Processing Systems, NeurIPS* Vol. 36 (eds Oh, A. et al.) 55006–55021 (Curran Associates, 2023).
35. Muttenthaler, L., Dippel, J., Linhardt, L., Vandermeulen, R. A. & Kornblith, S. Human alignment of neural network representations. In *Proc. International Conference on Learning Representations (ICLR)* (OpenReview.net, 2023).
36. Wang, X. et al. ChestX-Ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 3462–3471 (IEEE, 2017).
37. Cohen, J. P. et al. COVID-19 image data collection: prospective predictions are the future. Preprint at https://doi.org/10.48550/arXiv.2006.11988 (2020).
38. Azizi, S. et al. Big self-supervised models advance medical image classification. In *Proc. International Conference on Computer Vision (ICCV)* 3458–3468 (IEEE, 2021).
39. Eslami, S., Meinel, C. & de Melo, G. PubMedCLIP: how much does CLIP benefit visual question answering in the medical domain? In *Proc. Findings of the Association for Computational Linguistics (EACL)* 1151–1163 (Association for Computational Linguistics, 2023).
40. Huang, S.-C. et al. Self-supervised learning for medical image classification: a systematic review and implementation guidelines. *npj Digit. Med.* **6**, 74 (2023).
41. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. E. A simple framework for contrastive learning of visual representations. In *ICML Proc. Machine Learning Research* Vol. 119, 1597–1607 (PMLR, 2020).
42. Zbontar, J., Jing, L., Misra, I., LeCun, Y. & Deny, S. Barlow twins: self-supervised learning via redundancy reduction. In *ICML Proc. Machine Learning Research* Vol. 139, 2310–12320 (PMLR, 2021).
43. Deng, J. et al. Imagenet: a large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 248–255 (IEEE, 2009).
44. Chen, T., Luo, C. & Li, L. Intriguing properties of contrastive losses. In *Advances in Neural Information Processing Systems, NeurIPS* Vol. 34 (eds Ranzato, M. et al.) 11834–11845 (Curran Associates, 2021).
45. Robinson, J. et al. Can contrastive learning avoid shortcut solutions? In *Advances in Neural Information Processing Systems, NeurIPS* Vol. 34 (eds Ranzato, M. et al.) 4974–4986 (Curran Associates, 2021).
46. Dippel, J., Vogler, S. & Höhne, J. Towards fine-grained visual representations by combining contrastive learning with image reconstruction and attention-weighted pooling. In *ICML Workshop: Self-Supervised Learning for Reasoning and Perception* (2021).
47. Li, T. et al. Addressing feature suppression in unsupervised visual representations. In *Proc. Winter Conference on Applications of Computer Vision (WACV)* 1411–1420 (IEEE, 2023).
48. Roth, K. et al. Towards total recall in industrial anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* 14298–14308 (IEEE, 2022).
49. Batzner, K., Heckler, L. & König, R. Efficientad: accurate visual anomaly detection at millisecond-level latencies. In *Proc. Winter Conference on Applications of Computer Vision (WACV)* 127–137 (IEEE, 2024).
50. Harmeling, S., Dornhege, G., Tax, D., Meinecke, F. & Müller, K.-R. From outliers to prototypes: ordering data. *Neurocomputing* **69**, 1608–1618 (2006).
51. Aggarwal, C. C. *Outlier Analysis* (Springer, 2013).
52. Chandola, V., Banerjee, A. & Kumar, V. Anomaly detection: a survey. *ACM Comput. Surv.* **41**, 15:1–15:58 (2009).
53. Parzen, E. On estimation of a probability density function and mode. *Ann. Math. Stat.* **33**, 1065–1076 (1962).
54. Kim, J. & Scott, C. D. Robust kernel density estimation. *J. Mach. Learn. Res.* **13**, 2529–2565 (2012).
55. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J. & Williamson, R. C. Estimating the support of a high-dimensional distribution. *Neural Comput.* **13**, 1443–1471 (2001).
56. Montavon, G., Kauffmann, J. R., Samek, W. & Müller, K.-R. in *Lecture Notes in Computer Science* Vol. 13200 (eds Holzinger, A. et al.) 117–138 (Springer, 2020).
57. Yu, Y., Qian, J. & Wu, Q. Visual saliency via multiscale analysis in frequency domain and its applications to ship detection in optical satellite images. *Front. Neurorobot.* **15**, 767299 (2022).
58. Parmar, G., Zhang, R. & Zhu, J. On aliased resizing and surprising subtleties in GAN evaluation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 11400–11410 (IEEE, 2022).

59. Kirichenko, P., Izmailov, P. & Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. In *Proc. International Conference on Learning Representations (ICLR)* (OpenReview.net, 2023).
60. Sugiyama, M., Krauledat, M. & Müller, K.-R. Covariate shift adaptation by importance weighted cross validation. *J. Mach. Learn. Res.* **8**, 985–1005 (2007).
61. Sugiyama, M. & Kawanabe, M. *Machine Learning in Non-stationary Environments: Introduction to Covariate Shift Adaptation* (MIT Press, 2012).
62. Iwasawa, Y. & Matsuo, Y. Test-time classifier adjustment module for model-agnostic domain generalization. In *Advances in Neural Information Processing Systems, NeurIPS* Vol. 34 (eds Ranzato, M. et al.) 2427–2440 (Curran Associates, 2021).
63. Esposito, C., Landrum, G. A., Schneider, N., Stiefl, N. & Riniker, S. Ghost: adjusting the decision threshold to handle imbalanced data in machine learning. *J. Chem. Inf. Model.* **61**, 2623–2640 (2021).
64. Niven, T. & Kao, H. Probing neural network comprehension of natural language arguments. In *Proc. Conference of the Association for Computational Linguistics* (eds Korhonen, A. et al.) 4658–4664 (Association for Computational Linguistics, 2019).
65. Heinzerling, B. NLP's Clever Hans moment has arrived. *J. Cogn. Sci.* **21**, 161–170 (2020).
66. Braun, M. L., Buhmann, J. M. & Müller, K.-R. On relevant dimensions in kernel feature spaces. *J. Mach. Learn. Res.* **9**, 1875–1908 (2008).
67. Basri, R. et al. Frequency bias in neural networks for input of non-uniform density. In *ICML Proc. Machine Learning Research* Vol. 119, 685–694 (PMLR, 2020).
68. Fridovich-Keil, S., Lopes, R. G. & Roelofs, R. Spectral bias in practice: the role of function frequency in generalization. In *Advances in Neural Information Processing Systems, NeurIPS* Vol. 35 (eds Koyejo, S. et al.) (Curran Associates, 2022).
69. Arras, L. et al. in *Lecture Notes in Computer Science* Vol. 11700 (eds Samek, W. et al.) 211–238 (Springer, 2019).
70. Schnake, T. et al. Higher-order explanations of graph neural networks via relevant walks. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7581–7596 (2022).
71. Ali, A. et al. XAI for transformers: better explanations through conservative propagation. In *ICML Proc. Machine Learning Research* Vol. 162, 435–451, (PMLR, 2022).
72. Jafari, F. R., Montavon, G., Muller, K. R. & Eberle, O. MambaLRP: explaining selective state space sequence models. In *Advances in Neural Information Processing Systems, NeurIPS* Vol. 37 (eds Globerson, A. et al.) 118540–118570 (Curran Associates, 2024).
73. Munir, M., Siddiqui, S. A., Dengel, A. & Ahmed, S. DeepAnT: a deep learning approach for unsupervised anomaly detection in time series. *IEEE Access* **7**, 1991–2005 (2019).
74. Devlin, J., Chang, M., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)* (eds Burstein, J. et al.) 4171–4186 (Association for Computational Linguistics, 2019).
75. Pelka, O., Koitka, S., Rückert, J., Nensa, F. & Friedrich, C. M. in *Lecture Notes in Computer Science* Vol. 11043 (eds Stoyanov, D. et al.) 180–189 (Springer, 2018).
76. PubMedCLIP *Hugging Face* https://huggingface.co/sarahESL/PubMedCLIP (2024).
77. openaiCLIP *GitHub* https://github.com/openai/CLIP (2024).
78. Facebook Research. barlowtwins *GitHub* https://github.com/facebookresearch/barlowtwins (2022).
79. CXR8 *NIHCC* https://nihcc.app.box.com/v/ChestXray-NIHCC (2017).
80. ieee8023 COVID-chestxray-dataset *GitHub* https://github.com/ieee8023/covid-chestxray-dataset (2020).
81. Pang, G., Shen, C., Cao, L. & van den Hengel, A. Deep learning for anomaly detection: a review. *ACM Comput. Surv.* **54**, 38:1–38:38 (2022).
82. Rippel, O., Mertens, P. & Merhof, D. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *ICPR* 6726–6733 (IEEE, 2020).
83. Jelinek, F., Mercer, R. L., Bahl, L. R. & Baker, J. K. Perplexity—a measure of the difficulty of speech recognition tasks. *J. Acoust. Soc. Am.* **62**, S63–S63 (2005).
84. Amazon Science. patchcore-inspection *GitHub* https://github.com/amazon-science/patchcore-inspection (2022).
85. Bradski, G. The OpenCV library. *Dr. Dobb's Journal of Software Tools* **120**, 122–125 (2000).
86. Torchvision: PyTorch's computer vision library https://pytorch.org/vision (2016).
87. Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K. & Müller, K.-R. *Explainable AI: Interpreting, Explaining And Visualizing Deep Learning* Vol. 11700 (Springer, 2019).
88. zennit. *GitHub* https://github.com/chr5tphr/zennit (2021).
89. Kauffmann, J. et al. Explainable AI reveals clever hans effects in unsupervised learning models: code. *Zenodo* https://doi.org/10.5281/zenodo.14186119 (2024).
90. Ml-workgroup. COVID-19 image repository. *GitHub* https://github.com/ml-workgroup/covid-19-image-repository (2020).

Acknowledgements

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) and the DeSBi Research Unit (KI-FOR 5363, project ID 459422098) and DFG KI-FOR 5363. Furthermore, K.-R.M. was partly supported by the Institute of Information and Communications Technology Planning and 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 S. Ganscha for the valuable comments on the manuscript.

Author contributions

Conceptualization and methodology: J.K., J.D., L.R., W.S., K.-R.M. and G.M. Experiments and software: J.K. and J.D. Analysis of results: J.K., J.D., L.R. and G.M. Supervision: W.S., K.-R.M. and G.M. Writing: J.K., J.D., L.R., W.S., K.-R.M. and G.M.

Funding

Open access funding provided by Technische Universität Berlin.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-025-01000-2.

Correspondence and requests for materials should be addressed to Klaus-Robert Müller or Grégoire Montavon.

Peer review information *Nature Machine Intelligence* thanks Haiyang Huang and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Grégoire Montavon
Klaus-Robert Müller

Last updated by author(s): Nov 27, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☒	☐	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	no software was used for data collection
Data analysis	https://github.com/jacobkauffmann/unsupervised-ch

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All data used in this paper, in particular the NIH CXR8, "COVID-19 image data collection", ImageNet, and MVTec-AD datasets, as well as the pre-trained models, are publicly available. The URLs for these datasets and models are given in the Methods section.

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	N/A
Reporting on race, ethnicity, or other socially relevant groupings	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Sample size was predetermined by dataset size.
Data exclusions	No exclusion was performed.
Replication	The code is available for reproduction and we ran experiments with multiple seeds.
Randomization	Data was randomly split between train/test when applicable.
Blinding	N/A

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Plants

Seed stocks	N/A
Novel plant genotypes	N/A
Authentication	N/A