DORA: Exploring outlier representations in Deep Neural Networks

Kirill Bykov[*,1], **Mayukh Deb , Dennis Grinwald**[1], **Klaus-Robert Müller**[4], **and Marina M.-C. Höhne**[*,1,2]
[1]ML Group, UMI Lab, TU Berlin
[2]UiT The Arctic University of Norway
[4]Google Research, Brain Team, ML Group TU Berlin, MPII, and Korea University

Abstract

Deep Neural Networks (DNNs) draw their power from the representations they learn. In recent years, however, researchers have found that DNNs, while being incredibly effective in learning complex abstractions, also tend to be infected with artifacts, such as biases, Clever Hanses (CH), or Backdoors, due to spurious correlations inherent in the training data. So far, existing methods for uncovering such artifactual and malicious behavior in trained models focus on finding artifacts in the input data, which requires both availabilities of a data set and human intervention. In this paper, we introduce DORA (Data-agnOstic Representation Analysis): the first automatic *data-agnostic* method for the detection of potentially infected representations in Deep Neural Networks. We further show that contaminated representations found by DORA can be used to detect infected samples in any given dataset. We qualitatively and quantitatively evaluate the performance of our proposed method in both, controlled toy scenarios, and in real-world settings, where we demonstrate the benefit of DORA in safety-critical applications.

Keywords — *Explainable AI, Deep Learning, Representation Analysis*

1 Introduction

Deep Neural Networks are able to learn complex representations and abstractions of data [4] as well as genuine architectures (e.g. [26]). While deep learning can successfully harvest vast amounts of data giving rise to ever-increasing performance gains, there is a dilemma. While more data helps to build better models, it becomes more and more unfeasible to inspect this overwhelming abundance of training data – which in consequence can lead to models that place focus on (unknown or undetected) spurious correlations in the training data (e.g. [35, 1]). This may heavily compromise their broad generalization and may thus ultimately lead to unrepresentative, unfair, or even unsafe rsp. harmful models.

With Machine Learning (ML) being universally deployed in the sciences and industry, a growing need to increase the transparency and safety of ML models has thus emerged [54, 31]. This has in the recent years lead to the field of *Explainable AI* (XAI) (e.g. [53]). Numerous different explanation approaches were introduced to illustrate the decision-making process of machines: either on a 'local' level, i.e., explaining the model's prediction for individual samples, or on a 'global' level, i.e., explaining the general discriminative concepts learned by the model across the whole data set. In particular, with the help of local explanation methods, undesirable behavior of trained networks could be found that were based on learned artifactual data or data containing spurious correlations [35].

[*]Corresponding authors: kirill.bykov@campus.tu-berlin.de, marina.hoehne@tu-berlin.de

While XAI has already achieved significant progress, e.g. in the area of including uncertainty information in the explanation [12], there are still several remaining challenges. First of all, so far, the analysis of the DL model is based on a given dataset, which may induce a *dataset bias* since only artifacts can be found that are incorporated in that very dataset. Second, the size of the dataset can be very large, thus, the computational costs scale with the *size* of the dataset. Furthermore, since the network successively increases the complexity and abstraction of the learned concepts from layer to layer [41], the concepts result finally in the last layers that are easier to understand for human users. However, understanding artifactual parts of the representation, that might have been learned in initial layers, is complicated through the non-availability of accessible, say, natural images of lower-level representations.

To overcome these disadvantages, we propose *DORA*[*] — the first *data-agnostic* framework for inspecting the representation space of Deep Neural Networks for infected neurons (i.e. neurons that are representing spurious or artifactual concepts). Independent of data, for any given DL model DORA, allows to automatically detect anomalous representations, that bear a high risk of learning unintended spurious concepts deviating from the desired decision-making policy. In addition, we show that infected representations, found by DORA, can also be used as artifact detectors when applied to any given dataset — allowing furthermore an automatic detection and successive cleaning of infected data points. The contributions of our work can be summarized as follows:

- Introduction of DORA: the first automatic *data-agnostic* method for finding infected and unintended representations in Deep Neural Networks.
- A thorough quantitative and qualitative evaluation of the proposed method, both in controlled simulations and for real-world use-cases.
- We present several interesting discoveries on artifactual representations in DL using DORA, e.g. an unintended (spurious) Chinese-character detector in representations from standard ImageNet trained networks.

2 Related Work

In order to address the concerns about the black-box nature of the complex learning machines [3, 60, 10, 54], the field of *Explainable AI (XAI)* has emerged.Some recent research focuses on self-explaining models [13, 20], also in the light of learned artifacts [21], where the ability of explanations is inherent in the network. However, by reducing the freedom of the network through the inherent explanatory structure, self-explaining networks suffer from reduced model accuracy, which is why most XAI methods (typically referred to as post-hoc explanation methods) are decoupled from the training procedure. Here, the model behavior can be either explained on a local level, where the decision-making strategy of a system is explained for one particular input sample, or on a global level, where the aim is to explain the prediction strategy learned by the machine and the purpose of its individual components in a universal fashion detached from single data points (similarly to feature selection [24]).

Local explanation methods are often given in form of attribution maps, interpreting the prediction by attributing relevance scores to the features of the input signal, highlighting the influential characteristics that affected the prediction the most. Various methods, such as as Layer-wise Relevance Propagation (LRP) [2], GradCAM [56], Occlusion [64], MFI [61], Integrated Gradient [59], have proven effective in explaining DNNs. To further boost the quality of interpretations, several enhancing techniques were introduced, such as SmoothGrad [57, 46], NoiseGrad [11], and Augmented GradCam [42]. However, while the local explanation paradigm is incredibly powerful in transferring the understanding of the decision-making strategies for a particular data sample, it lacks the ability to provide a general overview of the inner processes accomplished by the representation within a network. In addition, the need for a dataset causes a dependency of explanations on that specific data, which in turn limits the explanation abilities of a network to the available dataset. In contrast, *global* explanation methods aim to interpret the general behavior of learning machines by investigating

[*]PyTorch implementation of the proposed method could be found by the following link: `https://github.com/lapalap/dora`.

the role of particular components (e.g., neurons, channels, or output logits), which we refer to as representations.

One popular approach for the global explanation of complex learning machines is to interpret the function of the representations by producing input signals, e.g. images, that activate a certain representation the most. The motivation behind this method is based on research that studies representations in the brain [19, 62], mapping the stimuli that would activate a certain brain area the most. These signals, which we will refer to as Activation-Maximization Signals (AMS), could be either natural, found in a *data-aware* fashion by selecting a "real" example from the existing data corpus [7], or artificial, found in a *data-agnostic* mode by synthetically generating the input, via an optimization procedure. We would refer to them as n-AMSs (natural) and s-AMSs (synthetic), respectively.

Recently local XAI methods have shown the potential to reveal predictions that are made based on artifacts learned by the model, such as Clever Hans or Backdoor artifacts. One step toward a more universal understanding of the learned decision strategies aiming to detect undesired behavior of a model at scale was established by the Spectral Relevance analysis method — SpRAy — which was introduced to unmask the presence of a CH-behavior in the model [35]. SpRAy aims to perform a global explanation of the model, by analyzing local explanation over the dataset — after collecting local attribution maps, they are clustered by the Spectral Clustering algorithm, and the clusters are forwarded for manual inspection. SpRAy has been successfully applied in recent works [55], however, it involves a considerable amount of human supervision.Another potential shortcoming of SpRAy is that in the original method clustering is performed on the collection of saliency maps — thus, if artifacts are not consistent with shape and position in the original images, it might be hard to detect them easily.

The general field of outlier detection (OD) (cf. [51]) is the research area that studies the detection of anomalies and atypical observations through different methods and algorithms, where the majority of the OD methods are unsupervised. Popular OD methods include ABOD [33], LOF and Cluster-based LOF [8], Feature Bagging [36], HBOS [23], Isolation Forest [38], kNN, MCD[29], OCSVM [37], PCA [52]. Modern approaches employ Deep Learning for outlier detection in high-dimensional data: notable methods include Deep SVDD [49] and Deep SAD [50].

3 Method

In the following, we present *DORA* — Data-agnOstic Representation Analysis — the first framework intended for exploring the representation space of Deep Neural Networks regarding outlier neurons with the potentially undesired behavior. DORA is based on the idea of using the network itself for extracting semantic information contained in the s-AMS, and employing this information further to analyze the representation space and identify outlier representations. We hypothesize that distances between the s-AMS in the encoding space are correlated to semantic distances, thus we can find semantic outliers by finding outliers in the embedding space. Using intrinsic correlations between representations, contained in the encoding of s-AMS, we can reveal anomalous concepts that are promising candidates for being infected with unintended and potentially poisonous concepts.

Given a representation r within a network $f(x)$, there exist a subnetwork $g(x)$, that describes the activation of the representation r. We are now interested in the *signal* x, that maximally activates the representation in order to extract the concept, which is inherent in the representation. This signal, we refer to as s-AMS - synthetic activation maximization signal, which we obtain by solving the following optimization problem

$$\arg\max_{x \in \mathcal{C}} g(x), \tag{1}$$

where $\mathcal{C}$ is a regularized input domain, specific to a particular implementation of an AM method. Different regularization and extensions to the original method have been introduced to improve the clarity of the generated visual abstractions of learned concepts, such as [18, 43, 44]. In this work, we will use a Feature Visualisation (FV) method [45] as the main method for computing AM signals.

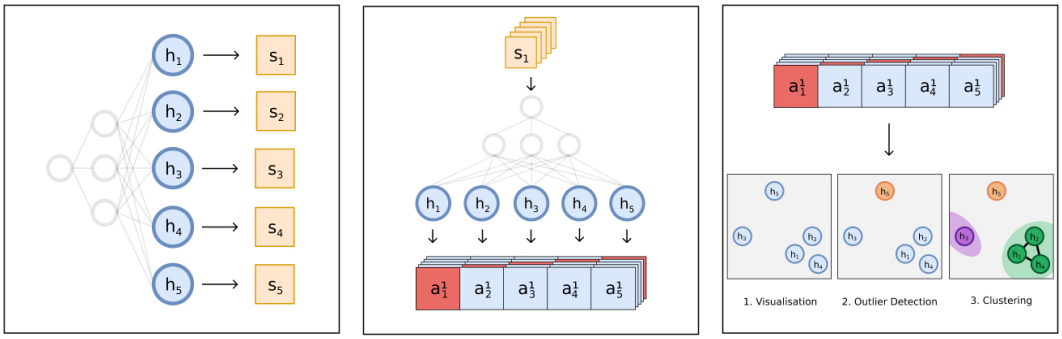

Figure 1: **DORA — Data-agnOstic Representation Analysis in three steps:** 1. Generation of s-AMS for a set of neurons (left), 2. Collecting the activations of the generated s-AMS of step 1 (middle), 3. Finding semantic outliers in the activation space (right).

DORA: Data-Agnostic Representation Analysis

Given a collection of K representations for analysis $\mathcal{H} = \{h_i, i \in [1, .., K]\}$, for example, neurons of a specific layer in the network, our proposed method could be summarised in three consecutive steps, that are visualized in Figure 1.

1. **Generation of synthetic Activation-Maximisation Signals** For each representation of the set $h_i \in \mathcal{H}$ we generate a synthetic Actvation Maximisation Signal – input signal that maximally activates this representation, e.g., computing a Feature Visualization for each representation. Thus, we obtain a set $\mathcal{S} = \{s_i, i \in [1, .., K]\}$, containing all s-AMS.
2. **Collection of AMS embeddings** In a second step, we feed the AMS $s \in \mathcal{S}$ computed in the previous step as inputs successively back into the network and collect their embeddings across the set $\mathcal{H}$ – same set of representations, signals were computed for, yielding a set of activations $\mathcal{A} = \{[a_1^i, a_2^i, ..., a_K^i], i \in [1, .., K]\}$.
3. **Representation Analysis** Finally, in a third step, we automatically explore the set of AMS embeddings with regard to any potential outlier representation, that might include unintended behavior, by applying Outlier Detection (OD) on top of the set of activation vectors $\mathcal{A}$.

Given a network, DORA can be used — in a data-agnostic manner — to locate neurons that are considered outliers in representation space and may have learned suspicious concepts. If the outlier representation indeed encodes a Clever Hans artifact, it can be further used in a non-data-agnostic manner to detect infected data points across a given dataset — after the forward pass of the network, the activation levels of infected representations could be used to score the degree of "infectedness". The final cleaned classification could be done, for example, by leaving out these top k-percent of the most-activated data points, where k is a pre-defined contamination fraction. Further, we show in our experiments, that activations of contaminated neurons that were detected, indeed have a high AUC-ROC measure for an artifact-vs-all binary-classification setting.

4 Experiments

In this section, we evaluate our proposed approach DORA both quantitatively for a controlled scenario with known ground truth as well as qualitatively in real-world experiments with SOTA architectures.

4.1 Synthetic Experiment

To showcase the high detection quality of DORA, we first generate a synthetic experiment, where the CH-behaviour of the model is encapsulated in a predetermined set of representations $\mathcal{I} \subset \mathcal{H}$ with

$|I| = 10$ in the average pooling layer of a ResNet18. Let X be the Animals10 dataset[†], which, after resizing and balancing, contains 1446 images of size $3 \times 224 \times 244$ per class. For the class "cat" we add a Clever Hans artifact – three yellow stripes on 15% of the images, illustrated in Figure 2, denoted as $X_{CH} \subset X$. To obtain a ground truth of infected representations we added two additional terms to the loss function, forcing the selected representations to react strongly to the infected data points, as well as forcing other representations not to be activated by infected images:

$$\mathcal{L}(X, y) = CE(f(X), y) + \alpha_1 \frac{1}{I} \sum_{i=1}^{I} CE(h_i(X), T) + \alpha_2 \|h_{\mathcal{H}\setminus\mathcal{I}}(X_{CH})\|_2,$$

where $CE(h_i(X), T)$ reflects the potential of neuron h_i to predict whether an images is infected or not, where the label information of the images is given by T. Furthermore, $||h_{\mathcal{H}\setminus\mathcal{I}}(X_{CH})||_2$ forces the activation of non-infected neurons to be small for infected images, and α_1 and α_2 are regularization parameters respectively. Intuitively, the first term is responsible for the classification performance, the second term forces the activations of predetermined infected representations to learn the artifact, and last term forces other representations not to be activated for infected images. More details about the training procedure can be found in the Appendix.

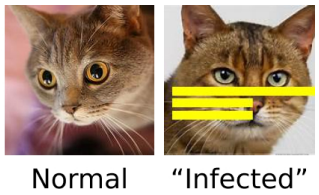

Figure 2: **Illustration of the CH-artifact placed on the images.** Left image represents a normal image from the Animals10 dataset, while the right image was manipulated by adding a set of yellow stripes as CH-feature.

From the projection in Figure 3 we can observe that infected representations are close to each other and grouped together. The computed s-AMS allow us to easily determine the CH behavior — the respective representations are activated by the yellow-stripe pattern in the data.

We further quantitatively evaluate the performance of different OD methods for finding outliers in representation space (see Results in Table 1). For this evaluation, we measure classification performance with both AUC and Precision @ N (P@N) and report computational time in seconds. All evaluations are averaged over 10 independent trials. From our quantitative experiment, we can observe that the PCA-based Outlier Detector performs best.

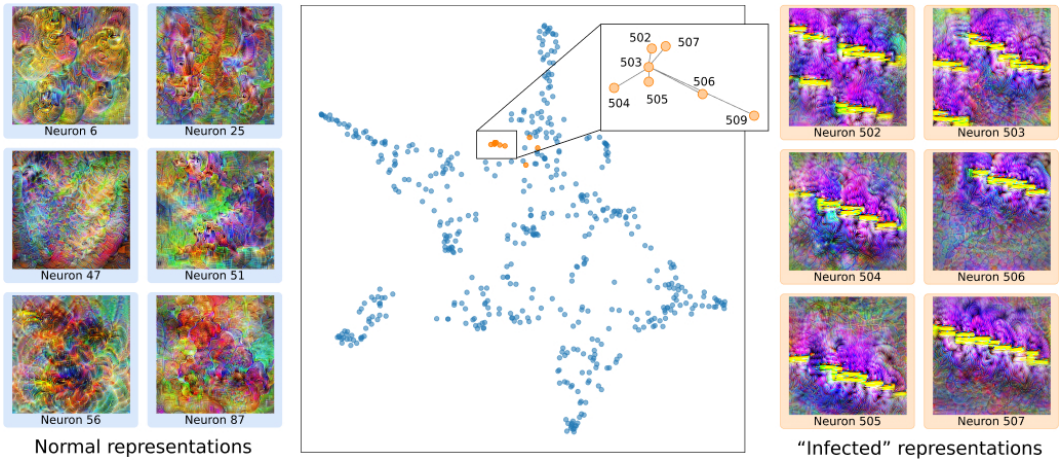

Figure 3: **Demonstration of DORA in a controlled environment.** The representation space of the s-AMS is illustrated via UMAP dimensionality reduction in the middle, where the set of outlier neurons found by DORA are highlighted in orange. We can observe yellow stripe patterns in the s-AMS of infected neurons shown on the left, while randomly selected representations of not outlier neurons which do not reflect the Clever Hans pattern in their s-AMS are shown on the right.

[†]Dataset could be found at `https://www.kaggle.com/alessiocorrado99/animals10`

Table 1: Results of ten different OD methods.

	ABOD	*CBLOF*	*FB*	*HBOS*	*IF*	*KNN*	*LOF*	*MCD*	*OCSVM*	*PCA*
AUC	0.74	0.93	0.94	0.67	0.51	0.83	0.93	0.73	0.93	**0.97**
P@N	0.5	**0.7**	**0.7**	0.2	0.0	**0.7**	**0.7**	0.5	**0.7**	**0.7**
Time	0.59	0.75	0.31	0.32	0.57	0.32	**0.03**	62.96	0.26	0.56

4.2 Finding Outliers in ImageNet Pre-trained models

Pre-trained models on ImageNet are broadly available and form a popular basis for transfer learning. Their representations are usually frozen during transfer learning in order to increase computational efficiency. Therefore users often only configure parameters for the layers on top of the "feature extractor" part of the CNN when using pre-trained models. In our experiment, we will study the representation space of standard pre-trained ResNet18[25] and DenseNet121[28] networks respectively (both easily accessible via two lines of code using the PyTorch framework).

4.2.1 ResNet18

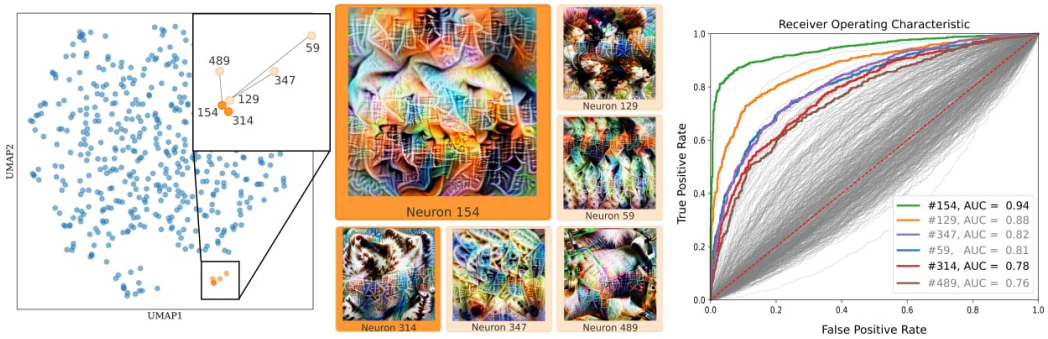

Figure 4: **Outlier Detection with DORA for ResNet18 pre-trained on ImageNet**. We investigate the last convolutional layer with DORA, consisting of 512 neurons, for which we compute the FV respectively (DORA step 1). On the left, the activation vectors of all 512 FV (DORA step 2) are plotted along with the first two UMAP components, where the highest outliers found by DORA (DORA step 3), neuron 154 and neuron 314 are marked in orange along with their closest neighbors, marked in light orange. The corresponding FVs (s-AMS) for each of the marked neurons are shown in the middle. From the FV of the outlier neuron 154 found by DORA, we can observe symbolic patterns, resembling Chinese symbols, represented by the neuron as well as by its closest neighbor neurons. Furthermore, on the right, each neuron is used as a detector for data points with added Chinese watermark symbols, where the performances are reported as ROC curves. We can observe that the outlier neuron 154 exhibits the highest AUC value (green curve), followed by its nearest neighbors.

We apply DORA to analyze the Average Polling layer – the last 512 high-level representations of the "feature extractor" — which are commonly used without further modification during transfer learning.

Following the first two steps of DORA, we compute synthetic Activation-Maximization Signals of all 512 neurons, which we then feed again into the network and collect for each s-AMS its 512-dimensional activation vector. Afterward, in the third step of DORA, we applied 10 OD methods, described previously, with the outlier fraction parameter set to 0.05. DORA is able to identify three outlier neurons, namely neurons 154, 314, and 168, that 8 of 10 methods labeled as an outlier. The same experiment with an outlier fraction set to 0.01 similarly showed that again 8 out of 10 OD methods agree on neurons 154 and 168 being outliers. All s-AMS of the outlier neurons can be found in the Appendix. Here, we will more closely inspect the neurons 154 and 314. The signal of the strongest outlier neuron 154 is shown in Figure 4 in the middle panel, framed in orange. In addition to different colors, we can observe patterns that include Chinese characters. Notably, there exists no specific ImageNet class for this type of representation, i.e. we are likely to observe a potential Clever

Hans artifact, that may lead to a degrading classifier performance (see [1]).

In order to explore whether other neurons may represent similar symbolic concepts, we investigate the neighborhood of the outlier representation 154, by analyzing the Euclidean distance in the space of the activation vectors as shown in Figure 4 on the left side, where the activation vectors of the Feature Visualizations of all 512 neurons are projected to a two-dimensional space using the first two UMAP [40] components. When inspecting the six closest neighbors of neuron 154, shown in the middle of Figure 4, we observe that they similarly exhibit a Clever Hans bias towards representing Chinese characters.

DORA identifies generic outlier representations, without having to resort to any specific data set. The detected outlier neurons are candidates for embodying artifacts that have been learned by the model and can severely compromise generalization. We now demonstrate, how the information gained by DORA in a data-independent manner, can be further used to detect candidates for data points that exhibit outlier behavior. Therefore, we create a dataset where a Chinese symbol watermark is added to half of the data points, hoping to be able to detect these manipulated images by neuron 154. The dataset created, contains 998 images, where each image belongs to a different ImageNet class [‡]. As indicated above we add a Chinese watermark randomly placed in the image 499 randomly selected images. The watermark was generated by randomly sampling seven out of the 20 most used Chinese characters [14]. The results are shown in Figure 4 on the right panel, where the performance of all neurons for detecting the data points including the Chinese watermark are reported as ROC curves. We can clearly observe that the outlier neuron 154 found by DORA, marked in green, performs best in this detection task with an AUC value of 0.94, followed by its nearest neighbors. Hence, outliers in the representation neurons found by DORA can indeed be used as an outlier detector for a given data set, e.g. for potentially removing data points with the undesired behavior.

4.2.2 Clever Hans representations survive transfer learning

Regarding the high popularity of pre-trained models in safety-critical areas, it is mandatory that the artifacts embodied in a pre-trained model are being made ineffective or are unlearned during the transfer learning task (see also [1]). Therefore, we will use DORA to find outliers in a pre-trained network — here DenseNet121, and further, analyze whether they survive transfer learning. For this purpose we inspect a state-of-the-art finetuned model on the CheXpert challenge [30], which benchmarks classifiers on a 5-class multilabel data set of chest radiographs and which is based on a DenseNet121 pre-trained on ImageNet. For detailed information about the data set and training procedure, we refer to the Appendix.

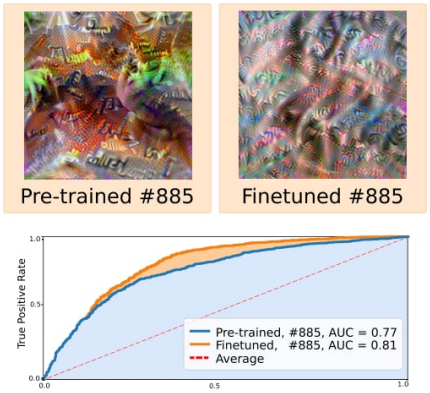

Figure 5: **Survived latin text detector**. Neuron 885 learns to detect Latin text during pre-training (top left), and does not unlearn this behavior after finetuning on the CheXpert data set (top right). The AUC values of the neurons' activation on images corrupted with Latin watermarks are high after pre-training and even increases after finetuning (bottom).

First, applying DORA to the pre-trained DenseNet121, we found several outlier neurons, among others, neuron 768, which detects Chinese Characters (similar to in ResNet18 neuron 154), and neuron 885, detecting Latin text, where the latter is shown in Figure 5 in the upper left panel (see Appendix for further experimental details and results). Note that, to the best of our knowledge, this is the first time that CH effects are reported for the DenseNet121 architecture.

Second, we apply DORA to the DenseNet121 finetuned on the CheXpert dataset — note that all layers are finetuned until convergence on the data. The outliers found by DORA on the pre-trained model and the finetuned model intersect — both neurons, i.e., 768, the Chinese symbol detector, and 885, the Latin text detector are still outliers and, most interestingly maintained their original semantic outlier

[‡]Images were taken from https://github.com/EliSchwartz/imagenet-sample-images, where two images (class "carton" and "terrapin") have been excluded from the dataset since they already exhibit Chinese watermarks.

information, i.e., exhibit Chinese and Latin Character concepts, as can be seen for neuron 885 on the upper right in Figure 5.

Furthermore, similar to the ResNet18 experiment discussed above, we study the ability of neuron 885 for being an efficient Latin text detector and report the AUC values in Figure 5 for both, the neuron in the pre-trained (blue curve, AUC value of 0.77) and in the finetuned model (orange curve, AUC value of 0.81). Note that – similarly to the experiment above – the underlying dataset was generated manually by adding to half of the images Latin text watermarks on randomly chosen positions in the image. From the results shown in Figure 5, we can observe that the neuron 885 indeed exhibits *persisting* strong activations for images with Latin text. The Clever Hans effect is even reinforced after finetuning, which might be due to small Latin text characters in the CheXpert dataset, e.g. R and L indicating left and right.

4.3 CLIP ResNet50

In the following, we apply DORA to "layer4" of the CLIP ResNet 50 model [48], trained in a Self-Supervised manner on a dataset unreleased to the public. As in previous experiments, we use the 10 OD methods, setting the outlier fraction parameter set to 0.05. Figure 6 shows the top-20 outlier representations on which the majority of OD methods agree, all highlighted in orange (see also appendix for further details). Most notably, DORA was able to find representations containing different conceptual information, which, by means of analyzing both s-AMS and n-AMS[§], we label as pornography (95), drugs (946), World War 2 (135), army (242), and obesity (22) neurons. Moreover, we were able to find a cluster, consisting of the neighbors of neuron 95, clearly sharing similar concepts. A list of all outlier neurons is given in the Appendix.

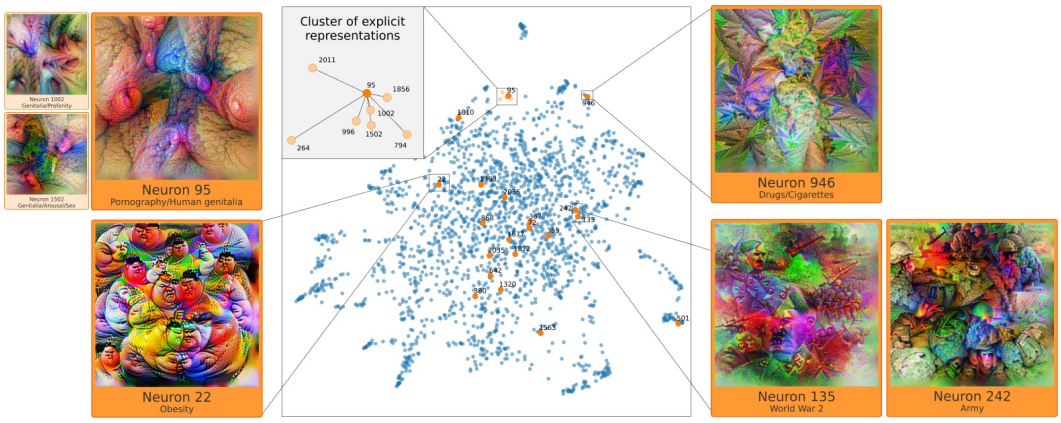

Figure 6: **DORA results on CLIP model.** Outlier neurons found by DORA are marked in orange, of which the Feature Visualizations are shown for five diverse concepts: pornography (95, and its closest neighbors 1002, 1502 that learned genitalia concepts), drugs (946), World War 2 (135), army (242), and obesity (22).

5 Understanding DORA outliers

In a number of neural network architectures analyzed above, we could observe that semantically close abstractions are also highly correlated in terms of common activations of the representing neurons. Intuitively, this can be explained by sharing low-level representations between similar classes, which is why e.g. given other dog classes receive higher activation than far away classes such as vehicles or electronics. This resonates well with findings in Neuroscience hypothesizing correlations and inter-connections of high-level representations in the human brain [34, 6, 47]. In recent years, also

[§]Both s-AMS and n-AMS for "layer4" representations could be found by the following link `https://microscope.openai.cm/models/contrastive_rn50/image_block_4_2_Add_6_0?models.op.feature_vis.type=channel&models.op.technique=feature_vis`

the relationship between visual and semantic similarities in Computer Vision was studied and a strong linkage between semantic and visual similarity has been shown Brust and Denzler [9]. In Deselaers and Ferrari [17], the authors confirm the assumption that visual similarity between categories grows with semantic similarity. In the following, we demonstrate that this also holds for the similarity structure between n-AMS and s-AMS, specifically, *semantically dissimilar concepts have visually dissimilar n-AMSs*. Note that we will only present the main findings for the relationship between n-AMS and s-AMS of the output neurons of the ResNet18 network, pre-trained on ImageNet (for further details and experiments see appendix).

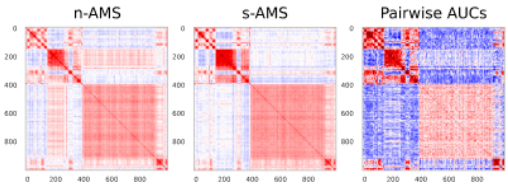

Figure 7: **Similarity between distance matrices.** We illustrate 3 distance matrices: cosine distance matrix between embeddings of n-AMS (left) and s-AMS (right); pairwise AUC-ROC for the binary detection of neurons top activating images (right).

In Figure 7 we observe a similar correlation pattern of the similarities across the cosine distance matrices computed between the embeddings of n-AMS (left) and s-AMS (middle). Furthermore, on the right panel, the pairwise AUC values are plotted, indicating the ability of other neurons (columns) in predicting the concepts of a specific neuron (rows) – computed by measuring the AUC-ROC classification performance of the individual neuron in detecting the top activating images for other neurons. Additionally, we demonstrate that the availability of real images is not necessary for the analysis of correlations between representations, as synthetic stimuli are equally informative. We could demonstrate this for the ImageNet classes, where we can conserve the semantic similarity given by natural images with the AM-signals for all ImageNet classes (see Appendix).

These findings reinforce our hypothesis about the existence of a strong relationship between s-AMS embedding distances and the semantic distances, i.e. representations, being semantically close in terms of class similarity, are also found close in terms of distance between the activation vectors of their AM signals. In the practical experiments, we have observed that this hypothesis holds – outliers, found in the embedding space indeed relate to the semantic outliers for the specific tasks, while s-AMS of conceptually neighboring representations are also neighbors in the embedding space.

6 Discussion and Conclusion

Representations from deep learning architectures embody the essence of the learned data. Since it is not uncommon that data sets to contain artifacts, spurious correlations, or biases, it becomes essential to inspect the deep learning model by XAI to avoid undesired or even harmful behavior of the model. So far, this was mainly done relying on a specific dataset, i.e., local XAI methods are used to locate artifacts within representations and data.

Figure 8: **N-AMSs failing in recognizing a concept.** Comparison of the s-AMSs and n-AMSs from ImageNet dataset for the unit 744 in the last convolutional layer of the CLIP ResNet50 model. Due to the inaccessibility of the training dataset, n-AMSs struggle to illustrate the concept of the "Star Wars" neuron. Illustrated signals were obtained from OpenAI Microscope.

The strength of our proposed method DORA is given by its simplicity and *data-independent nature* — any trained neural network can be inspected with DORA without the use of any dataset, such that the failure in finding the correct concepts with n-AMS is reduced as shown in Figure 8 in case of the Star Wars neuron. Hence, DORA is the first automatic data-agnostic method for finding infected representations in Deep Neural Networks. Interestingly, we could demonstrate that infected representations typically manifest themselves as outliers in representation space. Note, that the outlier representations found by DORA can be used as Clever-Hans detectors to analyze any data set with regard to the artifact concept found as an outlier. Furthermore, we have shown that such outlier representations can even persist when doing transfer learning. This adds to the raising awareness that the use of pre-trained models in safety-critical areas, even after fine-tuning on a new dataset, may contain representations that still exhibit undesired behavior (see also [58]).

While we showed DORA's broad applicability, we would like to address some remaining challenges in the following: the main limitation is related to the assumption that malicious or CH-behavior of the representations is not systemic. In other words, DORA would not be able to find infected representations if this behavior is shared across a large proportion of representational space. Another limitation is that DORA generates one s-AMS per representation, thereby limiting the expression of concepts that are multi-faceted or contain a latent structure. We will take these considerations towards future work.

In summary, we showed the functionality, high quality, and usefulness of DORA for finding artifactual aspects in representation space; this holds both for controlled environments as well as real-world scenarios. Note that, although we have introduced DORA as an automatic tool, if necessary, the final decision on the degree of harmfulness of any outlier representations could be subjected to human scrutiny. In this sense, DORA substantially facilitates human intervention and reduces it to a minimum, however, for applications of importance human supervision will still be necessary. In future work, we will apply the proposed solution broadly in the sciences, medicine, and other technical domains, such as NLP, where discovering artifacts and biases in the representations is of great value.

Acknowledgements

We would like to thank Filip Rejmus for his analysis regarding the visualization of the CH behavior in representations with global explanation methods. Furthermore, we would like to thank Sebastian Lapuschkin for fruitful discussions about CH-behavior in Deep Neural Networks. This work was partly funded by the German Ministry for Education and Research through the third-party funding project Explaining 4.0 (ref. 01IS20055). KRM acknowledges support from the Federal Ministry of Education and Research (BMBF) for BIFOLD (01IS18037A). KRM was also partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korean 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) and by the German Ministry for Education and Research (BMBF) under Grants 01IS14013B-E, 01GQ1115.

References

[1] C. J. Anders, L. Weber, D. Neumann, W. Samek, K.-R. Müller, and S. Lapuschkin. Finding and removing clever hans: Using explanation methods to debug and improve deep models. *Information Fusion*, 77: 261–295, 2022.

[2] S. Bach, A. Binder, G. on, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

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

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[5] E. Bisong. *Google Colaboratory*, pages 59–64. Apress, Berkeley, CA, 2019. ISBN 978-1-4842-4470-8. doi: 10.1007/978-1-4842-4470-8_7. URL `https://doi.org/10.1007/978-1-4842-4470-8_7`.

[6] M. F. Bonner and R. A. Epstein. Object representations in the human brain reflect the co-occurrence statistics of vision and language. *Nature Communications*, 12(1):1–16, 2021.

[7] J. Borowski, R. S. Zimmermann, J. Schepers, R. Geirhos, T. S. Wallis, M. Bethge, and W. Brendel. Natural images are more informative for interpreting cnn activations than state-of-the-art synthetic feature visualizations. In *NeurIPS 2020 Workshop SVRHM*, 2020.

[8] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[9] C.-A. Brust and J. Denzler. Not just a matter of semantics: The relationship between visual and semantic similarity. In *German Conference on Pattern Recognition*, pages 414–427. Springer, 2019.

[10] V. Buhrmester, D. Münch, and M. Arens. Analysis of explainers of black box deep neural networks for computer vision: A survey. *arXiv preprint arXiv:1911.12116*, 2019.

[11] K. Bykov, A. Hedström, S. Nakajima, and M. M.-C. Höhne. Noisegrad: enhancing explanations by introducing stochasticity to model weights. *arXiv preprint arXiv:2106.10185*, 2021.

[12] K. Bykov, M. M.-C. Höhne, A. Creosteanu, K.-R. Müller, F. Klauschen, S. Nakajima, and M. Kloft. Explaining bayesian neural networks. *arXiv preprint arXiv:2108.10346*, 2021.

[13] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin. This looks like that: deep learning for interpretable image recognition. *arXiv preprint arXiv:1806.10574*, 2018.

[14] J. Da. A corpus-based study of character and bigram frequencies in chinese e-texts and its implications for chinese language instruction. In *Proceedings of the fourth international conference on new technologies in teaching and learning Chinese*, pages 501–511. Citeseer, 2004.

[15] M. Deb. Feature visualization library for pytorch. `https://github.com/Mayukhdeb/torch-dreams`, 2021.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[17] T. Deselaers and V. Ferrari. Visual and semantic similarity in imagenet. In *CVPR 2011*, pages 1777–1784. IEEE, 2011.

[18] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[19] I. Fujita, K. Tanaka, M. Ito, and K. Cheng. Columns for visual features of objects in monkey inferotemporal cortex. *Nature*, 360(6402):343–346, 1992.

[20] S. Gautam, M. M.-C. Höhne, S. Hansen, R. Jenssen, and M. Kampffmeyer. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *arXiv preprint arXiv:2108.12204*, 2021.

[21] S. Gautam, M. M.-C. Höhne, S. Hansen, R. Jenssen, and M. Kampffmeyer. Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, 2022. doi: 10.1109/ISBI52829.2022.9761651.

[22] G. Goh, N. Cammarata, C. Voss, S. Carter, M. Petrov, L. Schubert, A. Radford, and C. Olah. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30, 2021.

[23] M. Goldstein and A. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, pages 59–63, 2012.

[24] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[26] X. He, K. Zhao, and X. Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212: 106622, 2021.

[27] G. Huang, Z. Liu, L. van der Maaten, and K. Weinberger. Densely connected convolutional networks. 07 2017. doi: 10.1109/CVPR.2017.243.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[29] M. Hubert and M. Debruyne. Minimum covariance determinant. *Wiley interdisciplinary reviews: Computational statistics*, 2(1):36–43, 2010.

[30] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. Mong, S. Halabi, J. Sandberg, R. Jones, D. Larson, C. Langlotz, B. Patel, M. Lungren, and A. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:590–597, 07 2019. doi: 10.1609/aaai.v33i01.3301590.

[31] J. Jiménez-Luna, F. Grisoni, and G. Schneider. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584, 2020.

[32] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452, 2008.

[34] N. Kriegeskorte, M. Mur, and P. A. Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4, 2008.

[35] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[36] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166, 2005.

[37] K.-L. Li, H.-K. Huang, S.-F. Tian, and W. Xu. Improving one-class svm for anomaly detection. In *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*, volume 5, pages 3077–3081. IEEE, 2003.

[38] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[39] S. Marcel and Y. Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488, 2010.

[40] L. McInnes, J. Healy, N. Saul, and L. Grossberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3:861, 09 2018. doi: 10.21105/joss.00861.

[41] G. Montavon, M. L. Braun, and K.-R. Müller. Kernel analysis of deep networks. *Journal of Machine Learning Research*, 12(78):2563–2581, 2011.

[42] P. Morbidelli, D. Carrera, B. Rossi, P. Fragneto, and G. Boracchi. Augmented grad-cam: Heat-maps super resolution through augmentation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4067–4071, 2020. doi: 10.1109/ICASSP40776.2020.9054416.

[43] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in neural information processing systems*, pages 3387–3395, 2016.

[44] A. Nguyen, J. Yosinski, and J. Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

[45] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[46] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*, 2019.

[47] K. Patterson, P. J. Nestor, and T. T. Rogers. Where do you know what you know? the representation of semantic knowledge in the human brain. *Nature reviews neuroscience*, 8(12):976–987, 2007.

[48] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[49] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4393–4402, 2018.

[50] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkgH0TEYwH.

[51] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5): 756–795, 2021.

[52] B. N. Saha, N. Ray, and H. Zhang. Snake validation: A pca-based outlier detection method. *IEEE signal processing letters*, 16(6):549–552, 2009.

[53] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

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

[55] P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H.-G. Luigs, A.-K. Mahlein, and K. Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

[56] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 10 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://dx.doi.org/10.1007/s11263-019-01228-7.

[57] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[58] V. Srinivasan, N. Strodthoff, J. Ma, A. Binder, K.-R. Müller, and W. Samek. On the robustness of pretraining and self-supervision for a deep learning-based analysis of diabetic retinopathy. *arXiv preprint arXiv:2106.13497*, 2021.

[59] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[60] M. M.-C. Vidovic, N. Görnitz, K.-R. Müller, G. Rätsch, and M. Kloft. Opening the black box: Revealing interpretable sequence motifs in kernel-based learning algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 137–153. Springer, 2015.

[61] M. M.-C. Vidovic, N. Görnitz, K.-R. Müller, and M. Kloft. Feature importance measure for non-linear learning algorithms. *arXiv preprint arXiv:1611.07567*, 2016.

[62] Y. Yamane, E. T. Carlson, K. C. Bowman, Z. Wang, and C. E. Connor. A neural code for three-dimensional object shape in macaque inferotemporal cortex. *Nature neuroscience*, 11(11):1352–1360, 2008.

[63] Z. Yuan, Y. Yan, M. Sonka, and T. Yang. Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. pages 3020–3029, 10 2021. doi: 10.1109/ICCV48922.2021.00303.

[64] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[65] Y. Zhao, Z. Nasrullah, and Z. Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019. URL http://jmlr.org/papers/v20/19-011.html.

A Parameters of Outlier Detection methods

As described in the paper, for the outlier detection (OD) we used a collection of 10 OD methods from the PyOD library [65], namely ABOD, CBLOF, FB, HBOS, IF, KNN, LOF, MCD, OCSVM, and PCA, with the outlier fraction parameter, is set to 0.05 [¶].

B Synthetic Experiment

For the synthetic experiment, we took a pre-trained ResNet18 [25] and fine-tuned it on the Animals10 dataset, described at Section 4. A set of $I = 10$ predetermined representations, namely neurons from 502 to 511, were selected to encapsulate the artifactual behavior.

We use the loss function, described in Section 4, with $\alpha_1 = 1$ and $\alpha_2 = 0.0001$. We use the Adam optimiser [32] with a learning rate of 0.00025 and an exponential decreasing parameter $\gamma = 0.5$. The trained network achieves 95% accuracy on the main 10-class classification task, while the second term of the loss forced a predetermined set of representations to learn the artifact and the third loss deactivates all other representations if an image contains the artifact.

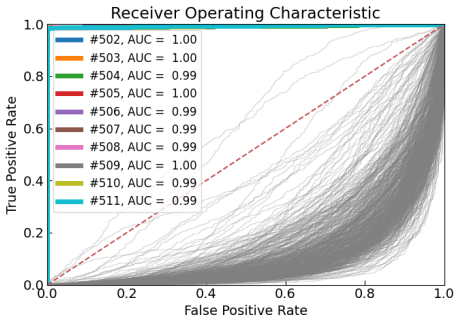

Figure 9: **Comparison of the AUC-ROC performance of the representations for detectting infected images.** The ROC curves for all 512 representations of the average pooling layer are shown. In gray, ROC curves are illustrated for the non-infected neurons, that were forced to be deactivated by the "infected" images, while the predetermined set of 10 neurons with their AUC-ROC performance is illustrated by the colored curves, which, are superimposed by each other regarding to their similar performance, shown also by their AUC values.

Figure 9 illustrates the performance of the representations for the binary artifact-vs-all detection task. For this, we have used 1000 images of different classes from the ImageNet dataset, and for each image, we created a copy with the artifact added. The ROC curves for each neuron in discriminating between artifact and non-artifact images are shown in Figure 9. As intended, "infected" representations perform best, as evidenced by the ROC curves having an AUC value between [0.99,1] (note that only the cyan ROC curve is visible since all curves have a similar performance). The other, non-infected representations, that were forced to be deactivated by the artifacts are visualized in gray and we can observe that they perform worse in discriminating between artifact and non-artifact images.

The s-AMS for the "infected" set of pre-determined representations could be observed in Figure 10. As expected, we can observe the distinct pattern of yellow stripes in the Feature Visualizations – however, we can also observe that for several representations, namely neurons 508, 510, and 511, this pattern could not be observed. This highlights the potential issue with the non-convergence of the optimization procedure. In Figure 3, we can observe that these neurons are further away from the main cluster of "infected" representations.

C ResNet18

For this experimet we used ResNet18 [25] pretrained on ImageNet [16], downloaded from the Torchvision library [39]. The s-AMS for the representations were computed with the torch-dreams library [15], using the default parameters.

From Figure 11 we can observe the concordance of the OD methods. Both neurons 154 and 314, i.e., the Chinese text detectors, have the highest degree (8/10 OD methods) of being outliers. The n-AMS

[¶]Parameters for the methods could be found by the following link `https://github.com/yzhao062/pyod/blob/master/notebooks/benchmark.py`

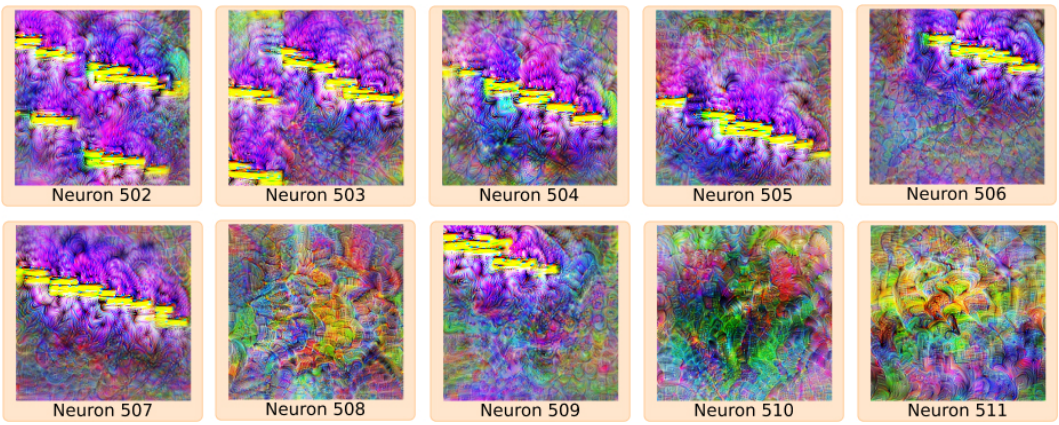

Figure 10: **s-AMS for "infected" representations**. Illustration of the s-AMS signals for the predefined set of "infected" representations. We can observe, that neurons, for which yellow stripes are less visible (508, 510, 511) in their Feature Visualizations, are further away from, representations, that have visible yellow stripes, which can be seen in Figure 3 based on the distances between their embeddings.

for those neurons, as well as their direct neighbors, reported in the Figure 4, are shown in Figure 13 – we can observe that indeed that the top activating natural signals for these neurons contain Chinese watermarks.

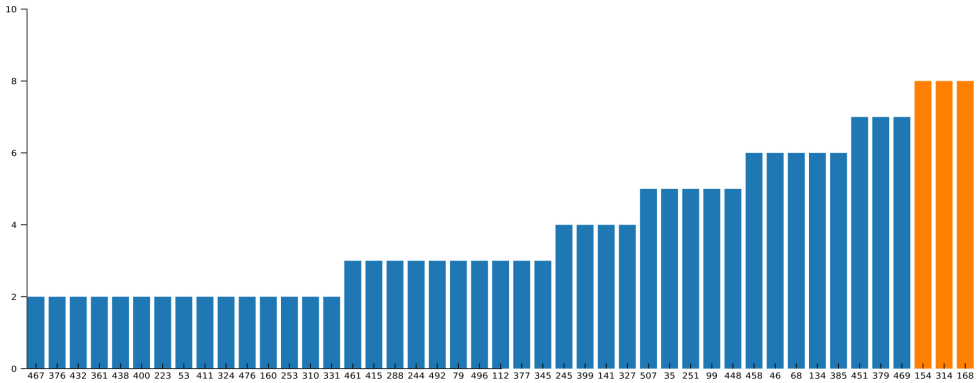

Figure 11: **Concordance between OD detection methods for the outliers in the Average Pooling layer of ResNet18**. Illustration of how many OD methods, out of 10 methods, agree on a representation being an outlier. We observe that neurons 154 and neurons 314 have the highest agreement among the OD methods.

The neuron 168, which is also considered an outlier by 8 methods, corresponds to the concept of "fountain/water splash", as could be comprehended for the n-AMS and s-AMS for this representation, illustrated in the Figure 12. We hypothesize that the reason for this is the fact that the learned concept operates in a different manner compared to the majority of representations.

Figure 12: **s-AMS and n-AMS for the ResNet18, neuron 168**. Illustration of the s-AMS (left) and 15 n-AMS (right) for the specified neuron 168, collected from the ImageNet Fall11 dataset.

(a) Neuron 154

(b) Neuron 314

(c) Neuron 129

(d) Neuron 347

(e) Neuron 59

(f) Neuron 489

Figure 13: **n-AMS for different ResNet18 neurons.** Illustration of the 15 n-AMS signals for the different neurons in "avgpool" layer of ResNet18 network, found to de in the Chinese-watermark-detector cluster, namely neurons a) 154, b) 314, c) 129, d) 347, e) 59 and f) 489.

D DenseNet 121

D.1 DORA on pre-trained DenseNet 121

In the following, we apply DORA to DenseNet121, particularly, to the last layer of the feature extractor, consisting of 1024 representations. We focus on two outliers found by DORA, neuron 768 and neuron 885, and some of its closest neighbors. We identify, the first outlier as Chinese and the latter as Latin text character detectors by inspecting at the feature visualizations and AUC-ROC values in Figure 14 and Figure 15. The AUC-ROC values were computed on an ImageNet subset with both uncorrupted and corrupted images, i.e., were Chinese and Latin text watermarks were added to the images for corruption, respectively. The high AUC-ROC values for detecting images with Chinese or Latin text characters of the corresponding channels underline the assumption that these channels learned Chinese and Latin text character concepts. In Figure 16 we show the fifteen highest activating n-AMS for some of the found outliers.

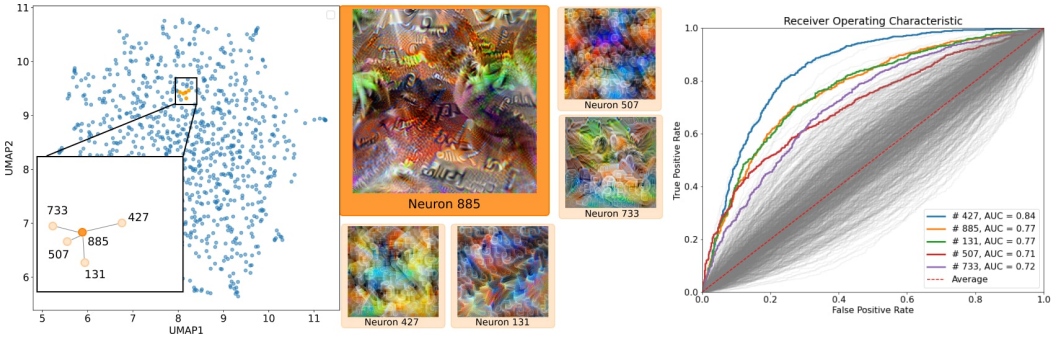

Figure 14: **DenseNet121 — Latin text detector.** Applying DORA to the last layer of the feature extractor of DenseNet121 yields, among others, Neuron 885, which is visualized in the middle, framed in orange. From neuron 885 as well as from its five closest neighbors (shown left), we can observe semantic concepts resembling Latin text characters. The AUC values were computed using the channel activations on a data set that was corrupted with Latin text watermarks. As shown, the AUCs are high for the representation outliers found by DORA, compared to most of the other representations, which indicates that they indeed learned to detect Latin text.

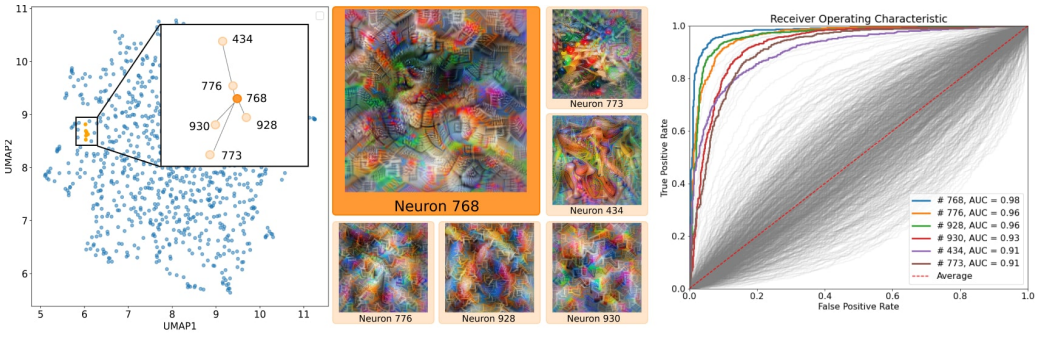

Figure 15: **DenseNet121 — Chinese text detector.** Applying DORA to the last layer of the feature extractor of DenseNet121 yields, among others, Neuron 768, which is visualized in the middle, framed in orange. From Neuron 768 as well as from its five closest neighbors (shown left), we can observe semantic concepts resembling Chinese text characters. The AUC values were computed using the channel activations on a data set that was corrupted with Chinese text watermarks. As shown, the AUCs are high for the representation outliers found by DORA, compared to most of the other representations, which indicates that they indeed learned to detect Chinese text characters.

(a) Neuron 885

(b) Neuron 768

(c) Neuron 427

(d) Neuron 434

Figure 16: **n-AMS for different DenseNet121 neurons.** Illustration of the 15 n-AMS signals for the different neurons in "features" layer of DenseNet121 network, found to de in the Chinese-watermark-detector cluster, namely neurons a) 885, b) 768, c) 427, and d) 434.

D.2 DORA on finetuned DenseNet 121

As mentioned in Section 4.4.2, we find that the outliers found by DORA are maintained during fine-tuning on another dataset, e.g. the CheXpert challenge. The CheXpert challenge benchmarks various deep learning models on the task of classifying multilabel chest radiographs and additionally provides human experts, e.g. radiologists, with performance metrics for comparison. The data set itself consists of 224,316 training, 200 validation, and 500 test data points. The current best approach in terms of AUC-ROC score uses an ensemble of five DenseNet121's [27] that were pre-trained on the ImageNet dataset and fine-tuned by optimizing a special surrogate loss for the AUC-ROC score [63]. The training code can be found in this public repository `https://github.com/Optimization-AI/LibAUC/`. We choose to fine-tune one DenseNet121 using this approach on a down-sampled version of the CheXpert data with a resolution of 256x256x3. The converged model yields an AUC-ROC score of 87.93% on the validation dataset. Having the finetuned DenseNet121 and the outlier neuron 768 at hand we show the Feature Visualizations and the AUC-ROC curves for both the pre-trained and fine-tuned channel on an ImageNet subset with both uncorrupted and corrupted images with Chinese text watermarks in Figure 17.

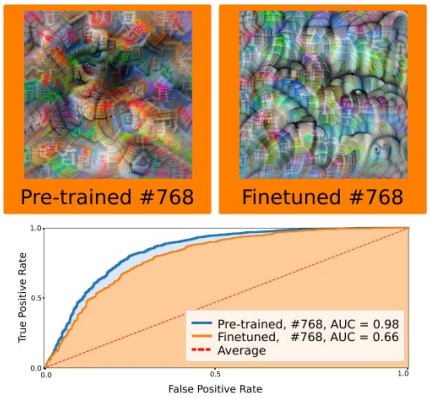

Figure 17: **Survived Chinese text detector**. Neuron 768 learns to detect Chinese symbols during pre-training (top left), and does not unlearn this behavior during fine-tuning on the CheXpert dataset (top right). The AUC values of the neurons' activation on images corrupted with Chinese watermarks are still high after pre-training.

E CLIP ResNet 50

For computing the s-AMS for the CLIP ResNet 50 we employed the same parameters[‖] as in Goh et al. [22] using the Lucent library[**], setting the number of optimization steps to 512.

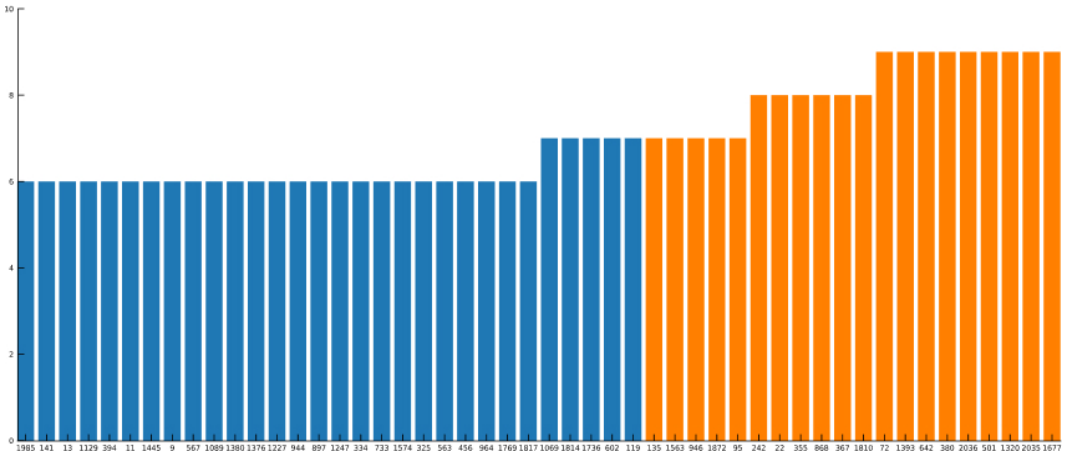

Figure 18: **Concordance between OD detection methods for the outliers in the "layer4" of CLIP ResNet 50** Illustration of how many OD methods, out of 10 methods, agree on a representation being an outlier. In orange we highlighted the top-20 outliers by the agreement between OD methods.

Using the OD methods in the same fashion, described in previous experiments, in Figure 6 we have reported top-20 neurons measured by the agreement between OD methods. Figure 18 illustrates other outliers with high concordance score.

F Comparing computational costs for n-AMS and s-AMS

In order to find n-AMS samples, we had to iterate over the Imagenet Fall11 dataset (1281167 images) and collect the activations from the layers `model.layer4`, `model.features` and `model.visual` respectively from ResNet18, Densenet121 and the CLIP image encoder. We calculated the time taken to iterate over the whole dataset with a batch size of 50 on an Nvidia Tesla T4 GPU.

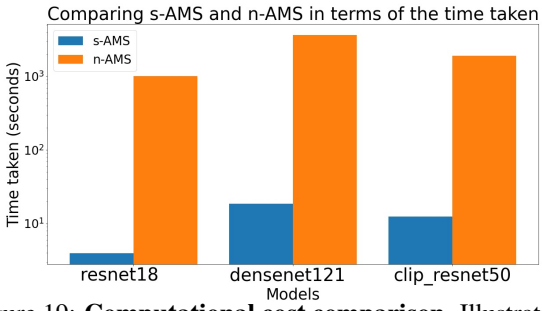

Figure 19: **Computational cost comparison**. Illustration of the computational time, required to compute an a s-AMS and n-AMS for one representation, logarithmic scale. n-AMS were computed on the Imagenet "Fall11" dataset.

We used torch-dreams [††] to generate the s-AMS. The primary advantage of generating s-AMS has been the fact that it does not require the dataset upon which the model was trained. When generating s-AMS for all the neurons in the target layers, the time taken is comparable to that of finding the n-AMS for each neuron. It is also worth noting that generating s-AMS is multiple orders of magnitude faster than finding the n-AMS for a single neuron since one does not have to iterate over the whole dataset in the former case. We

[‖] Parameters for the s-AMS generation could be found by the following link: `https://github.com/openai/CLIP-featurevis`

[**] `https://github.com/greentfrapp/lucent`

[††] `https://github.com/Mayukhdeb/torch-dreams`

generated s-AMS with a width and height of 224 with a weight decay of 1e-2 for 300 iterations per neuron. For robustness of the optimization process, Random-Affine transformations were used with a maximum rotation of 15 degrees and the maximum absolute fraction of x and y translations set to 0,1.

In order to generate the AMS for a single neuron, it took 1017.24 seconds for n-AMS compared to 3.92 seconds for s-AMS in the ResNet18. Similar results were also found for DenseNet121 and the CLIP image encoder with 3651.32 seconds for n-AMS and 18.53 seconds for s-AMS on the DenseNet121 and 1903.81 seconds for n-AMS and 12.44 seconds for s-AMS on the CLIP image encoder.

G Understanding DORA outliers

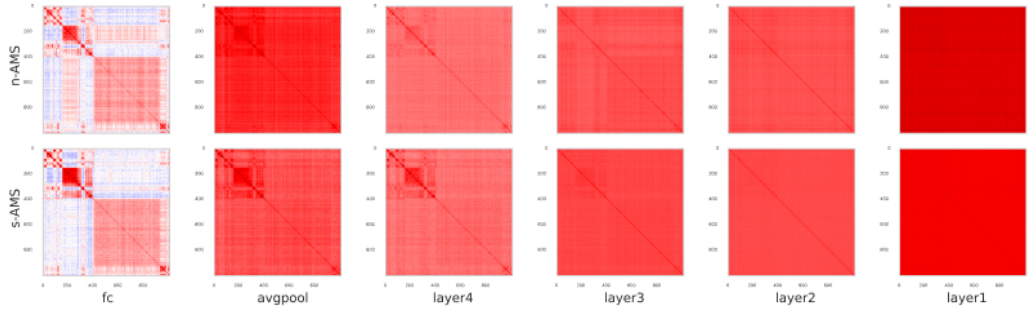

Figure 20: **Cosine similarities between embeddings of natural and synthetic AMS:** each heatmap illustrates cosine similarities between embeddings of AMS for different layers of ResNet 18. Natural AMS (top row) and synthetic AMS (bottom row) were computed for the 1000 logits of the network. 'Seismic' colorscheme is used to illustrate cosine similarity values (1, red – collinearity; 0, white – orthogonality; -1, blue – negative collinearity). From this graph we can observe similar behaviour of n-AMS and s-AMS, i.e., distances between concepts in real n-AMS are preserved in synthetic images s-AMS.

To study the relationship between natural and synthetic Activation-Maximization signals, we compute both n-AMS and s-AMS for the output layer of the ResNet18 network, pre-trained on ImageNet. In detail, let $\mathcal{H}$ be the set of output logits $h_i, i \in [1, .., 1000]$, for which we compute n-AMS and s-AMS for each logit yielding $\mathcal{S}_n$ and $\mathcal{S}_s$, both containing 1000 signals (images). For each of the signals, we collect the activations across different layers of the network. Figure 20 expands the results from Figure 7 in the main paper, where we can observe the cosine distances between the embeddings of n-AMS and s-AMS across multiple layers of the model. We can observe that for low-layer representations, and activations of signals, both s-AMS and n-AMS are becoming more and more alike. Figure 21 illustrates the correlations between pairwise distances within n-AMS and within s-AMS. From this figure, we can observe, that for the high-level representation layers of ResNet18, namely "fc", "avgpool", "layer4" cosine and euclidean distances between activations within n-AMS and s-AMS signals are highly correlated, hence concluding a strong relationship between s-AMS and n-AMS.

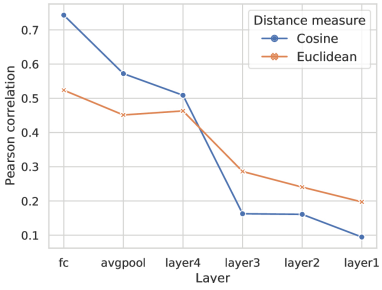

Figure 21: **Correlations between pairwise distances of n-AMS and s-AMS across different layers.** Illustration of the correlation between pairwise distances between n-AMS and s-AMS signals. The high correlation between embeddings of s-AMS and n-AMS which can be observed in the last layers of the network illustrates the similarity of the relationship between the high-level concepts inherent in s-AMS and n-AMS.

Furthermore, we study how much neurons are activated by other neurons top activating images. Therefore, we create a dataset for each individual neuron consisting of 100 top n-AMS and with additional 100 random images created a dataset for binary

classification. we compute a matrix, consisting of AUC-ROC scores — for each individual neuron we have collected 100 top n-AMS and with additional 100 random images created a dataset for binary classification. More precisely, for each logit $h_i, i \in [1, ..., 1000]$, a set $\mathcal{S}_i = \{s_i^j, j \in [1, .., N]\}$ of top $N = 100$ n-AMS was collected. Further, we create a dataset $D_i = (\mathcal{S}_i, \mathcal{R}_i)$, where $\mathcal{R}_i$ are 100 random images from ImageNet, for one-vs-all binary classification, between representation's i and just random images. Finally we compute the resulting matrix E, where each entry could be computed as $E_{ij} = AUC(h_j)|_i$, where AUC is a function that measures AUC-ROC performance of representation h_j in binary classification setup for the dataset D_i.

Pairwise cosine distances between embeddings of n-AMS and s-AMS, as well as the matrix E are illustrated in the Figure 7. We can assess visually and empirically by the correlation between the distances, that distances between s-AMS are strongly related not only to the distances between n-AMS but also with a measure of how close are representations in behaving toward similar concepts.

The availability of real images is not necessary for the analysis of connections between the representations, i.e., the neurons – natural images could be easily substituted with the collection of synthetic stimuli, without loss of the information. This is also illustrated in Figure 22, from which we can observe how the network's activations inferred by the collection of AM-signals for 1000 ImageNet classes conserve the semantic similarity.

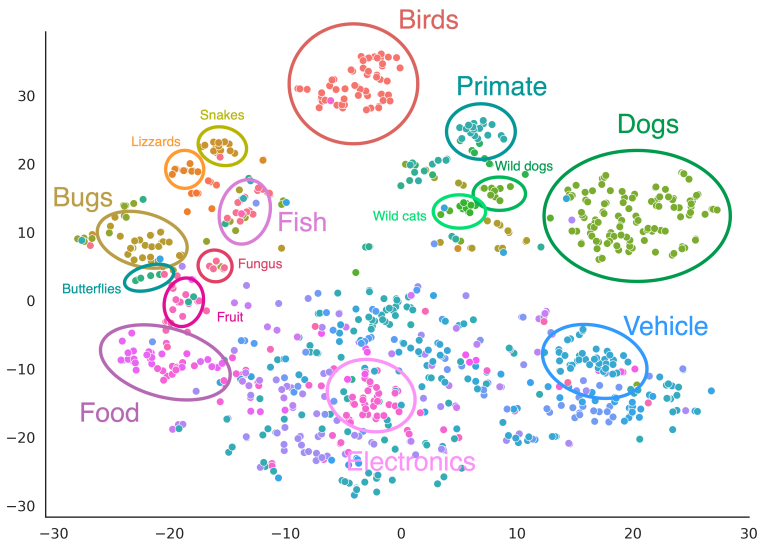

Figure 22: **Comparison of the class similarities between ImageNet classes based on their s-AMS.** We computed the s-AMS for each of the 1000 output (class) neurons of a VGG-16 network, trained on ImageNet. Afterward, we re-fed the Feature Visualizations into the network and collect their activations at the logit layer. Finally, we plotted the activations with the t-SNE in a 2D space and colored them regarding the given meta-classes information, that bundle similar classes. We can observe that semantically similar classes are also close in terms of the distance between the corresponding activation vectors of their Feature Visualizations.

H Experimental setup

All described experiments, if not stated otherwise, were performed on the Google Colab Pro [5] environment with the GPU accelerator.