Contextualised out-of-distribution detection using pattern iIdentification

Romain Xu-Darme, Julien Girard, Darryl Hond, Gabriele Incorvaia, Zakaria
Chihani

Contextualised Out-of-Distribution Detection using Pattern Identification

Romain Xu-Darme[1,3](✉), Julien Girard-Satabin[1], Darryl Hond[2], Gabriele Incorvaia[2], and Zakaria Chihani[1]

[1] Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
`<first>.<last>@cea.fr`
[2] Thales UK, Research, Technology and Innovation, Reading, UK
`<first>.<last>@uk.thalesgroup.com`
[3] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France

Abstract In this work, we propose CODE, an extension of existing work from the field of explainable AI that identifies class-specific recurring patterns to build a robust Out-of-Distribution (OoD) detection method for visual classifiers. CODE does not require any classifier retraining and is OoD-agnostic, *i.e.,* tuned directly to the training dataset. Crucially, pattern identification allows us to provide images from the In-Distribution (ID) dataset as reference data to provide additional context to the confidence scores. In addition, we introduce a new benchmark based on perturbations of the ID dataset that provides a known and quantifiable measure of the discrepancy between the ID and OoD datasets serving as a reference value for the comparison between OoD detection methods.

Keywords: Out-of-distribution detection · Explainable AI · Pattern identification

1 Introduction

A fundamental aspect of software safety is arguably the modelling of its expected operational domain through a formal or semi-formal specification, giving clear boundaries on when it is sensible to deploy the program, and when it is not. It is however difficult to define such boundaries for machine learning programs, especially for visual classifiers based on artificial neural networks (ANN) that process high-dimensional data (images, videos) and are the result of a complex optimisation procedure. In this context, Out-of-Distribution (OoD) detection - which aims to detect whether an input of an ANN is In-Distribution (ID) or outside of it - serves several purposes: 1) it helps characterise the extent to which the ANN can operate outside a bounded dataset; 2) it constitutes a surrogate measure of the generalisation abilities of the ANN; 3) it can help assess when an input is too far away from the operational domain, which prevents misuses of the program and increases its safety. However, one crucial aspect missing from current OoD detection methods is the ability to provide some form of explanation of their decision. Indeed, most approaches are based on a

statistical model of the system behaviour, built upon an abstract representation of the input data, sometimes turning OoD detection into an opaque decision that may appear arbitrary to the end-user. While it would be possible to generate a visual representation of the abstract space using tSNE and to highlight ID data clusters for justifying the OoD-ness of a given sample, tSNE is extremely dependent on the choice of hyper-parameters, sometimes generating misleading visualisations [25]. In this regard, methods from the field of Explainable AI (XAI), which are typically used to provide some insight about the decision-making process of the model, can be adapted to build models for OoD detection that provide some context information to justify their decision. In the particular task of image classification, XAI methods can help extract visual cues that are class-specific (*e.g.,* a bird has wings), and whose presence or absence can help characterise the similarity of the input image to the target distribution (*e.g.,* an object classified as a bird that shows neither wings nor tail nor beak is probably an OoD input). Therefore, in this work we make the following contributions:

1. We introduce a new benchmark based on perturbations of the ID dataset which provides a known and quantifiable evaluation of the discrepancy between the ID and OoD datasets that serves as a reference value for the comparison between various OoD detection methods (Sec. 3).
2. We propose CODE, an OoD agnostic detection measure that does not require any fine-tuning of the original classifier. Pattern identification allows us to provide images from the ID dataset as reference points to justify the decision (Sec. 4). Finally, we demonstrate the capabilities of this approach in a broad comparison with existing methods (Sec. 5).

2   Related Work

Out-of-distribution detection. In this work, we focus on methods that can apply to pre-trained classifiers. Therefore, we exclude methods which integrate the learning of the confidence measure within the training objective of the model, or specific architectures from the field of Bayesian Deep-Learning that aim at capturing uncertainty by design. Moreover, we exclude *OoD-specific* methods that use a validation set composed of OoD samples for the calibration of hyper-parameters, and focus on *OoD-agnostic* methods that require only ID samples. In this context, the maximum softmax probability (MSP) obtained after normalisation of the classifier logits constitutes a good baseline for OoD detection [8]. More recently, ODIN [15] measures the local stability of the classifier using gradient-based perturbations, while MDS [13] uses the Mahalanobis distance to class-specific points in the feature space. [16] proposes a framework based on energy scores, which is extended in the DICE method [21] by first performing a class-specific directed sparsification of the last layer of the classifier. ReAct [20] also modifies the original classifier by rectifying the activation values of the penultimate layer of the model. [6] proposes two related methods: MaxLogit - based on the maximum logit value - and KL-Matching which measures the KL divergence between the output of the model and the class-conditional mean

softmax values. The Fractional Neuron Region Distance [9] (FNRD) computes the range of activations for each neuron over the training set in order to empirically characterise the statistical properties of these activations, then provides a score describing how many neuron outputs are outside the corresponding range boundaries for a given input. Similarly, for each layer in the model, [18] computes the range of pairwise feature correlation between channels across the training set. ViM [24] adds a dedicated logit for measuring the OoD-ness of an input by using the residual of the feature against the principal space. KNN [22] uses the distance of an input to the k-th nearest neighbour. Finally, GradNorm [10] measures the gradients of the cross-entropy loss *w.r.t.* the last layer of the model.

Evaluation of OoD detection. All methods presented above are usually evaluated on different settings (*e.g.,* different ID/OoD datasets), sometimes using only low resolution images (*e.g.,* MNIST [2]), which only gives a partial picture of their robustness. Therefore, recent works such as [6,21] - that evaluate OoD methods on datasets with higher resolution (*e.g.,* ImageNet [1]) - or Open-OoD [27] - which aims at standardising the evaluation of OoD detection, anomaly detection and open-set recognition into a unified benchmark - are invaluable. However, when evaluating the ability of a method to discriminate ID/OoD datasets, it is often difficult to properly quantify the margin between these two datasets, *independently* from the method under test, and to establish a "ground truth" reference scale for this margin. Although [27] distinguishes *"near-OoD datasets [that] only have semantic shift compared with ID datasets"* from *"far-OoD [that] further contains obvious covariate (domain) shift"*, this taxonomy lacks a proper way to determine, given two OoD datasets, which is "further" from the ID dataset. Additionally, [17] generates "*shifted sets*" that are *"perceptually dissimilar but semantically similar to the training distribution"*, using a GAN model for measuring the *perceptual* similarity, and a deep ensemble model for evaluating the *semantic* similarity between two images. However, this approach requires the training of multiple models in addition to the classifier. Thus, in this paper we propose a new benchmark based on gradated perturbations of the ID dataset. This benchmark measures the correlation between the OoD detection score returned by a given method when applied to a perturbed dataset (OoD), and the intensity of the corresponding perturbation.

Part detection. Many object recognition methods have focused on part detection, in supervised (using annotations [28]), weakly-supervised (using class labels [14]) or unsupervised [4,26,29] settings, primarily with the goal of improving accuracy on hard classification tasks. To our knowledge, the PARTICUL algorithm [26] is the only method that includes a confidence measure associated with the detected parts (used by the authors to infer the visibility of a given part). PARTICUL aims to identify *recurring patterns* in the latent representation of a set of images processed through a pre-trained CNN, in an unsupervised manner. It is, however, restricted to homogeneous datasets where all images belong to the same macro-category. For more heterogeneous datasets, it becomes difficult to find recurring patterns that are present across the entire training set.

3 Beyond cross-dataset evaluation: measuring consistency against perturbations

In this section, we present our benchmark for evaluating the consistency of OoD detection methods using perturbations of the ID dataset.

Let $f : \mathcal{X} \to \mathbb{R}^N$ be a classifier trained on a dataset $X_{train} \sim \mathcal{P}_{id}$, where $\mathcal{P}_{id}$ is a distribution over $\mathcal{X} \times \mathbb{R}^N$ and N is the number of categories learned by the classifier. We denote $\mathcal{D}_{id}$ the marginal distribution of $\mathcal{P}_{id}$ over $\mathcal{X}$. For any image $x \in \mathcal{X}$, f outputs a vector of logits $f(x) \in \mathbb{R}^N$. The index of the highest value in $f(x)$ corresponds to the most probable category (or *class*) of x - relative to all other categories. Without loss of generality, the goal of an OoD detection method is to build a class-conditional[4] confidence function $C : \mathcal{X} \times \mathbb{R}^N \to \mathbb{R}$ assigning a score to each pair (x, y), where y can be either the ground truth label of x when known, or the prediction $f(x)$ otherwise. This function constitutes the basis of OoD detection, under the assumption that images belonging to $\mathcal{D}_{id}$ should have a higher confidence score than images outside $\mathcal{D}_{id}$.

A complete evaluation of an OoD detection method would require the application of the confidence function C on samples representative of the ID and OoD distributions. However, it is not possible to obtain a dataset representative of all possible OoD inputs. Instead, **cross-dataset OoD evaluation** consists in drawing a test dataset $X_{test} \sim \mathcal{D}_{id}$ (with $X_{test} \neq X_{train}$), choosing a different dataset $D_{ood} \not\sim \mathcal{D}_{id}$, then measuring the *separability* of $C(X_{test})$ and $C(D_{ood})$, where $C(X)$ denotes the distribution of scores computed over dataset X using C. Three metrics are usually used: Area Under the ROC curve (AUROC); Area Under the Precision-Recall curve (AUPR), False Positive Rate when the true positive rate is 95% (FPR95).

In this work, in addition to cross-dataset evaluation, we propose to *generate* an OoD distribution $\mathcal{D}_{ood}$ by applying a perturbation to all images from $\mathcal{D}_{id}$. Although image perturbation is a standard method for evaluating the robustness of classifiers [7], our intent differs: rather than trying to capture the point of failure of a classifier, we monitor how the various confidence scores evolve when applying a perturbation of increasing intensity to the ID dataset. In practice, we use four transformations: Gaussian noise, Gaussian blur, brightness changes and rotations. More generally, a perturbation P_α is a function that applies a transformation of magnitude α to an image $x \in \mathcal{X}$ (*e.g.,* a rotation with angle α). When applying P_α over $\mathcal{D}_{id}$, we define the expected confidence as

$$E(P_\alpha, C) = \mathbb{E}_{x \sim \mathcal{D}_{id}}\big[C\big(P_\alpha(x), f(P_\alpha(x))\big)\big] \tag{1}$$

which is evaluated over the test set X_{test}. Although it would again be possible to measure the separability of ID and OoD confidence distributions, perturbations of small magnitude would result in almost identical distributions. Instead, we evaluate the correlation between the magnitude of the perturbation and the average confidence value of the perturbed dataset as the Spearman Rank Correlation

---

[4] Class-agnostic methods simply ignore the image label/prediction.

Coefficient (SRCC) r_s between α and $E(P_\alpha, C)$, using multiple magnitude values $(\alpha_0, \ldots, \alpha_n)$. $r_s = 1$ (resp. -1) indicates that the average confidence measure increases (resp. decreases) *monotonically* with the value of α, *i.e.,* that the measure is correlated with the magnitude of the perturbation. The key advantage of the SRCC resides in the ability to compare the general behaviour of various OoD detection methods that usually have different calibrations (*i.e.,* different range of values). Assuming that the discrepancy between $\mathcal{D}_{id}$ and $P_\alpha(\mathcal{D}_{id})$ is correlated to the magnitude of the perturbation α (ground truth), this benchmark measures the *consistency* of the OoD methods under test.

4   Contextualised OoD Detection using Pattern Identification

In this section, we present CODE, our proposal for building a *contextualised* OoD detector. CODE is an extension of the PARTICUL algorithm described in [26], which is intended to mine recurring patterns in the latent representation of a set of images processed through a CNN. Patterns are learnt from the last convolutional layer of the classifier f over the training set X_{train}, in a plug-in fashion that does not require the classifier to be retrained. Let v be the restriction of classifier f up to its last convolutional layer, *i.e., $f = l \circ v$,* where l corresponds to the last pooling layer followed by one or several fully connected layers. $\forall x \in \mathcal{X}$, $v(x) \in \mathbb{R}^{H \times W \times D}$ is a convolutional map of D-dimensional vectors. The purpose of the PARTICUL algorithm is to learn p distinct $1 \times 1 \times D$ convolutional kernels $K = [k_1, \ldots, k_p]$ (or *detectors*), such that $\forall x \in X_{train}$: 1) each kernel k_i strongly correlates with exactly one vector in $v(x)$ (*Locality* constraint); 2) each vector in $v(x)$ strongly correlates with at most one kernel k_i (*Unicity* constraint).

Learning class-conditional pattern detectors. While PARTICUL is an unsupervised approach restricted to fine-grained recognition datasets, CODE uses the training labels from X_{train} to learn p detectors *per class*. More precisely, let $K^{(c)} = [k_1^{(c)}, \ldots k_p^{(c)}]$ be the set of kernel detectors for class c. Similar to [26], we define the normalised activation map between kernel $k_i^{(c)}$ and image x as:

$$P_i^{(c)}(x) = \sigma\big(v(x) * k_i^{(c)}\big) \in \mathbb{R}^{H \times W} \tag{2}$$

where σ is the *softmax* normalisation function. We also define the cumulative activation map, which sums the normalised scores for each vector in $v(x)$, *i.e.,*

$$S^{(c)}(x) = \sum_{i=1}^{p} P_i^{(c)}(x) \in \mathbb{R}^{H \times W} \tag{3}$$

Then, we define the *Locality* and *Unicity* objective functions as follows:

$$\mathcal{L}_l = - \sum_{(x,y) \in X_{train}} \sum_{c=1}^{N} \sum_{i=1}^{p} \mathbb{1}_{[c=y]} \times \max \big(P_i^{(c)}(x) * u\big) \tag{4}$$

$$\mathcal{L}_c = \sum_{(x,y) \in X_{train}} \sum_{c=1}^{N} \mathbb{1}_{[c=y]} \times \max\left(0, \max\left(S^{(c)}(x)\right) - t\right) \tag{5}$$

where $\mathbb{1}$ is the indicator function, and u is a 3×3 uniform kernel that serves as a relaxation of the Locality constraint. Due to the softmax normalisation of the activation map $P_i^{(c)}(x)$, $\mathcal{L}_l$ is minimised when, for all images x of class c, each kernel $k_i^{(c)}$ strongly correlates with one and only one 3×3 region of the convolutional map $v(x)$. Meanwhile, $\mathcal{L}_c$ is minimised when, for all images x of class c, the sum of normalised correlation scores between a given vector in $v(x)$ and all kernels $k_i^{(c)}$ does not exceed a threshold $t = 1$, ensuring that no vector in $v(x)$ correlates too strongly with multiples kernels. The final training objective is $\mathcal{L} = \mathcal{L}_l + \lambda_u \mathcal{L}_u$. Importantly, *we do not explicitly specify a target pattern for each detector*, but our training objective will ensure that we obtain detectors for p different patterns for each class. Moreover, since patterns may be similar across different classes (e.g., the wheels on a car or a bus), we *do not treat images from other classes as negative samples* during training.

Confidence measure. After training, we build our confidence measure using the function $H_i^{(c)}(x) = \max_{v^* \in v(x)} (v^* * k_i^{(c)})$ returning the maximum correlation score between kernel $k_i^{(c)}$ and $v(x)$. Assuming that each detector will correlate more strongly with images from $\mathcal{D}_{id}$ than images outside of $\mathcal{D}_{id}$, we first estimate over X_{train} the mean value $\mu_i^{(c)}$ and standard deviation $\sigma_i^{(c)}$ of the distribution of maximum correlation scores $H_i^{(c)}$ for $(x,c) \sim \mathcal{P}_{id}$. Then, we define

$$C^{(c)}(x) = \frac{1}{p} \sum_{i=1}^{p} C_i^{(c)}(x), \text{with } C_i^{(c)}(x) = sig\left(\left(H_i^{(c)}(x) - \mu_i^{(c)}\right)/\sigma_i^{(c)}\right) \tag{6}$$

as the *class confidence score* for class c. Though it could be confirmed using a KS-test on empirical data, the logistic distribution hypothesis used for $H_i^{(c)}$ - rather of the normal distribution used in PARTICUL - is primarily motivated by the computational effectiveness[5] and the normalisation effect of the sigmoid sig that converts a raw correlation score into a value between 0 and 1. During inference, for $x \in \mathcal{X}$, the confidence measure $C(x)$ is obtained by weighting each class confidence score by the probability that x belongs to this class:

$$C(x) = \sum_{c=1}^{N} C^{(c)}(x) \times P(Y = c \mid X = x) \tag{7}$$

where the categorical distribution $P(Y \mid X = x)$ is obtained from the vector of normalised logits $n = \sigma(f(x))$, as shown in Fig. 1. Note that it would be possible to build $C(x)$ using only the confidence score of the most probable class $c = \arg\max(f(x))$. However, using the categorical distribution allows us to mitigate the model (in)accuracy, as we will see in Sec. 5.

---

[5] Although good approximations of the normal CDF using sigmoids exist [3]

Figure 1: **CODE inference overview**. When processing a new sample x, the confidence measure sums up the average contribution of the detectors from each class weighted by the probability of x belonging to that class.
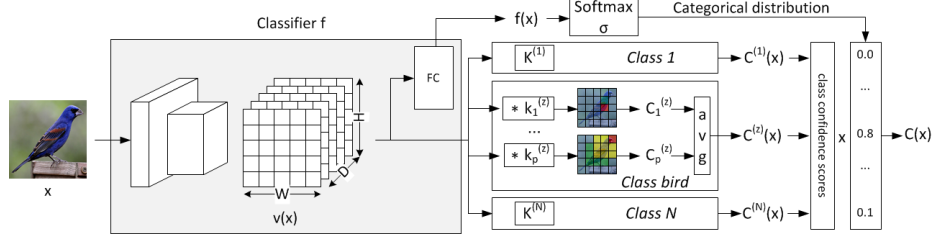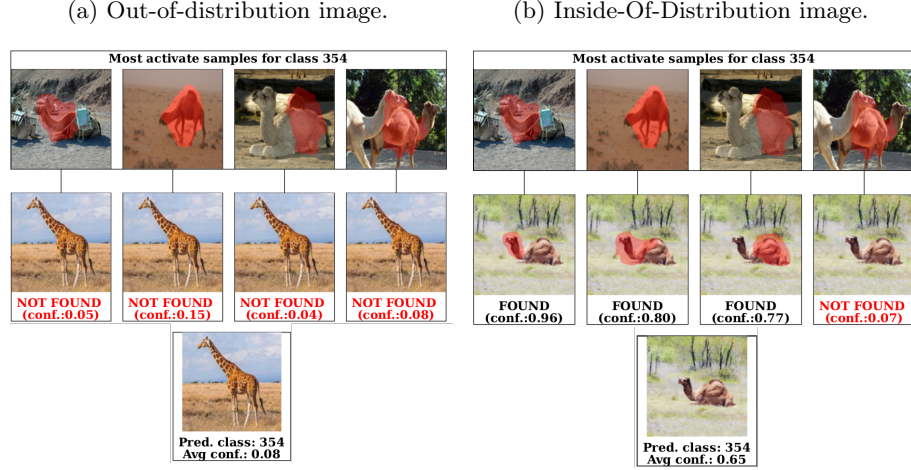

Figure 2: **Explanations generated by CODE for ID and OoD samples**. For each image, the classification as ID/OoD rely on the presence/absence of class-specific visual cues extracted from the training set.

(a) Out-of-distribution image.   (b) Inside-Of-Distribution image.

Extracting examples One of the key advantages of CODE over existing OoD detection methods resides in the ability to provide a visual justification of the confidence measure. For each detection kernel $k_i^{(c)}$ for class c, we first identify the sample $(x, c) \in X_{train}$ that most faithfully represents the distribution of correlation scores $H_i^{(c)}$ across the training (in practice, we select the sample whose correlation score is closest to $\mu_i^{(c)}$). Then, as in [26], we locate the pattern associated with this detector inside image x using the SmoothGrads [19] algorithm. This operation anchors each detector for each class to a part of an image in the training set. Moreover, the ability to visualise patterns can also serve as a sanity check to verify that our method has indeed learned unique and *relevant* patterns *w.r.t.* to the class object.

For each new image, as shown in Fig. 2, we first identify the predicted class $c = \arg\max \big(f(x)\big)$ returned by the classifier. Then, we use the individual con-

fidence scores $C_i^{(c)}(x)$ for each detector of class c to infer the presence or absence of each pattern. When the confidence score of a given detector is above a given threshold (*e.g.*, $C_i^{(c)}(x) > 0.3$), we highlight the corresponding pattern inside image x (again using SmoothGrads) and display the most correlated sample from the training set as a reference. In summary, *we justify the OoD-ness of the new image by pointing out the presence or absence of class-specific recurring patterns that were found in the training set.* Note that although our confidence measure is computed using *all* class confidence scores (weighted by the categorical distribution, see above), we believe that an explanation built solely on the most probable class can provide enough justification for the decision, while being sufficiently concise to be understandable by the user.

5   Experiments

In this section, we start by describing the experimental setup designed to answer the following research questions: 1) How does CODE fare against other detection methods on a cross-dataset OoD evaluation benchmark? 2) What is the influence of weighting all class-condition confidence scores (Eq. 7) rather than using only the confidence score of the most probable class? 3) How does the number p of detectors per class influences CODE detection capabilities? 4) How do OoD detection methods behave when applying perturbations on the ID dataset?

Setup We performed our evaluation using the OpenOoD framework [27], which already implements most recent OoD detection methods. For each ID dataset, we used the provided pre-trained classifier for feature extraction and trained 4 or 6 pattern detectors per class, using the labels of the training set and the objective function described in Sec. 4. After cross-validation on a CIFAR10 v. CIFAR100 detection benchmark, we set $\lambda_u = 1$, putting equal emphasis on the locality and unicity constraints. Although CODE trains a high number of detectors, the learning process remains computationally efficient since the classifier is not modified and only the detectors of the labelled class are updated during the back-propagation phase. Additionally, for large datasets such as ImageNet, detectors from different classes can be trained *in parallel* on chunks of the dataset corresponding to their respective class. We trained our detectors with RMSprop (learning rate 5×10^{-4}, weight decay 10^{-5}), for 30 epochs (ImageNet) or 200 epochs (all other ID datasets). As a comparison, we also implemented a class-based FNRD [9], extracting neuron activation values at different layers of the classifier.

Cross-dataset OoD evaluation The cross-dataset evaluation implemented in OpenOoD includes a OoD detection benchmark and an Open Set Recognition (OSR) benchmark. For the OoD detection benchmark, we use the ID/Near-OoD/Far-OoD dataset split proposed in [27]. For the OSR benchmark, as in [27], M-6 indicates a 6/4 split of MNIST [2] (dataset split between 6 closed set classes used for training and 4 open set classes), C-6 indicates a 6/4 split of CIFAR10 [11],

Table 1: **Comparison of AUROC scores between CODE and state-of-the-art methods on a cross-dataset benchmark.** Results with * are extracted from [27] - keeping only OoD-agnostic methods. We also add results of our implementation of a class-based FNRD [9]. Experiments on ImageNet using 6 CODE detectors have not yet been conducted due to limited ressources (denoted ⊙). For readability, AUPR and FPR95 are omitted but available upon request.

| | OSR | | | | | OoD Detection (Near-OoD / Far-OoD) | | | | |
	M-6	C-6	C-50	T-20	Avg.	MNIST	CIFAR-10	CIFAR-100	ImageNet	Avg.
MSP* [8]	96.2	85.3	81.0	73.0	83.9	91.5 / 98.5	86.9 / 89.6	80.1 / 77.6	69.3 / 86.2	81.9 / 87.9
ODIN* [15]	98.0	72.1	80.3	**75.7**	81.8	92.4 / 99.0	77.5 / 81.9	79.8 / 78.5	73.2 / 94.4	80.7 / 88.4
MDS* [13]	89.8	42.9	55.1	57.6	62.6	**98.0** / 98.1	66.5 / 88.8	51.4 / 70.1	68.3 / 94.0	71.0 / 87.7
Gram* [18]	82.3	61.0	57.5	63.7	66.1	73.9 / **99.8**	58.6 / 67.5	55.4 / 72.7	68.3 / 89.2	64.1 / 82.3
EBO* [16]	**98.1**	84.9	82.7	75.6	85.3	90.8 / 98.8	87.4 / 88.9	71.3 / 68.0	73.5 / 92.8	80.7 / 87.1
GradNorm* [10]	94.5	64.8	68.3	71.7	74.8	76.6 / 96.4	54.8 / 53.4	70.4 / 67.2	75.7 / 95.8	69.4 / 78.2
ReAct* [20]	82.9	85.9	80.5	74.6	81.0	90.3 / 97.4	87.6 / 89.0	79.5 / 80.5	79.3 / 95.2	84.2 / 90.5
MaxLogit* [6]	98.0	84.8	82.7	75.5	85.3	92.5 / 99.1	86.1 / 88.8	**81.0** / 78.6	73.6 / 92.3	83.3 / 89.7
KLM* [6]	85.4	73.7	77.4	69.4	76.5	80.3 / 96.1	78.9 / 82.7	75.5 / 74.7	74.2 / 93.1	77.2 / 86.7
ViM* [24]	88.8	83.5	78.2	73.9	81.1	94.6 / 99.0	88.0 / 92.7	74.9 / **82.4**	79.9 / **98.4**	84.4 / **93.1**
KNN* [22]	97.5	**86.9**	**83.4**	74.1	**85.5**	96.5 / 96.7	**90.5** / 92.8	79.9 / 82.2	**80.8** / 98.0	**86.9** / 92.4
DICE* [21]	66.3	79.3	82.0	74.3	75.5	78.2 / 93.9	81.1 / 85.2	79.6 / 79.0	73.8 / 95.7	78.2 / 88.3
FNRD [9]	59.4	68.2	58.4	54.3	60.1	84.8 / 97.1	70.2 / 71.5	54.6 / 58.5	75.4 / 87.5	71.3 / 78.7
- This work										
CODE (p=4)	74.7	86.7	76.5	62.4	75.1	81.8 / 99.5	87.8 / 90.7	73.9 / 72.4	76.6 / 84.4	80.0 / 86.8
most probable class only	73.7	86.4	74.6	61.3	74.0	80.5 / 99.5	87.4 / 90.3	72.2 / 71.0	73.7 / 77.3	78.5 / 84.5
CODE (p=6)	73.7	86.0	76.1	61.5	74.3	82.2 / 99.2	88.5 / 92.4	73.0 / 76.4	⊙	
most probable class only	72.8	85.7	73.9	60.4	73.2	81.8 / 98.7	87.8 / 91.8	70.9 / 74.5	⊙	

C-50 indicates a 50/50 split of CIFAR100 [11] and TIN-20 indicates a 20/180 split of TinyImageNet [12]. The AUROC score is averaged over 5 random splits between closed and open sets.

The results, summarised in Table 1, show that CODE displays OoD detection capabilities on par with most state-of-the-art methods (top-10 on OSR benchmark, top-8 on Near-OoD detection, top-9 on Far-OoD detection). Moreover, as discussed in Sec. 4, using the categorical distribution of the output of the classifier to weight class confidence scores systematically yields better results than using only the confidence score of the most probable class (up to 7% on the Far-OoD benchmark for ImageNet). Interestingly, increasing the number of detectors per class from 4 to 6 does not necessarily improve our results. Indeed, the Unicity constraint (Eq. 5) becomes harder to satisfy with a higher number of detectors and is ultimately detrimental to the Locality constraint (Eq. 4). This experiment also shows that the choice of Near-OoD/Far-OoD datasets in OpenOoD is not necessarily reflected by the average AUROC scores. Indeed, for CIFAR100, most methods exhibit a higher AUROC for Near-OoD datasets than for Far-OoD datasets. This observation highlights the challenges of selecting and sorting OoD datasets according to their relative "distance" to the ID dataset, without any explicit formal definition of what this distance should be. In this regard, our proposed benchmark using perturbations of the ID dataset aims at providing a quantifiable distance between ID and OoD datasets.

Table 2: **Summary of the perturbations**, with definition of α and its range.

Perturbation P	Description	Range for α
Blur	Gaussian blur with kernel 3×3 and standard deviation $\sigma = \alpha$	$\alpha \in [0.0, 10]$
Noise	Gaussian noise with ratio α	$\alpha \in [0, 1.0]$
Brightness	Blend black image with ratio $1 - \alpha$	$\alpha \in [0.1, 1.0]$
Rotation forth (R+)	Rotation with degree α	$\alpha \in [0, 180]$
Rotation back (R-)	Rotation with degree α	$\alpha \in [180, 360]$

Consistency against perturbations We also evaluated all methods on our perturbation benchmark (see Sec. 3), measuring the Spearman Rank correlation coefficient (SRCC) between the magnitude of the perturbation (see Table 2) and the average confidence measured on the perturbed dataset. The results, shown in Table 3, reveal that, on average, *CODE seem to correlate more strongly to the magnitude of the perturbation than all other methods.* Moreover, some OoD methods sometimes display unexpected behaviours, depending on the choice of dataset and perturbation, as shown in Fig. 3. In particular, MSP tends to increase with the noise ratio, hence the success of adversarial attacks [5,23]. Additionally, by construction, any perturbation reducing the amplitude of neuron activation values (blur, brightness) has the opposite effect of increasing the FNRD. Gram also increases with the noise ratio and is highly sensitive to rotations, although we do not have a satisfactory explanation for this particular behaviour. We also notice that - contrary to our expectations - the average confidence does not monotonously decrease when rotating images from 0 to 180°: all methods show periodic local maximums of the average confidence that may indicate a form of invariance of the network *w.r.t.* rotations of specific magnitude (45° for CIFAR10, 90° for CIFAR100/ImageNet, 180° for MNIST). This effect seems amplified for CIFAR100 (see Fig. 3). Finally, we notice that the top-3 methods for Near-OoD detection (KNN, ViM and ReAct) also strongly correlate with the magnitude of the perturbation, which opens the door to a more in-depth analysis of the relationship between the two benchmarks.

6   Conclusion & Future Work

In this paper, we have demonstrated how the detection of recurring patterns can be exploited to develop CODE, an OoD-agnostic method that also enables a form of visualisation of the detected patterns. We believe that this unique feature can help the developer verify visually the quality of the OoD detection method and therefore can increase the safety of image classifiers. More generally, in the future we wish to study more thoroughly how part visualisation can be leveraged to fix or improve the OoD detection method when necessary. For instance, we noticed some redundant parts during our experiments and believe that such redundancy could be identified automatically, and pruned during the training process to pro-

Table 3: **Comparison of OoD methods on our perturbation benchmark.** For each perturbation, ↑ (resp. ↓) indicates that the average confidence on the perturbed dataset should increase (resp. decrease) with α, *i.e.,* that the sign of the SRCC should be positive (resp. negative). Results in red indicate either a weak correlation (absolute value lower than 0.3) or an unexpected sign of the correlation coefficient, *e.g.,* the average Gram confidence score increases with the noise ratio on CIFAR100 ($r_s = 1.0$) when it should be decreasing. Results in **bold** indicate a strong expected correlation (absolute value greater than 0.9). The last column represents the average correlation score, taking into account the expected sign of the correlation (results with * are partial average values). ☉ indicates a timeout during the experiments.

	CIFAR10					CIFAR100					ImageNet					Avg.
	Noise ↓	Blur ↓	Bright. ↑	R+ ↓	R- ↑	Noise ↓	Blur ↓	Bright. ↑	R+ ↓	R- ↑	Noise ↓	Blur ↓	Bright. ↑	R+ ↓	R- ↑	
MSP	-0.22	-0.88	**0.98**	-0.55	0.56	0.33	-0.78	**0.99**	-0.32	0.31	0.71	**-1.0**	**1.0**	-0.77	0.85	0.54
ODIN	-0.85	-0.7	0.18	-0.15	0.13	-0.15	-0.77	0.75	-0.22	0.21	0.12	-0.87	0.2	-0.81	0.81	0.45
MDS	**-1.0**	0.41	0.84	-0.03	0.19	**-1.0**	0.68	0.84	-0.03	0.2	**-1.0**	0.98	-0.35	-0.16	0.11	0.20
Gram	1.0	**-1.0**	**1.0**	-0.15	-0.02	1.0	-0.83	**1.0**	-0.23	0.25	☉	☉	☉	☉	☉	0.24*
EBO	-0.62	-0.88	**0.96**	-0.33	0.29	-0.32	-0.78	**0.99**	-0.22	0.22	0.63	**-0.93**	**1.0**	-0.78	0.75	0.56
GradNorm	0.05	-0.69	**-1.0**	-0.04	-0.01	-0.71	-0.78	0.88	-0.32	0.31	0.75	**-0.93**	**1.0**	-0.47	0.41	0.32
ReAct	-0.44	-0.88	**0.96**	-0.37	0.33	-0.75	-0.78	**0.99**	-0.22	0.21	-0.25	**-0.95**	**1.0**	-0.66	0.66	0.63
MaxLogit	-0.62	-0.88	**0.96**	-0.33	0.33	0.0	-0.78	**0.99**	-0.22	0.22	0.65	**-0.93**	**1.0**	-0.78	0.78	0.54
KLM	-0.1	**-0.93**	0.95	-0.53	0.44	-0.01	-0.83	**0.99**	-0.27	0.26	☉	☉	☉	☉	☉	0.53*
ViM	-0.78	-0.83	**0.92**	-0.29	0.3	**-1.0**	-0.88	**1.0**	-0.18	0.38	**-1.0**	**-1.0**	**1.0**	-0.43	0.43	0.69
KNN	-0.36	-0.88	**0.99**	-0.46	0.4	-0.02	-0.79	**1.0**	-0.26	0.35	**-0.99**	**-1.0**	**1.0**	-0.5	0.5	0.63
DICE	**-0.97**	-0.88	**0.92**	-0.46	0.37	**-0.99**	-0.78	**0.99**	-0.32	0.22	0.65	**-0.93**	**1.0**	-0.74	0.74	0.64
FNRD	**-1.0**	0.58	**-0.99**	-0.11	0.08	**-1.0**	0.49	**-0.88**	-0.21	0.13	**-1.0**	-0.85	**0.99**	-0.35	0.35	0.21
CODE	-0.69	-0.88	**1.0**	-0.5	0.35	**-0.95**	-0.78	**0.99**	-0.3	0.29	-0.85	**-0.93**	**1.0**	-0.85	0.83	**0.75**

Figure 3: **Evolution of the average confidence score v. magnitude of the perturbation.** Curves in red indicate anomalous behaviours. Since all methods have different calibration values, we omit the units on the y-axis, focusing on the general evolution of the average confidence score over the perturbed dataset.
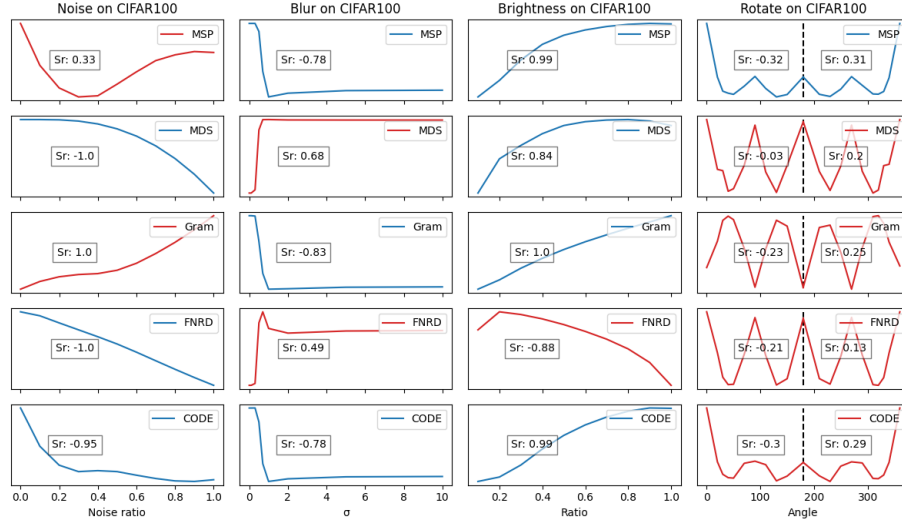
duce a more precise representation of each class. Additionally, providing a form of justification of the OoD-ness of a sample could also increase the *acceptability* of the method from the end-user point of view, a statement that we wish to confirm by conducting a user study in the future. Our experiments show that CODE offers consistent results on par with state-of-the-art methods in the context of two different OoD detection benchmarks, including our new OoD benchmark based on perturbations of the reference dataset. This new benchmark highlights intriguing behaviours by several state-of-the-art methods (*w.r.t.* specific types of perturbation) that should be analysed in details. Moreover, since these perturbations are equivalent to a *controlled covariate shift*, it would be interesting to evaluate covariate shift detection methods in the same setting. Finally, note that CODE could be applied to other part detection algorithms, provided that a confidence measure could be associated with the detected parts.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. CVPR 2009. pp. 248–255
2. Deng, L.: The MNIST database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)
3. Eidous, O.M., Al-Rawash, M.: Approximations for standard normal distribution function and its invertible. ArXiv (2022)
4. Han, J., Yao, X., Cheng, G., Feng, X., Xu, D.: P-CNN: Part-based convolutional neural networks for fine-grained visual categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(2), 579–590 (2022).
5. Hein, M., Andriushchenko, M., Bitterwolf, J.: Why RELU networks yield high-confidence predictions far away from the training data and how to mitigate the problem? CVPR 2019. pp. 41–50
6. Hendrycks, D., Basart, S., Mazeika, M., Mostajabi, M., Steinhardt, J., Song, D.X.: Scaling out-of-distribution detection for real-world settings. ICML 2022.
7. Hendrycks, D., Dietterich, T.G.: Benchmarking neural network robustness to common corruptions and perturbations. ArXiv (2018)
8. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. ICLR 2017.
9. Hond, D., Asgari, H., Jeffery, D., Newman, M.: An integrated process for verifying deep learning classifiers using dataset dissimilarity measures. International Journal of Artificial Intelligence and Machine Learning **11**(2), 1–21 (jul 2021).
10. Huang, R., Geng, A., Li, Y.: On the importance of gradients for detecting distributional shifts in the wild. NeurIPS 2021

11. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
13. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. NeurIPS (2018)
14. Li, H., Zhang, X., Tian, Q., Xiong, H.: Attribute Mix: Semantic Data Augmentation for Fine Grained Recognition. In: 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP). pp. 243–246 (2020).
15. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. ICLR 2018
16. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. NeurIPS 2020, pp. 21464–21475
17. Mukhoti, J., Lin, T.Y., Chen, B.C., Shah, A., Torr, P.H.S., Dokania, P.K., Lim, S.N.: Raising the bar on the evaluation of out-of-distribution detection. ArXiv (2022)
18. Sastry, C.S., Oore, S.: Detecting out-of-distribution examples with gram matrices. ICML (2020)
19. Smilkov, D., Thorat, N., Kim, B., Viégas, F.B., Wattenberg, M.: Smoothgrad: removing noise by adding noise. ArXiv (2017)
20. Sun, Y., Guo, C., Li, Y.: React: Out-of-distribution detection with rectified activations. NeurIPS 2021
21. Sun, Y., Li, Y.: Dice: Leveraging sparsification for out-of-distribution detection. ICCV 2021
22. Sun, Y., Ming, Y., Zhu, X., Li, Y.: Out-of-distribution detection with deep nearest neighbors. ICML 2022
23. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. ICLR 2014
24. Wang, H., Li, Z., Feng, L., Zhang, W.: Vim: Out-of-distribution with virtual-logit matching. CVPR 2022
25. Wattenberg, M., Viégas, F., Johnson, I.: How to use t-SNE effectively. Distill (2016). http://distill.pub/2016/misread-tsne
26. Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, M.C.: PARTICUL: Part Identification with Confidence measure using Unsupervised Learning (Jun 2022), XAIE: 2nd Workshop on Explainable and Ethical AI – ICPR 2022
27. Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., PENG, W., Wang, H., Chen, G., Li, B., Sun, Y., Du, X., Zhou, K., Zhang, W., Hendrycks, D., Li, Y., Liu, Z.: OpenOOD: Benchmarking generalized out-of-distribution detection. NeurIPS 2022 – Datasets and Benchmarks Track.
28. Zhao, X., Yang, Y., Zhou, F., Tan, X., Yuan, Y., Bao, Y., Wu, Y.: Recognizing Part Attributes With Insufficient Data. ICCV 2019
29. Zheng, H., Fu, J., Mei, T., Luo, J.: Learning Multi-attention Convolutional Neural Network for Fine-Grained Image Recognition. ICCV 2017