Data Invariants to Understand Unsupervised Out-of-Distribution Detection

Lars Doorenbos, Raphael Sznitman, and Pablo Márquez-Neila

University of Bern, Bern, Switzerland
{lars.doorenbos,raphael.sznitman,pablo.marquez}@unibe.ch

Abstract. Unsupervised out-of-distribution (U-OOD) detection has recently attracted much attention due to its importance in mission-critical systems and broader applicability over its supervised counterpart. Despite this increased attention, U-OOD methods suffer from important shortcomings. By performing a large-scale evaluation on different benchmarks and image modalities, we show in this work that most popular state-of-the-art methods are unable to consistently outperform a simple anomaly detector based on pre-trained features and the Mahalanobis distance (MahaAD). A key reason for the inconsistencies of these methods is the lack of a formal description of U-OOD. Motivated by a simple thought experiment, we propose a characterization of U-OOD based on the invariants of the training dataset. We show how this characterization is unknowingly embodied in the top-scoring MahaAD method, thereby explaining its quality. Furthermore, our approach can be used to interpret predictions of U-OOD detectors and provides insights into good practices for evaluating future U-OOD methods.

Keywords: Out-of-distribution detection · Unsupervised learning

1 Introduction

The use of deep learning (DL) models for mission-critical systems, such as in autonomous driving or medicine, is one of the most active research areas in computer vision. Yet, despite impressive performances in recent methods, their ability to extrapolate beyond their training data remains limited. For trained and deployed models, this is particularly problematic when processing images that are corrupted or whose content differs from their expectation. Predictions for unexpected images are often incorrect with high confidence and cannot be identified as such [4]. Ultimately, these silent failures deeply impact the reliability of machine learning systems in mission-critical applications and can have fatal consequences.

To mitigate these limitations, numerous *out-of-distribution* (OOD) detection methods have emerged in the recent past. Closely related to anomaly detection [44] and one-class learning [38], OOD detection aims to spot samples at inference time that do not belong to the training distribution and should not be processed by subsequent machine learning models. At their core, OOD detection

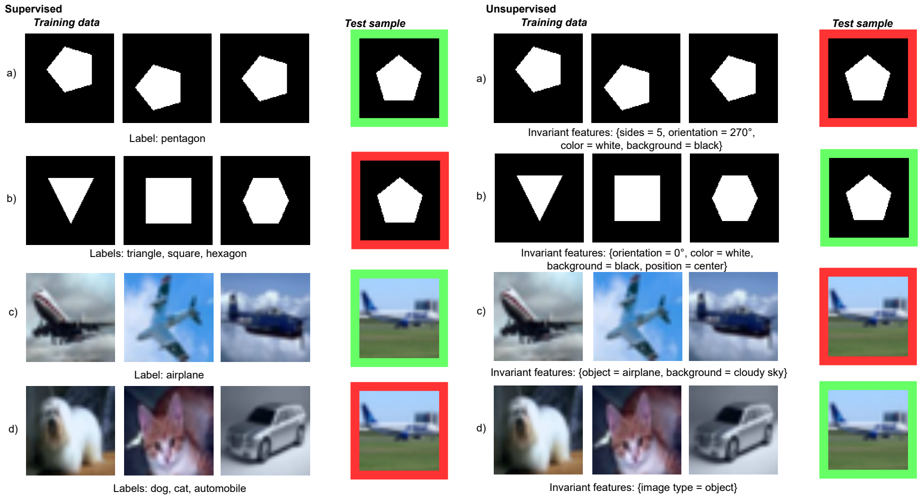

Supervised
Training data
Test sample
a)
Label: pentagon
b)
Labels: triangle, square, hexagon
c)
Label: airplane
d)
Labels: dog, cat, automobile
Unsupervised
Training data
Test sample
a)
Invariant features: {sides = 5, orientation = 270°, color = white, background = black}
b)
Invariant features: {orientation = 0°, color = white, background = black, position = center}
c)
Invariant features: {object = airplane, background = cloudy sky}
d)
Invariant features: {image type = object}

Fig. 1: The difference between supervised and unsupervised OOD. For the unsupervised case, invariants in the training data rather than class labels define what should be considered as OOD: in (a) a pentagon at a different angle leads to an OOD test sample, while (b) shows variants in shapes in the training set such that a pentagon is in-distribution at test time. While the train and test data are the same in each row, the interpretation of OOD differs in the supervised and unsupervised cases. Green and red boxes denote in- and out-of-distribution samples, respectively.

methods learn scoring functions that measure the level of anomaly, or *out-of-distributionness*, in test samples with respect to a training data distribution.

Broadly, OOD methods are categorized into supervised and unsupervised, as illustrated in Figure 1. Supervised OOD methods compute an OOD score by using the labels of the training dataset or by knowing the trained downstream network [19,22,23]. Conversely, unsupervised OOD (U-OOD) methods are agnostic to the downstream task or data labels, and learn tractable representations of the training images to compute OOD scores [7,12,15,45], which makes them more general than supervised methods and applicable to a larger range of scenarios.

Considering its significance and generality, the recent emergence of U-OOD methods is unsurprising. Yet with many methods reporting state-of-the-art performance [18,21,25,34,40,48,53,54,56], the overwhelming majority of these only validate their approach on one or two tasks. Given the broadness of U-OOD, these limited experimental validations have produced an inconsistent state-of-the-art, while simultaneously establishing an unclear sense of progress in the field. For instance, [20] showed excellent results for one-class tasks using CIFAR10 and ImageNet, only to be contradicted 8 months later in [6] using different data. More alarmingly, this trend of inconsistencies is being perpetuated with evaluation protocols remaining unchanged [5,21,32].

For this reason, we first aim to explore and assess the performance and robustness of existing U-OOD detectors by establishing a wide and varied panel of experiments using different datasets and setups. Not only do we show that U-OOD state-of-the-art methods perform erratically when evaluated over a wide and varied range of datasets and tasks (*i.e.* methods that perform extremely well on some datasets, frequently perform poorly on others), but that the relatively unnoticed MahaAD method [43] consistently outperforms all considered methods by remarkable margins in addition to being extremely simple, stable, and easy to train.

More fundamentally however, we hypothesize that despite the large number of recently proposed U-OOD methods, the main reason for this lack of overall consistency is that the fundamental concept of U-OOD remains vague and ill-defined. In fact, the vast majority of works fail to clearly define U-OOD, let alone provide an intuition to their approach's functioning. This subsequently leads to brittle methods and weak evaluation protocols.

Intuitively, a test sample should be considered OOD if it *looks different* from training samples. While this intuition seems straightforward, it is unclear how to characterize a training dataset or identify what makes a test sample similar or not to training samples. Yet, characterizing OOD is a fundamental necessity to not only produce reasonable U-OOD detectors, but also to properly evaluate and understand their behavior. Previous works have overlooked this important step and devised OOD detectors following more or less reasonable heuristics with limited formal justification. For example, using the observation that blurred images are assigned higher likelihoods compared to their original counterparts, SVD-RND [13] leveraged this property to characterize OOD by directly optimizing for it. Similarly, [42,54] identified OOD samples by correcting for their input complexity and the number of background pixels. Other examples include Puzzle-AE [47], which relied on solving puzzles of OOD images worse than their in-distribution counterparts, and MHRot [20] assumed that geometric transformations of OOD samples will be predicted incorrectly.

We also address here this apparent lack of a proper U-OOD definition by proposing a characterization based on identifying and leveraging image *invariants* of the training set. Following this idea, we formulate the general problem of finding dataset invariants and show that, when constrained to a linear setting, this formulation reduces to the MahaAD method, which unknowingly embodies a dataset invariant characterization. Importantly, we show that the invariants found within a training set are more relevant for U-OOD detection than its variant counterparts.

In summary, the contributions of this paper include (1) a thorough evaluation of numerous state-of-the-art U-OOD methods on different tasks and datasets, whereby highlighting that most methods perform erratically and inconsistently, (2) a novel interpretation of U-OOD using training set invariants, which allows for an appropriate definition of U-OOD and (3) a new U-OOD benchmark derived from our novel interpretation with invariants. A consequence of these contributions is that we shed light on why most recent methods do not perform well across datasets and, importantly, why the relatively unknown MahaAD method, that has been disregarded so far by most recent works in the field, is an excellent off-the-shelf U-OOD detector that should be included as a competitive baseline in future comparisons.

2 Related works

Methods such as one-class support vector machines [52], isolation forest [30], and local outlier factor [10] have traditionally been used for OOD detection in classical machine learning. However these methods suffer greatly when applied to high-dimensional spaces (*i.e.* images). Unsurprisingly, DL based methods have come to replace these more recently. Summarized here are some of the most relevant works on OOD detection using DL, while comprehensive surveys can be found in [11,60].

Supervised OOD detection approaches require either an explicitly trained classifier or a labelled dataset to work. One line of works uses a classifier's maximum softmax probability output as the OOD score [19,22,29]. Another, more closely related to U-OOD, exploits deep features of the task-specific trained classifiers [23,27,50]. However, as all these methods exploit relations between network predictions and the path taken to arrive at those predictions in some way, they are simply incompatible with the U-OOD setting.

On the other hand, U-OOD detection methods rely only on a set of *in*-distribution images to learn the characteristics of the in-distribution data. That is, they do not assume, or have access to, a trained downstream deep network or labeled dataset. Broadly, two families of methods are found in the literature. The first are generative models while the second are based on representation learning.

Generative models: These learn the distribution of images in high-dimensional spaces. However, most generative models are known to perform poorly in OOD detection [12,35], and many augmentations and improvements have been proposed to increase their performances. [54] showed that the likelihoods obtained by models such as Glow [24] or PixelCNN++ [49] are heavily influenced by the input complexity, and propose a likelihood ratio to correct for this. Interestingly, the work in [42] showed that background pixels dominate test sample likelihood scores, and attempt to correct for these by using the likelihood of a second model that tries to capture the population level background information. Similarly, Schirrmeister *et al.* [51] use the likelihood ratio with respect to a second model trained on a general, large scale dataset.

Representation learning: Instead of working in the image space, most U-OOD methods aim to learn a low-dimensional image embedding. Here, many works have opted for self-supervised learning strategies to simulate classification problems and train DL models to representative image features. One popular approach is predicting geometric transformations, such as image rotations, translations, scales, flips, or patch re-arrangements [7,15,20,58]. Other self-supervised approaches rely on auto-encoders and optionally perturb the input in some way to create more robust feature descriptions. Example perturbations include adding noise [46] or shuffling patches [47]. Further extensions propose to fit an auto-regressive model to the latent space [1] or to add a memory module [16]. Most recently, approaches based on contrastive learning have been advantageous [41,53,56].

However, various papers showed that learning features on the target domain is not necessary to reach high performance [6,36,43,59]. Bergman *et al.* [6] find that scoring samples by the distance to their k-nearest neighbours in the space of pre-trained ImageNet features outperformed all previous self-supervised methods. Xiao *et al.* [59] showed that exploiting features obtained from self-supervised —rather than supervised— training on ImageNet can lead to to high performance. Finally, Rippel *et al.* [43] combined Mahalanobis distances in the space of ImageNet features for state-of-the-art results on the MVTec dataset.

3 Invariants for Unsupervised OOD

In the supervised setting, similar to the problem of zero-shot learning, a sample is considered OOD if it cannot be assigned to one of the training set classes. In the unsupervised setting, however, defining OOD is more challenging as we do not know *a-priori* what and if any classes are present at all. As done in anomaly detection [44], one potential approach to define U-OOD could be to measure if a sample lies in a low-density region of the training data. But doing so would be inappropriate because whether few or many image examples of a specific class appear in a training set may only be a reflection of their natural prevalence, rather than being a real OOD sample. For instance, if one had a training set of dogs, the Norwegian Lundehund (*i.e.* a rare dog breed) would most likely appear in low-density regions of the training distribution, in contrast to German Shepherds (*i.e.* very common bread). Yet both should still be considered dogs. Instead, we propose to use *invariants* as a way to characterize U-OOD. Specifically, our idea is to first determine image invariants in the training set, and then detect OOD test samples by identifying if they keep the invariants of the training set.

To illustrate this, consider the toy examples in Fig. 1, where four different combinations of training sets and test examples are given. Recall that for the unsupervised case, no labels in the training data are available thus losing context as to what is or is not semantically OOD. However, the necessity to leverage context to disentangle relevant and irrelevant aspects of images remains key for U-OOD detection, since it is too broad to be meaningful without it (as stated in [2]). Hence, we assume that this necessary context is provided by a set of general features that we have at our disposal, that can describe the input images $\mathbf{x}$. For instance, these features could be $\mathbf{f}(\mathbf{x}) = \{\text{sides}(\mathbf{x}), \text{orientation}(\mathbf{x}), \text{color}(\mathbf{x}), ...\}$, or features coming from a network pre-trained on a general dataset. Given this, we want to summarize a training set by the *union* of features that are invariant over the entire training set. For example, Fig. 1(a) would use the combination of invariant features

$\{\text{sides} = 5, \text{orientation} = 270°, \text{color} = \text{white}, \text{background} = \text{black}\}$, and similarly $\{\text{orientation} = 0°, \text{color} = \text{white}, \text{background} = \text{black}, \text{position} = \text{center}\}$ for Fig. 1(b). At inference time then, a test sample described by this union of invariant features would be OOD if these features are no longer invariant with respect to the training set. In this sense, variant features from a dataset are in fact irrelevant for U-OOD detection, which stands in contrast to many previous methods that focused on learning a representation of the training distribution (*e.g.*, [13,31,61]).

In the remainder of this section, we begin by formalizing the above-mentioned idea and propose an approach to identifying these invariants for the general case. We then show how this is related to the MahaAD method [51]. In the experimental section, we demonstrate how MahaAD performs in comparison to recent methods and how it behaves in light of image invariants.

3.1 Formalization

Given a training set $\{\mathbf{x}_i\}_{i=1}^N$, with corresponding feature vectors, $\mathbf{f}(\mathbf{x}_i) \equiv \mathbf{f}_i \in \mathcal{F}$, we define an invariant as a non-constant function $g : \mathcal{F} \to \mathbb{R}$, such that $g(\mathbf{f}_i) = 0,\ \forall i$. That is, g is an invariant if it computes a constant value (*i.e.*, $g(\mathbf{f}_i) = 0$) for the elements of the training set, but in general may not compute the same constant value for other elements (*e.g.*, elements of a test set). Our goal then is to find a set of invariants, $G = \{g_1, \ldots, g_K\}$, over the set of training feature vectors. While doing so in one global optimization is challenging, we propose to solve this by solving a sequence of K problems, one per invariant,

$$\begin{aligned} g_k(\mathbf{f}_i) &= 0 \quad \forall i, \\ \|\nabla g_k(\mathbf{f}_i)\|_2 &\neq 0 \quad \forall i, \\ \nabla g_k(\mathbf{f}_i) \cdot \nabla g_j(\mathbf{f}_i) &= 0 \quad \forall i, j < k, \end{aligned} \tag{1}$$

where the first equality makes g_k zero for all training samples, the second equality prevents g_k from becoming a projection (*i.e.*, effectively making it non-constant) and the third equality requires that new invariants are different from all previously found invariants by making their gradients mutually orthogonal. After finding G, a test feature vector $\mathbf{f}$ will be considered OOD if $g_k(\mathbf{f}) \neq 0$ for any invariant k.

As noisy real-world data rarely lies in an exact manifold, solving Eq. (1) is unfeasible in practice even for a small number of invariants K. Instead, we relax Eq. (1) and express it as a minimization problem to find a set of soft invariants,

$$\begin{aligned} &\min_{g_k} \frac{1}{N} \sum_i g_k(\mathbf{f}_i)^2, \\ &\text{s.t.} \quad \|\nabla g_k(\mathbf{f}_i)\|_2 = 1 \quad \forall i, \\ &\qquad \nabla g_k(\mathbf{f}_i) \cdot \nabla g_j(\mathbf{f}_i) = 0 \quad \forall i, j < k, \end{aligned} \tag{2}$$

where we constrain the magnitude of the gradient to 1 to prevent g_k from arbitrarily compressing its output and minimizing the loss artificially.

Once $G = \{g_1, \ldots, g_K\}$ is established, any test vector $\mathbf{f}$ can be scored by computing the ratios between the test error and the average training error,

$$s^2(\mathbf{f}) = \sum_k \frac{g_k(\mathbf{f})^2}{e_k}, \tag{3}$$

where e_k is the training MSE of the soft invariant g_k,

$$e_k = \frac{1}{N} \sum_i g_k(\mathbf{f}_i)^2. \tag{4}$$

Intuitively, tight invariants with low e_k values will have a high influence in the final score, while weak invariants with large e_k values will essentially be ignored. Given that the contribution of weak invariants is negligible in s^2, we can circumvent the problem of setting an optimal number of invariants K and safely set K to the dimensionality of the feature space.

We can further simplify the optimization problem of Eq. (2) by constraining the invariants to the family of affine functions $g_k(\mathbf{f}) = \mathbf{a}_k^T \mathbf{f} + b_k$ with unitary $\mathbf{a}_k$. Under these conditions, Eq. (2) reduces to a PCA problem. Its solution sets $\mathbf{a}_k$ to the k-th smallest principal component and the squared error e_k is set to its corresponding eigenvalue. Moreover, the score function Eq. (3) can be re-written as the square of the Mahalanobis distance using the mean and the covariance of the training feature vectors. Ultimately, computing Mahalanobis distances properly weighs and exploits the linear invariants in the training dataset, which, in turn, suggests that the Mahalanobis distance could lead to good OOD detectors despite its simplicity.

Given that the invariants are computed, in practice, from a collection of feature vectors describing the training set, the performance of an invariant-based U-OOD detection method is contingent on the chosen pre-trained feature extractor. We experimentally found that this is not an important limitation and that general ImageNet-based features lead to descriptive invariants for U-OOD detection even for image modalities that are very different from ImageNet, such as medical images.

3.2 The Mahalanobis anomaly detector

Given the above, we briefly revisit the the Mahalanobis anomaly detector (MahaAD) from Rippel *et al.* [43] as it embodies the invariant feature learning we propose. Fig. 2 illustrates the approach.

MahaAD uses the spatial pooling of the feature maps of a pre-trained CNN to define feature descriptors $\mathbf{f}$. Instead of choosing a specific CNN layer for $\mathbf{f}$, MahaAD works in a multi-layered manner describing each input image $\mathbf{x}$ with a collection of feature vectors $\{\mathbf{f}_\ell(\mathbf{x})\}_{\ell=1}^{L}$ computed at L different layers.

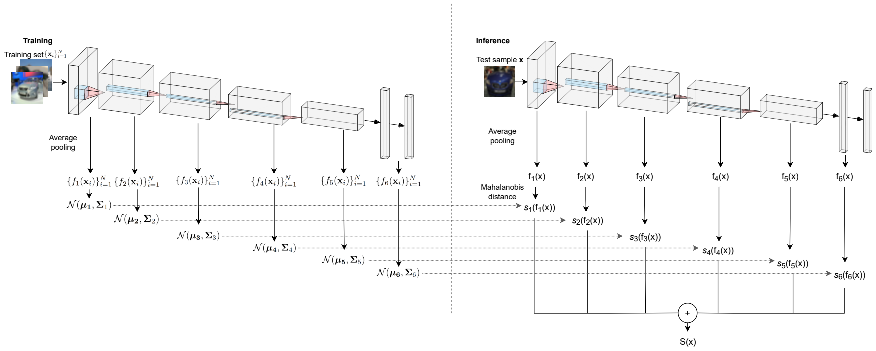

Training
Training set $\{\mathbf{x}_i\}_{i=1}^N$
Average pooling
$\{f_1(\mathbf{x}_i)\}_{i=1}^N$
$\{f_2(\mathbf{x}_i)\}_{i=1}^N$
$\{f_3(\mathbf{x}_i)\}_{i=1}^N$
$\{f_4(\mathbf{x}_i)\}_{i=1}^N$
$\{f_5(\mathbf{x}_i)\}_{i=1}^N$
$\{f_6(\mathbf{x}_i)\}_{i=1}^N$
$\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$
$\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$
$\mathcal{N}(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$
$\mathcal{N}(\boldsymbol{\mu}_4, \boldsymbol{\Sigma}_4)$
$\mathcal{N}(\boldsymbol{\mu}_5, \boldsymbol{\Sigma}_5)$
$\mathcal{N}(\boldsymbol{\mu}_6, \boldsymbol{\Sigma}_6)$
Inference
Test sample **x**
Average pooling
$f_1(x)$
$f_2(x)$
$f_3(x)$
$f_4(x)$
$f_5(x)$
$f_6(x)$
Mahalanobis distance
$s_1(f_1(x))$
$s_2(f_2(x))$
$s_3(f_3(x))$
$s_4(f_4(x))$
$s_5(f_5(x))$
$s_6(f_6(x))$
+
S(x)

Fig. 2: Training and inference stages of the MahaAD method.

At training time, MahaAD computes the mean and the covariance of the descriptor vectors of the images in the training dataset $\{\mathbf{x}_i\}_{i=1}^N$. Specifically, for each layer ℓ, the mean is computed as,

$$\boldsymbol{\mu}_\ell = \frac{1}{N}\sum_{i=1}^{N}\mathbf{f}_\ell(\mathbf{x}_i), \tag{5}$$

while the corresponding covariance matrix is,

$$\boldsymbol{\Sigma}_\ell = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{f}_\ell(\mathbf{x}_i) - \boldsymbol{\mu}_l)(\mathbf{f}_\ell(\mathbf{x}_i) - \boldsymbol{\mu}_l)^\intercal. \tag{6}$$

To avoid singular covariance matrices in high-dimensional or low-data regimes, shrinkage is applied using the standard hyperparameter-free method of [26], although we empirically found that the shrinkage has limited impact on the overall performance of MahaAD. By using multi-layer feature vectors, MahaAD is able to find linear invariants at different image scales.

Importantly, the CNN from which the features are computed is not trained or tuned to the training set whereby making this training phase simple and extremely fast. In practice, it makes the approach more stable and robust across a larger number of datasets. This differs from most recent U-OOD methods that opt to fine-tune their DL models to the training set [40,53,56].

At test time, MahaAD computes the layer-wise Mahalanobis distances between the descriptor vectors of the test image $\mathbf{x}$ and the means $\{\boldsymbol{\mu}_\ell\}_\ell$,

$$s_\ell(\mathbf{f}) = \sqrt{(\mathbf{f} - \boldsymbol{\mu}_\ell)^\intercal \boldsymbol{\Sigma}_\ell^{-1}(\mathbf{f} - \boldsymbol{\mu}_\ell)}, \tag{7}$$

which is equivalent to the square root of Eq. (3). The final OOD score is the sum of the scores over all layers,

$$S(\mathbf{x}) = \sum_{\ell=1}^{L} s_\ell(\mathbf{f}_\ell(\mathbf{x})). \tag{8}$$

4 Experiments

To explore the current state of U-OOD detection, we design a benchmark comparing the performance of several state-of-the-art U-OOD methods over a broad collection of 73 experiments that involve different image modalities, sizes, perturbations, and different criteria for the in- and out-distributions. These experiments aim to identify in what scenarios different methods may be effective and which may not be. Our benchmark is organized in five tasks (see Fig. 3):

Unimodal CIFAR (*uni-class*). Similar to most works [20,21,25,41,48,53,56], we perform 10 experiments using the CIFAR10 dataset, where each experiment takes one of the 10 classes as in-distribution and uses the remaining 9 as OOD. We also use CIFAR100 for 20 experiments, where each of the 20 semantic superclasses of CIFAR100 are used as in-distribution and treat all remaining 19 superclasses as OOD [6,15,41,56].

Unimodal anomaly (*uni-ano*). We use the MVTec dataset [8] which contains 15 classes of images of both normal and defect objects. As in [14,28,41,43,55], we perform one experiment per class, where the defect-free images are used for the in-distribution and defect test images are considered OOD samples.

Unimodal anomaly medical (*uni-med*). We perform 7 experiments with different medical image modalities. The first 2 experiments use optical coherence tomography (OCT) scans and chest X-rays as training in-distributions and corrupted images as OOD samples. The 3$^{\text{rd}}$ experiment trains the models with healthy chest X-rays and uses pathological chest X-rays as OOD. In the remaining 4 experiments, healthy retinal fundus photographs are used for the in-distribution and pathological fundus photographs of four increasing severity levels are used for the OOD images [9,31,36,57].

Low-resolution domain shift (*shift-low-res*). 1 experiment using CIFAR10 as the in-distribution and SVHN as OOD [33,34,53,54,56]. In contrast to previous works, we do not consider CIFAR100 as OOD.

High-resolution domain shift (*shift-high-res*). An extended version of the experiments on the dataset DomainNet presented by Hsu et al. [22]. We run 20 experiments separated into two groups: 10 experiments with `Real-A` as the

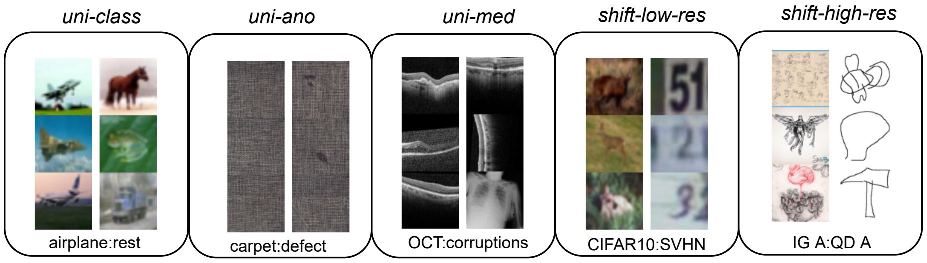


Fig. 3: Example in-distribution and OOD images for each task in our proposed benchmark. IG stands for infograph, QD for quickdraw.

in-distribution and 10 experiments with Infograph-A as the in-distribution. We avoid using Real-B and Infograph-B as OOD in the first and the second group of experiments respectively.

We refer to a specific experiment by the notation in-dataset:out-dataset.

Evaluated methods: We evaluate a selection of top-performing U-OOD detection methods, which we selected if they were among the top-performing methods in at least one of the above tasks. These are: **MSCL** [41], **DN2** [6], **SSD** [53], **MHRot** [20], **DDV** [31], **Glow** [24], **IC** [54], **HierAD**[51] and **CFlow** [17].

All methods are used with their default hyperparameters as given in their official implementations, with (where applicable) the same backbone architecture. More details can be found in the supplementary material. No hyperparameter search was performed, given that no validation metrics exist. Specifically, **Glow**, **HierAD** and **IC** models are based on the same Glow network. All other methods use a ResNet-101. All methods resize the input images to 224 × 224, with the exception of the Glow-based and **SSD** models, where we found that resizing images to 32 × 32 worked better.

Additionally, we show results when using **MahaAD** with an EfficientNet-b4. As **MahaAD** requires no neural network training and thus, unlike all other methods except **DN2**, the additional computational cost for this is minimal. Note that using a pre-trained CNN to extract image features is not a major limitation in practice, as all standard deep learning libraries offer tools to load and use such models in very few steps.

4.1 Results

The evaluated methods were compared in terms of performance, training times and training complexity. We detail the results of our experiments below and provide a deeper breakdown of the results, including additional methods, in the supplementary material.

Performance. Most methods were inconsistent across different tasks (see Table 1). **MSCL**, which performed very well in *uni-class*, is challenged in *uni-ano* and in *shift-low-res*. Conversely, **CFlow**'s performance is high for *uni-ano*, but heavily drops in *uni-class* and especially *shift-low-res*. **SSD** had the best results on *shift-low-res* but struggled with tasks involving high-dimensional images, and **DN2** scored very well on average except on *shift-low-res*. On the other hand, **MahaAD** performed very well and with high stability across tasks. Specifically, it performed among the top three methods in all tasks but in the low resolution domain shift task, for which it still beats **MSCL**, **DN2** and **DDV** by large margins. Furthermore, **MahaAD** was the best performing method on average, beating the second-best method, **MSCL**, by more than 2 percent points across tasks. These performance instabilities were not only observed across the different tasks reported in Table 1, but also within the tasks with fixed in-distribution across different OOD datasets. For example, for the *shift-high-res* task, performance of most methods fluctuated depending on the chosen OOD dataset (see Fig. 4). In contrast, **MahaAD** again is the only method that stands out in terms

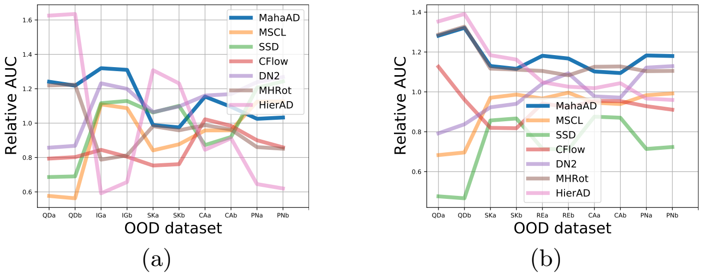
Table 1: Performance summary in AUC over three runs on our U-OOD benchmark. We report performances for each task, as well as the mean over tasks and over experiments. No standard deviation is reported for **MahaAD** and **DN2** as they are deterministic. (*) Taken from original publication; ($^+$) taken from [41]; ($^-$) taken from [56]; (†) taken from [51]

Method	Architecture	*uni-class*	*uni-ano*	*uni-med*	*shift-low-res*	*shift-high-res*	Task Mean	Experiment Mean
Glow	K = 32, L = 3	$53.8_{\pm 0.1}$	$82.0_{\pm 2.5}$	$55.8_{\pm 0.8}$	$8.8^{\dagger}$	$34.5_{\pm 0.1}$	47.0	53.9
IC	K = 32, L = 3	$55.7_{\pm 0.1}$	$73.6_{\pm 2.6}$	$65.1_{\pm 0.5}$	$95.0^{\dagger}$	$65.8_{\pm 0.1}$	71.0	63.6
HierAD	K = 32, L = 3	$63.0_{\pm 0.4}$	$81.6_{\pm 2.1}$	$72.5_{\pm 0.6}$	93.9^{*}	$75.0_{\pm 0.3}$	77.2	71.4
MHRot	ResNet-101	83.4^{+}	$70.8_{\pm 1.0}$	$69.0_{\pm 0.7}$	97.8^{-}	$73.3_{\pm 0.9}$	78.9	76.9
DDV	ResNet-101	$65.8_{\pm 1.4}$	$65.5_{\pm 0.2}$	$60.3_{\pm 3.2}$	$47.9_{\pm 6.6}$	$63.9_{\pm 4.9}$	60.7	64.5
MSCL	ResNet-101	$\mathbf{96.3_{\pm 0.0}}$	$86.4_{\pm 0.0}$	$75.2_{\pm 0.1}$	$88.3_{\pm 0.0}$	$74.4_{\pm 0.0}$	84.1	86.1
CFlow	ResNet-101	$75.0_{\pm 0.0}$	$\mathbf{95.7_{\pm 0.1}}$	$68.8_{\pm 0.3}$	$6.6_{\pm 0.2}$	$61.8_{\pm 0.3}$	61.6	74.1
DN2	ResNet-101	91.2	86.2	**76.7**	57.4	76.0	77.5	84.1
SSD	ResNet-101	$83.6_{\pm 0.3}$	$65.8_{\pm 3.0}$	$64.6_{\pm 0.6}$	$\mathbf{99.6}^{*}$	$60.4_{\pm 0.9}$	74.8	72.0
MahaAD	ResNet-101	92.4	91.3	75.7	94.3	**78.6**	**86.5**	**86.8**
MahaAD	EfficientNet-b4	95.1	94.4	76.8	96.2	83.8	89.3	90.1

of stability, as it performs well regardless of the in and the out datasets selected.

Training times. **MahaAD** was faster to train than its counterparts (see Fig. 5). For example, in the CIFAR10:SVHN experiment (task *shift-low-res*), using two GeForce RTX 3090s, **MahaAD** was the fastest to train, taking roughly 90 seconds to process the entire CIFAR10 dataset. Other methods with similar performances were orders of magnitude slower: **MSCL** took more than half an hour for airplane:rest and **SSD** took more than 12 hours for CIFAR10:SVHN. In addition, no method performed consistently better than **MahaAD** on either of these two experiments. This behavior was also observed for the other tasks.

Training complexity. Furthermore, **MahaAD** was simpler to train, with fewer hyperparameters and more predictable behavior. Predicting the conver-



(a) (b)

Fig. 4: Relative performance (AUC divided by mean AUC on that task) for seven methods on the *shift-high-res* tasks. X-axis indicates the out distribution. (a) `Real-A` as in-distribution. (b) `Infograph-A` as in-distribution.

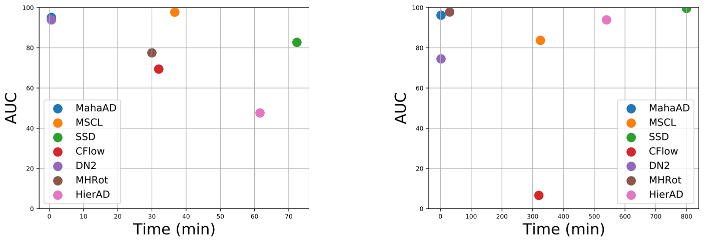
gence of methods such as **MSCL**, **CFlow** and **DDV** was challenging as there is no apparent correlation between the training loss and OOD performance, as also reported in [41]. It is thus unclear when to stop training before the performance starts degrading. While this lack of obvious stopping criterion is problematic for many methods [37,40,41,45], **MahaAD** is convenient as it avoids this necessity altogether.

4.2 Importance of data invariants

We report here additional results that support the importance of data invariants, both for the quality of U-OOD detection and as a tool to analyse U-OOD predictions and evaluation datasets.

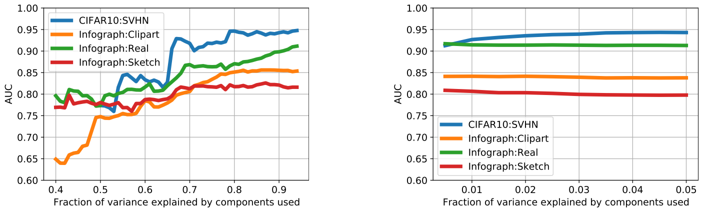
In order to assess the importance of data invariants for U-OOD detection, we examined which principal components are most effective at identifying OOD samples. To that end, we measured the AUC score in four experiments by limiting the Mahalanobis score of Eq. (3) to only use the subset of principal components with highest variance, corresponding to the modes of variation of the data. Similarly, we observed the performance with the subset of principal components with the smallest variance, corresponding to data invariants. The latter outperformed the former by a large margin in U-OOD detection (see Fig. 6). Starting from the most variant principal components, the performance slowly increases when adding more components, converging when over 80% of the variance is explained. On the other hand, when starting from the most invariant component, the performance quickly converges when as little as 3% of the variance has been explained, supporting the idea that invariants are more representative to characterize training data and OOD samples. While other works had observed similar findings, they either consider the supervised case [23,39], or frame it in the context of reducing dimensionality [43].

An interesting consequence of our invariant-based interpretation of U-OOD is that, when we considered what to include in our benchmark, some experiments that are valid for evaluating supervised OOD methods, are in fact not suitable for the U-OOD case. For instance, CIFAR10:CIFAR100 [33,34,53] or 9-classes:1-



(a) (b)

Fig. 5: Training times and performances for different methods on (a) *uni-class*'s `airplane:rest` and (b) *shift-low-res*'s `CIFAR10:SVHN`.



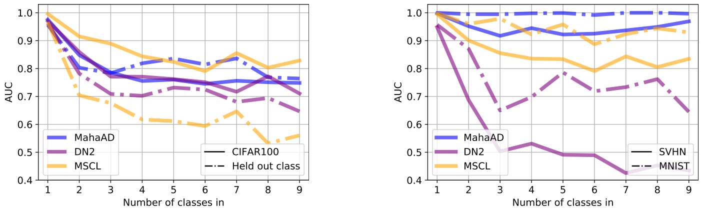
(a) (b)

Fig. 6: The OOD AUC for four experiments with a ResNet50 using different sets of principal components. (a) gives result starting from the first principal components, while (b) does so from the last principal components. The x-axis of (a) starts at 0.4 as for Infograph the most variant component of the first layer is responsible for almost 40% of all variance.

held-out-class of CIFAR10 [3,6] were used in previous U-OOD works even though they do not appear to meet the U-OOD criteria.

More specifically, according to our definition for U-OOD, one would expect that by increasing the number of classes present in a training set, the invariants associated to the high semantic features will decrease, effectively reducing the probability that new classes are considered U-OOD. For example, training with multiple classes from CIFAR10 (*e.g.*, cats, dogs, cars) reduces the probability that an additional class (*e.g.*, plane) from CIFAR10 or CIFAR100 should be considered OOD (Fig. 1(d)), as the class stops being an invariant. However, the number of training classes should not affect the probability that images from a different modality are detected as OOD, as they break other kind of invariants. For instance, when training with images from CIFAR10, the test images from SVHN or MNIST should still be considered OOD regardless of the number of CIFAR10 training classes, as they are clearly distinct in appearance.

In Fig. 7, we investigate this desired behavior experimentally by analyzing the performance of the three best-performing methods when we increased the number of in-distribution CIFAR10 training classes. As expected, all methods consider fewer images from CIFAR100 and one held-out class from CIFAR10 as OOD when the number of training classes increased (Fig. 7(a)). Conversely, increasing the number of training CIFAR10 classes did not affect the predictions for SVHN and MNIST with **MahaAD**, which correctly kept detecting both datasets as OOD (Fig. 7(b)). In contrast, this did negatively affect the predictions of **DN2** and **MSCL**. According to our invariant-based interpretation of U-OOD, **MahaAD**'s behavior is reasonable and consistent in these configurations, yet the unexpected **DN2** and **MSCL** results are hard to justify. To the extent of our knowledge, no previous work on U-OOD detection had provided a similar theoretical tool capable of interpreting and explaining results.



(a) (b)

Fig. 7: OOD AUC performance for different methods as a function of the number of classes in the training set (CIFAR10), keeping its size constant. (a) Performance on CIFAR100 and a held out CIFAR10 class as out-distribution, which should not be considered U-OOD under our interpretation. (b) Performance on SVHN and a held out MNIST class as out-distribution, which should be considered U-OOD under our interpretation.

5 Conclusion

Our work explores the state of U-OOD detection by observing the behavior of methods on an extensive and varied set of tasks. By doing so, we show a complicated landscape, with most methods being highly inconsistent among and within tasks. **MahaAD** is however an exception to this trend, behaving consistently in a large majority of experimental configurations. Despite being neglected in most recent U-OOD papers, **MahaAD** appears to be the current best off-the-shelf unsupervised OOD detector, as it offers good performance and consistency without requiring time-consuming data pre-processing, careful tuning of the training procedure, or hyperparameter search.

In order to explain these inconsistent results, we introduced a characterization of U-OOD based on training set invariants and showed that the **MahaAD** method embodies a linear version of this concept. We found this framework and the proposed benchmark to be useful to not only qualitatively understand U-OOD detector predictions, but also to assess whether a test dataset is in fact suitable for U-OOD evaluation or not. A key take-away is that we cannot purely rely on semantic labels from datasets to design U-OOD evaluation methods, as done in previous works.

In general, this points to a rather bleak conclusion: at the moment, no method can consistently outperform a simple anomaly detector that uses naively extracted features from a network trained on a different dataset that was optimized for a different task. We believe that with our invariant-based U-OOD characterization, new appropriate methods can be designed and validated in comprehensive ways.

Acknowledgements

This work was funded by the Swiss National Science Foundation (SNSF), research grant 200021_192285 “Image data validation for AI systems”.

References

1. Abati, D., Porrello, A., Calderara, S., Cucchiara, R.: Latent space autoregression for novelty detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 481–490 (2019)
2. Ahmed, F., Courville, A.: Detecting semantic anomalies. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 3154–3162 (2020)
3. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Asian conference on computer vision. pp. 622–637. Springer (2018)
4. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in ai safety. arXiv preprint arXiv:1606.06565 (2016)
5. Battikh, M.S., Lenskiy, A.A.: Latent-insensitive autoencoders for anomaly detection and class-incremental learning. arXiv preprint arXiv:2110.13101 (2021)
6. Bergman, L., Cohen, N., Hoshen, Y.: Deep nearest neighbor anomaly detection. arXiv preprint arXiv:2002.10445 (2020)
7. Bergman, L., Hoshen, Y.: Classification-based anomaly detection for general data. International Conference on Learning Representations (2020)
8. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Mvtec ad – a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (6 2019)
9. Bozorgtabar, B., Mahapatra, D., Vray, G., Thiran, J.P.: Salad: Self-supervised aggregation learning for anomaly detection on x-rays. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 468–478. Springer (2020)
10. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. pp. 93–104 (2000)
11. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407 (2019)
12. Choi, H., Jang, E., Alemi, A.A.: Waic, but why? generative ensembles for robust anomaly detection. arXiv preprint arXiv:1810.01392 (2018)
13. Choi, S., Chung, S.Y.: Novelty detection via blurring. International Conference on Learning Representations (2020)
14. Defard, T., Setkov, A., Loesch, A., Audigier, R.: Padim: a patch distribution modeling framework for anomaly detection and localization. In: International Conference on Pattern Recognition. pp. 475–489. Springer (2021)
15. Golan, I., El-Yaniv, R.: Deep anomaly detection using geometric transformations. In: Advances in Neural Information Processing Systems. pp. 9758–9769 (2018)
16. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1705–1714 (2019)
17. Gudovskiy, D., Ishizaka, S., Kozuka, K.: Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 98–107 (2022)
18. Havtorn, J.D.D., Frellsen, J., Hauberg, S., Maaløe, L.: Hierarchical vaes know what they don't know. In: International Conference on Machine Learning. pp. 4117–4128. PMLR (2021)

19. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. International Conference on Learning Representations (2017)
20. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. Advances in Neural Information Processing Systems **32** (2019)
21. Hou, J., Zhang, Y., Zhong, Q., Xie, D., Pu, S., Zhou, H.: Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8791–8800 (2021)
22. Hsu, Y.C., Shen, Y., Jin, H., Kira, Z.: Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10951–10960 (2020)
23. Kamoi, R., Kobayashi, K.: Why is the mahalanobis distance effective for anomaly detection? arXiv preprint arXiv:2003.00402 (2020)
24. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018)
25. Koner, R., Sinhamahapatra, P., Roscher, K., Günnemann, S., Tresp, V.: Oodformer: Out-of-distribution detection transformer. arXiv preprint arXiv:2107.08976 (2021)
26. Ledoit, O., Wolf, M.: A well-conditioned estimator for large-dimensional covariance matrices. Journal of multivariate analysis **88**(2), 365–411 (2004)
27. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. Advances in Neural Information Processing Systems **31**, 7167–7177 (2018)
28. Li, C.L., Sohn, K., Yoon, J., Pfister, T.: Cutpaste: Self-supervised learning for anomaly detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9664–9674 (2021)
29. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. International Conference on Learning Representations (2018)
30. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422. IEEE (2008)
31. Márquez-Neila, P., Sznitman, R.: Image data validation for medical systems. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 329–337. Springer (2019)
32. Mesarcik, M., Ranguelova, E., Boonstra, A.J., van Nieuwpoort, R.V.: Improving novelty detection using the reconstructions of nearest neighbours. arXiv preprint arXiv:2111.06150 (2021)
33. Mohseni, S., Vahdat, A., Yadawa, J.: Multi-task transformation learning for robust out-of-distribution detection. arXiv preprint arXiv:2106.03899 (2021)
34. Morningstar, W., Ham, C., Gallagher, A., Lakshminarayanan, B., Alemi, A., Dillon, J.: Density of states estimation for out of distribution detection. In: International Conference on Artificial Intelligence and Statistics. pp. 3232–3240. PMLR (2021)
35. Nalisnick, E., Matsukawa, A., Teh, Y.W., Gorur, D., Lakshminarayanan, B.: Do deep generative models know what they don't know? arXiv preprint arXiv:1810.09136 (2018)

36. Ouardini, K., Yang, H., Unnikrishnan, B., Romain, M., Garcin, C., Zenati, H., Campbell, J.P., Chiang, M.F., Kalpathy-Cramer, J., Chandrasekhar, V., et al.: Towards practical unsupervised anomaly detection on retinal images. In: Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data, pp. 225–234. Springer (2019)
37. Perera, P., Nallapati, R., Xiang, B.: Ocgan: One-class novelty detection using gans with constrained latent representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2898–2906 (2019)
38. Perera, P., Oza, P., Patel, V.M.: One-class classification: A survey. arXiv preprint arXiv:2101.03064 (2021)
39. Podolskiy, A., Lipin, D., Bout, A., Artemova, E., Piontkovskaya, I.: Revisiting mahalanobis distance for transformer-based out-of-domain detection. arXiv preprint arXiv:2101.03778 (2021)
40. Reiss, T., Cohen, N., Bergman, L., Hoshen, Y.: Panda: Adapting pretrained features for anomaly detection and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2806–2814 (2021)
41. Reiss, T., Hoshen, Y.: Mean-shifted contrastive loss for anomaly detection. arXiv preprint arXiv:2106.03844 (2021)
42. Ren, J., Liu, P.J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., Lakshminarayanan, B.: Likelihood ratios for out-of-distribution detection. In: Advances in Neural Information Processing Systems. pp. 14707–14718 (2019)
43. Rippel, O., Mertens, P., Merhof, D.: Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 6726–6733. IEEE (2021)
44. Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W., Kloft, M., Dietterich, T.G., Müller, K.R.: A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE (2021)
45. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International conference on machine learning. pp. 4393–4402. PMLR (2018)
46. Sakurada, M., Yairi, T.: Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis. pp. 4–11 (2014)
47. Salehi, M., Eftekhar, A., Sadjadi, N., Rohban, M.H., Rabiee, H.R.: Puzzle-ae: Novelty detection in images through solving puzzles. arXiv preprint arXiv:2008.12959 (2020)
48. Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M.H., Rabiee, H.R.: Multiresolution knowledge distillation for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14902–14912 (2021)
49. Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. International Conference on Learning Representations (2017)
50. Sastry, C.S., Oore, S.: Detecting out-of-distribution examples with gram matrices. In: International Conference on Machine Learning. pp. 8491–8501. PMLR (2020)
51. Schirrmeister, R., Zhou, Y., Ball, T., Zhang, D.: Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. Advances in Neural Information Processing Systems **33**, 21038–21049 (2020)
52. Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C., et al.: Support vector method for novelty detection. In: NIPS. vol. 12, pp. 582–588. Citeseer (1999)

53. Sehwag, V., Chiang, M., Mittal, P.: Ssd: A unified framework for self-supervised outlier detection. International Conference on Learning Representations (2021)
54. Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J.F., Luque, J.: Input complexity and out-of-distribution detection with likelihood-based generative models. International Conference on Learning Representations (2019)
55. Sohn, K., Li, C.L., Yoon, J., Jin, M., Pfister, T.: Learning and evaluating representations for deep one-class classification. International Conference on Learning Representations (2021)
56. Tack, J., Mo, S., Jeong, J., Shin, J.: Csi: Novelty detection via contrastive learning on distributionally shifted instances. Advances in neural information processing systems **33**, 11839–11852 (2020)
57. Tang, Y.X., Tang, Y.B., Han, M., Xiao, J., Summers, R.M.: Abnormal chest x-ray identification with generative adversarial one-class classifier. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). pp. 1358–1361. IEEE (2019)
58. Wang, S., Zeng, Y., Liu, X., Zhu, E., Yin, J., Xu, C., Kloft, M.: Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. In: Advances in Neural Information Processing Systems. pp. 5962–5975 (2019)
59. Xiao, Z., Yan, Q., Amit, Y.: Do we really need to learn representations from in-domain data for outlier detection? ICML 2021 Workshop on Uncertainty & Robustness in Deep Learning (2021)
60. Yang, J., Zhou, K., Li, Y., Liu, Z.: Generalized out-of-distribution detection: A survey. arXiv preprint arXiv:2110.11334 (2021)
61. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International conference on learning representations (2018)