Sample-free white-box out-of-distribution detection for deep learning

Jean-Michel Begon Pierre Geurts
University of Liege
{jm.begon,p.geurts}@uliege.be

Abstract

Being able to detect irrelevant test examples with respect to deployed deep learning models is paramount to properly and safely using them. In this paper, we address the problem of rejecting such out-of-distribution (OOD) samples in a fully sample-free way, i.e., without requiring any access to in-distribution or OOD samples. We propose several indicators which can be computed alongside the prediction with little additional cost, assuming white-box access to the network. These indicators prove useful, stable and complementary for OOD detection on frequently-used architectures. We also introduce a surprisingly simple, yet effective summary OOD indicator. This indicator is shown to perform well across several networks and datasets and can furthermore be easily tuned as soon as samples become available. Lastly, we discuss how to exploit this summary in real-world settings.

1. Out-of-distribution sample detection

Imagine you are running some medical tests to determine whether you have cancer or not, but erroneous data are fed to the machine learning (ML) model in charge of establishing the diagnosis. Would you prefer to get a positive or a negative answer? Or would you rather the model refrained from making a prediction and instead alerted the operator?

Out-of-distribution (OOD) detection [15] precisely aims at detecting samples which come from a different distribution than the one used to train the model. There are many reasons why a model would be fed such OOD inputs: faulty equipment, user mistake, malicious intent, etc. Whether intentional or not, not being able for a ML model to detect when it receives an OOD sample raises legitimate concerns about the reliability of systems built from such model, especially in critical applications.

The OOD sample-free setting. As more and more ML models are being deployed, addressing this issue becomes a very pressing matter. Unfortunately, such concerns might not have been anticipated at training time. This tends to be the standard at the moment, with models released "as is". As a consequence, enforcing trustworthiness must be done at a further stage, incurring the risk of the original, in-distribution (ID) data not being available. Reasons for this include privacy constraints (*e.g.*, with medical or personal data), the overly large size of the dataset, which prevents easy distribution, the reluctance of companies to share their data with competitors, or simply data loss, either due to carelessness or storage constraints.

Although restrictive, the sample-free setting is worth investigating for several additional reasons: (i) it will provide, by construction, data-efficient solutions, (ii) it is relevant for other data-free paradigms, such as zero-shot distillation (*e.g.* [4, 5, 6, 25]), and (iii) analyzing its limits will allow to understand how much information about the ID distribution is buried within a network trained in a standard way.

Goal. Under such circumstances, the general question we want to address in this paper is whether it is possible to extract from a pre-trained model one or several indicators that allow to distinguish between OOD and ID samples, in a fully sample-free setting, *i.e.*, with no data. We assume a white-box access to the networks, *i.e.*, their exact structures and parameters are fully accessible. We can thus extract information/statistics as the sample is run through the network.

On the other hand, the sample-free setting prevents from learning new models, modifying existing ones (as alteration cannot be assessed and would potentially be harmful), or fine-tuning hyper-parameters.

Although the question we tackle is generic, we focus on the problem of image classification using modern, frequently-used deep neural network architectures. We will naturally favor indicators which are fast to compute.

Contributions. Our fourth-fold constribution is:

- we introduce the sample-free setting;
- we review which prior works provide indicators falling into our setting and propose/adapt several new ones (Section 2);
- we conduct an extensive empirical analysis of these indicators on classical benchmark datasets (Section 3);

- we introduce a summary indicator to serve as an ultimate criterion for OOD detection, which is shown to perform well in comparison to the individual indicators (Section 4);

Outline. We first formalize the problem in Section 1.1 and discuss some related works in Section 1.2. The proposed indicators are described in Section 2 and are empirically studied in Section 3. A summary indicator is proposed and studied in Section 4. Finally, we discuss how to use the proposed indicators in several real-word settings in Section 5 before concluding in Section 6.

1.1. Problem definition

We consider a neural network composed of L layers:

$$z(.,\Theta) = f_L(\cdot;\theta_L) \circ \ldots \circ f_1(\cdot;\theta_1) : X \to \mathcal{L} \quad (1)$$

mapping from the input space $X \subset \mathbb{R}^p$ to the *logit* space $\mathcal{L} = \mathbb{R}^K$ (assuming K classes), with $\Theta = [\theta_1, \ldots, \theta_L]$ the set of all trainable network parameters. We will denote below by $z^{(l)}(x;\Theta_l)$ the feature vector of layer l ($1 \leq l \leq L$) for an input x:

$$z^{(l)}(x;\Theta_l) = (f_l(\cdot;\theta_l) \circ \ldots \circ f_1(\cdot;\theta_1))(x), \quad (2)$$

with $\Theta_l = [\theta_1, \ldots, \theta_l]$ the parameters of the first l layers. We further assume that Θ has been optimized on some so-called "in" distribution $\mathcal{I}$ so that:

$$\Theta \approx \min_{\Theta'} \mathbb{E}_{x,y\sim\mathcal{I}} \left(E(x,y;\Theta') + \mu R(\Theta') \right) \quad (3)$$

where μ weighs the two components of the loss, the regularization function R is typically the weight decay and the loss function E is usually the cross-entropy on the softmax logits in the case of classification:

$$E(x,y;\Theta) = -\sum_j^K y_j \log p_j(x;\Theta) \quad (4)$$

$$p_j(x;\Theta) = \frac{e^{z_j(x;\Theta)}}{\sum_{k=1}^K e^{z_k(x;\Theta)}} \quad (5)$$

We assume that at test time samples that are sent through the network comes from a mixture of two distributions: the in-distribution $\mathcal{I}$ and another different distribution $\mathcal{A}$, called the OOD distribution. Our goal in this paper is to construct a function $h : X \to \mathbb{R}$, called an indicator, that allows to discriminate as well as possible ID from OOD samples, with respect to the given neural network. We will design indicators that take low values for ID samples and large values for OOD samples. In practice, a test example x can thus be rejected as soon as $h(x) > h_{th}$, where h_{th} is a threshold that can be set to minimize a given error type, taking into account the needs of the application. Denoting by $D_{\mathcal{A}}(x)$ and $D_{\mathcal{I}}(x)$ the density at x for distributions $\mathcal{A}$ and $\mathcal{I}$ respectively, an ideal indicator function is therefore $h(x) = \frac{D_{\mathcal{A}}(x)}{D_{\mathcal{I}}(x)}$, that allows to implement a bayes optimal discriminator of ID and OOD samples.

In this paper, we adopt a sample-free setting, where $\mathcal{A}$ is a priori unknown and no samples from $\mathcal{I}$ (or $\mathcal{A}$) are available. We however assume a white-box access to the neural network, which allows us to investigate candidate indicator functions of the following general form:

$$h(x) = H\left(x, \Theta, z^{(1)}(x;\Theta_1), \ldots, z^{(L)}(x;\Theta_L)\right). \quad (6)$$

Indicators can thus be defined from features computed anywhere in the network, as well as from network parameters. Given that $\mathcal{A}$ is unknown, our main incentive will be to craft h functions such that $h(x)$ is low for $x \sim I$, mainly by taking into account the way the neural network was trained.

1.2. Related work

OOD detection methods can be categorized based on what data they rely on and how they impact the base model. We restrict the discussion below to methods which do not require to learn a model for the base task from scratch.

Early methods. [15] coined the term out-of-distribution while proposing to use the maximum softmax probability as an indicator of OODness. ODIN [24] is a popular alternative where samples are adversarially perturbed and the softmax is taken with a high temperature.

Model alteration. Several methods sacrifice some accuracy to better detect OOD samples. They usually lower the network confidence, known to be unreasonably high [27], by adding some regularization so that OOD samples are better captured. They rely either only on ID data [2, 10, 11, 22, 35] or on both ID and OOD data [16, 31, 32].

Supervised approaches. Among the methods which do not alter the network, some cast the problem as a supervised binary classification problem [3, 29, 34]. This setting makes sense, for instance, if one want to discard samples with unusual lighting conditions at inference time (because the model was not built with such robustness at training time). In general, however, it is very difficult to predict the exact nature of the OOD samples as they are expected to be the result of intrinsically unpredictable phenomena such as human mistakes or faulty equipment. As a consequence, despite being efficient (as shown later), these approaches raise the concern of the adequacy between the OOD training and testing sets, which is hard to resolve outside of a clear application domain [34].

Outlier/novelty detection. In the absence of a clear target OODistribution, the problem is better cast as outlier detection, *i.e.* detecting sample of low density. Many methods fall into this category, relying on the availability of ID data, possibly slightly polluted with OOD ones [1, 7, 12, 15, 17, 23, 30, 33, 36, 37].

Sample-free approaches Only the earliest work, the maximum softmax probability of [15] and ODIN [24] (provided we use the default hyper-parameters), match our setting, where (i) the model cannot be altered, and (ii) no ID data is available. These two indicators will be discussed later and included in our empirical comparison.

2. Sample-free white-box OOD indicators

In this section, we introduce a number of indicators for OOD detection. An indicator assesses how unlikely it is for a sample to be ID (Section 1.1). We describe two categories of such indicators: (i) optimality-based indicators (Section 2.1), and (ii) batch-normalization-based indicators (Section 2.2). Note that we will conform to the notations of Section 1.1, dropping the dependency to x and Θ when there is no ambiguity.

2.1. Optimality-based indicators

Hopefully, a deployed network should be well trained, resulting in ID samples having a small loss gradient with high probability. This happens when $p_j(x) \approx y_j \ (1 \leq j \leq K)$, which allows us to derive several indicators for which we expect the values on ID samples to be low (see A.1 for a more detailed discussion of the optimality consequences). Note that this is also the motivation behind other, non-necessarily sample-free methods (such as [15, 17, 23, 24]).

Baselines. Two common baseline indicators which derive directly from the optimality condition are

$$\text{MP}(x) = 1 - \max_{1 \leq j \leq K} p_j(x) \tag{7}$$

$$\text{H}(x) = -\sum_{j=1}^{K} p_j(x) \log p_j(x) \tag{8}$$

Using the maximum probability was proposed by [15] when introducing the topic of OOD detection. When the probabilities given by the network for the minority classes are uniform and close to zero, the entropy H should behave like MP. The entropy might convey a little more information than the maximum probability when the uniformity constraint is not satisfied.

ODIN. ODIN was introduced by [24] and is quite popular in the OOD context. It relies on two ideas. First, some adversarial noise [13] is added to the input x. Then, the softmax probability vector given by the network is computed using a temperature T of 1000 in the softmax:

$$x' = x + \epsilon \ \text{sign}\left(\nabla_x E(x, p(x))\right) \tag{9}$$

$$p_{j|T}(x) = \frac{e^{z_j(x)/T}}{\sum_{k=1}^{K} e^{z_k(x)/T}} \tag{10}$$

$$\text{T1000}(x) = 1 - \max_{1 \leq j \leq K} p_{j|T=1000}(x) \tag{11}$$

$$\text{ODIN}(x) = \text{T1000}(x') \tag{12}$$

The rationale is that the adversarial perturbation will have different effects on ID/OOD samples. Additionally, if we let k be the class predicted by the network for a given x, it can be shown (Appendix A.2), that

$$p_{k|T} \approx \frac{c}{K} + \frac{1}{TK} z_k \tag{13}$$

so long as $z_k \ll T$. As such, using T1000 is a way of normalizing the logit of the predicted class in the range $0 \ll \text{T1000}(x) \leq 1 - 1/K$.

When $\epsilon = 0$, ODIN reduces to T1000 and the expensive cost of computing the adversarial perturbation is avoided. Tuning ϵ in a sample-free setting is not trivial. Arguably though, the magnitude of the perturbation might not vary much due to the sign function. In any case, we will use the default value of the noise magnitude proposed in the original paper ($\epsilon = 8 \times 10^{-4}$). Considering it was established on CIFAR 10(0) as well, it should constitute a strong baseline anyway.

Latent space indicators. Let $u = z^{(L-1)}$ be the latent pre-linear vector and $z = Wu + b$ be the logit vector, with $\theta_{L-1} = [W, b]$.

In order for the loss gradient to be small, $w_k^T u + b_k$ (where k is the predicted class at x) must be high. Since $w_k^T u + b_k = ||w_k||\,||u|| \cos \alpha_{u,k} + b_k$, this suggests the following two necessary conditions:

1. $||u||$ is high;
2. $\cos \alpha_{u,k}$ is close to 1.

From them, we can derive the following indicators:

$$\text{NORM}(x) = -||u|| \tag{14}$$

$$\text{ANG}(x) = 1 - \cos \alpha_{u,k} = 1 - \frac{w_k^T u}{||w_k||\,||u||} \tag{15}$$

$$\text{PROJ}(x) = -||u|| \cos \alpha_{u,k} = -\frac{w_k^T u}{||w_k||} \tag{16}$$

$$\text{ACT}(x) = -w_k^T u \tag{17}$$

The NORM indicator should not be sufficient by itself, as a high norm possibly benefits all the logits. ANG stands

for angularity and is the cosine distance between u and w_k. Compared to the logit (close to ACT), it will favor more samples which align well with the hyperplanes and will favor less samples which just have a high latent norm. The PROJ indicator combines the information from both ANG and NORM. Therefore PROJ is expected to be closely related to the logit.

Positivity. ReLU-based architectures, which include most modern ones in image classification, end the feature extraction phase with a ReLU activation, possibly followed by max or average pooling. As a result, the latent vectors are non-negative, whereas most components of the hyperplane weights are negative (see Appendix A.1) and are used to bid *against* the other classes, rather than *for* the predicted one. This suggests that it might be worth looking at the positive and negative parts of the previous indicators separately.

We define three new indicators NORM+, ANG++ and ACT+ that are obtained by reducing the vectors w_k and u to the components with positive weights in w_k in the definitions of NORM (Eq. 14), ANG (Eq. 15), and ACT (Eq. 17) respectively. In other words, we only consider the positive subspace of w_k.

2.2. Batchnorm-based indicators

Beyond optimality conditions, the presence of batch normalization layers [19] offers the opportunity to define additional indicators. Indeed, those layers are based on statistical parameters directly estimated on the training data, promising a direct route to ID statistical information.

Using batch-normalization-derived features for OOD detection has been proposed by [29], however in the context of one-class and supervised OOD detection. Here we propose *indicators* based on them. Such features also tend to be used more and more in the context of data-free compression [5, 39], which basically relies on the definition of OOD losses.

The batch normalization layer operates in two steps. First, it standardizes the input batch with respect to some estimated statistics (Eq. 18). Then it applies a linear transformation (Eq. 19). Let B be the set of layer indices corresponding to the batchnorm layers. Then for all $l \in B$,

$$y^{(l)}_{c,w,h} = \frac{z^{(l-1)}_{c,w,h} - \mu^{(l)}_c}{\sqrt{\left(\left[\sigma^{(l)}_c\right]^2 + \epsilon\right)}} \quad \forall c, w, h \tag{18}$$

$$z^{(l)}_{c,w,h} = y^{(l)}_{c,w,h} \times \gamma^{(l)}_c + \beta^{(l)}_c \quad \forall c, w, h \tag{19}$$

with C_l ($1 \leq c \leq C_l$), W_l ($1 \leq c \leq W_l$) and H_l ($1 \leq c \leq H_l$), standing respectively for the number of channels, the width and the height of the input tensors at layer l.

Since the $\mu^{(l)}_c$ and $\sigma^{(l)}_c$ parameters are estimated during training and are specific to ID samples, we can hope to use them for OOD rejection. More precisely, defining,

$$M^{(l)}_c = \frac{1}{H_l \times W_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} y^{(l)}_{c,w,h} \tag{20}$$

$$S^{(l)}_c = \sqrt{\frac{1}{(H_l \times W_l) - 1} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} \left[y^{(l)}_{c,w,h} - M^{(l)}_c\right]^2} \tag{21}$$

$$V^{(l)}_c = (H_l \times W_l - 1) \left[S^{(l)}_c\right]^2 \tag{22}$$

we can expect that

$$\mathbb{E}_{\mathcal{I}}\{y^{(l)}_c\} = 0 \tag{23}$$

$$\mathbb{E}_{\mathcal{I}}\{M^{(l)}_c\} = 0 \tag{24}$$

$$\mathbb{E}_{\mathcal{I}}\{S^{(l)}_c\} = 1 \tag{25}$$

$$\mathbb{E}_{\mathcal{I}}\{V^{(l)}_c\} \sim \chi^2_{(H_l \times W_l - 1)} \tag{26}$$

Given these conditions, we propose to derive the following indicators (where $\mathcal{S} \subseteq B$ is a subset of batchnorm layers):

$$\text{DMS}_{\mathcal{S}} = \frac{1}{C_{\mathcal{S}}} \sum_{l \in \mathcal{S}} \sum_{c=1}^{C_l} \left(M^{(l)}_c\right)^2 \tag{27}$$

$$\text{DMS-AOS}_{\mathcal{S}} = \frac{1}{C_{\mathcal{S}}} \sum_{l \in \mathcal{S}} \sum_{c=1}^{C_l} \frac{1}{H_l \times W_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} \left(y^{(l)}_c\right)^2 \tag{28}$$

$$\text{DSS}_{\mathcal{S}} = \frac{1}{C_{\mathcal{S}}} \sum_{l \in \mathcal{S}} \sum_{c=1}^{C_l} \left(S^{(l)}_c - 1\right)^2 \tag{29}$$

$$\text{DSS-EXT}_{\mathcal{S}} = \frac{1}{C_{\mathcal{S}}} \sum_{l \in \mathcal{S}} \sum_{c=1}^{C_l} \mathbb{I}\left[\text{ext}^{(\alpha)}_{\chi^2_{(H_l \times W_l - 1)}}\left(V^{(l)}_c\right)\right] \tag{30}$$

where $C_{\mathcal{S}} = \sum_{l \in \mathcal{S}} C_l$ is the total number of channels in the considered set, and $\text{ext}^{(\alpha)}_{\mathcal{A}}$ is true only when its argument has a (bilateral) p-value according to law $\mathcal{A}$ below the significance level α.

DMS/DSS stands for **d**eparture from the **m**ean/**s**tandard deviation **s**tandardization, and AOS stands for **a**verage **o**f sum. In the remainder of the paper, we take $\alpha = 0.1$ for DSS-EXT.

The intuition behind DMS, DMS-AOS and DSS is that they should produce small values on $\mathcal{I}$. Note however that if $\mathcal{S}$ contains many layers, the value might rise quickly since inter-channel correlations are expected. The intuition behind DSS-EXT is that the variance of $y^{(l)}_{c,w,h}$ should not produce extreme values too often on $\mathcal{I}$.

Relevant subsets. Since the input vectors of the network must also be standardized, we treat the preprocessing as a

Table 1. Summary of sample-free indicators and their bounds, when available.

$0 \leq \text{MP} \leq 1 - 1/K$	$0 \leq \text{ANG++} \leq 1$
$0 \leq \text{H} \leq \log K$	$0 \leq \text{IN-DMS}$
$0 \ll \text{T1000} \leq 1 - 1/K$	$0 \leq \text{IN-DMS-AOS}$
$0 \ll \text{ODIN} \leq 1 - 1/K$	$0 \leq \text{IN-DSS}$
NORM	$0 \leq \text{IN-DSS-EXT} \leq 1$
$0 \leq \text{ANG} \leq 1$	$0 \leq \text{DMS}$
PROJ	$0 \leq \text{DMS-AOS}$
ACT	$0 \leq \text{DSS}$
ACT+	$0 \leq \text{DSS-EXT} \leq 1$

Table 2. Percentiles of the indicator distributions. The indicators were extracted from a DenseNet 121 learned on CIFAR 10.

	CIFAR 10 (test set)			Tiny ImageNet		
	p25	p50	p75	p25	p50	p75
MP	0.000	0.000	0.003	0.014	0.112	0.339
H	0.000	0.004	0.036	0.123	0.628	1.233
T1000	0.898	0.898	0.899	0.899	0.899	0.899
ODIN	0.898	0.898	0.898	0.899	0.899	0.899
NORM	−6.97	−6.34	−5.78	−6.63	−6.02	−5.46
ANG	0.267	0.314	0.391	0.479	0.577	0.657
PROJ	−4.87	−4.28	−3.61	−3.28	−2.55	−1.96
ACT+	−12.9	−11.5	−10.1	−10.2	−8.78	−7.54
ANG+	0.231	0.266	0.319	0.361	0.424	0.477
IN-DMS	0.405	0.677	1.065	0.430	0.735	1.154
DMS	9.377	10.36	11.49	8.722	9.634	10.78
IN-DSS	0.236	0.423	0.669	0.267	0.450	0.670
DSS	7.690	8.116	8.661	7.850	8.385	9.112
1C-Sum	-153.4	-138.5	-123.0	-120.6	-96.9	-79.2

pseudo-batchnorm layer. Some subsets of batchnorm layers might work better than other. However, in the absence of data, we cannot hope to learn which one is the best. In consequence, we propose to focus on two sets: (i) the input pseudo-batchnorm layer, and (ii) all the layers. We denote by the prefix IN- all indicators relating solely on the input normalization so that $\text{IN-DMS} = \text{DMS}_{\{1\}}$. We will refer to those as the IN- indicators. We also drop $\mathcal{S}$ from the notation when $\mathcal{S} = B$.

2.3. Summary

Table 1 summarizes the sample-free indicators and their bounds, when available. All indicators are such that ID samples should portray small values. Although interpretable, probability-based indicators (MP, H, T1000, ODIN) are not necessarily easier to bound (See Table 2 for some statistics about the indicator distributions). Unbounded indicators are *de facto* harder to use in a sample-free setting.

3. Empirical analysis

In this section, we evaluate how the proposed indicators perform individually. After detailing our methodology, we discuss the main results (Section 3.1) and briefly go over some additional findings (Section 3.2).

Table 3. OOD dataset characteristics.

Gaussian	$32 \times 32 \times 3$	$\mu = 0.5, \sigma = 0.25$ clipped on $[0, 1]$
SVHN	$32 \times 32 \times 3$	[26]
MNIST	28×28	[21]
fashion MNIST	28×28	[38]
Tiny ImageNet	$64 \times 64 \times 3$	[9]
LSUN	$256 \times 256 \times 3$	[40][1]
CIFAR 10/100	$32 \times 32 \times 3$	[20]

ID tasks. In order to evaluate the indicator performances, we have trained three networks on three image classification tasks to serve as ID datasets, namely we used CIFAR 10, CIFAR 100 [20] and ImageNet [9].

The networks are a ResNet 50 [14], a WideResNet-40 [41] and a DenseNet 121 [18]. All three architectures are ReLU-based and output non-negative latent vectors. On ImageNet, we used pre-trained networks available in PyTorch [28]. As such, we display only the score on one run. Experiments were all carried out with PyTorch. Overall average accuracy on CIFAR 10 is 94.2, accuracy on CIFAR 100 ranges from 74.2 to 77.9 and worse top-1 and top-5 errors on ImageNet are 25.3 and 7.8, respectively. Details on networks and learning procedure, useful to reproduce our results, can be found in Appendix B.

OOD datasets. For each ID dataset, we will consider multiple OOD datasets, mostly with disjoint label spaces and whose proximity with the ID data will vary, offering a broad spectrum of cases to assess on which tasks each indicator is effective. Table 3 describes the datasets we used as OOD. All are standard image classification benchmarks, except Gaussian (generated noise). Tiny ImageNet is not used against ImageNet; we used CIFAR 10/100 as OOD instead. All images were resized and cast to RGB when needed, then rescaled in the range $[0, 1]$ and normalized channel-wise according to the ID dataset input statistics.

Metric. We tackle the problem from the OOD rejection perspective. This means we consider OOD samples as *positive*. We use the test sets of CIFAR 10, CIFAR 100 or ImageNet as *negative* (ID) samples. Those have never been seen during training.

We report the area under the ROC curve (auroc) for each indicator used to discriminate between positive (OOD) and negative (ID) samples. Most papers in the domain also report the OOD rejection rate for a fixed ID acceptance rate. In our setting, ID samples are not available, and setting the threshold at a given acceptance rate is a challenge in itself. Contrary to precision-recall curves, ROC curves are fully independent of the—typically unknown—proportion of ID/OOD samples. We therefore feel auroc is the most relevant metric.

Table 4. Area under the ROC curve for OOD detection with CIFAR 10 as ID on ResNet 50. Shading highlights the 50% best scores per column (darker is better). The scores are averaged over three runs (*i.e.* network initializations). Note that IN- indicators are independent of the network, hence the single value.

	Gaussian	SVHN	MNIST	fashion MNIST	Tiny ImageNet	LSUN (test set)
ODIN	91.36 ± 5.42	90.22 ± 4.03	96.88 ± 0.70	95.89 ± 0.75	87.22 ± 2.12	92.38 ± 1.56
T1000	83.17 ± 9.00	93.14 ± 3.05	94.81 ± 0.78	95.43 ± 0.62	88.70 ± 1.23	92.66 ± 1.04
MP	89.27 ± 4.90	91.89 ± 1.30	90.76 ± 0.65	91.97 ± 0.47	87.05 ± 0.61	90.08 ± 0.60
H	89.05 ± 5.03	92.51 ± 1.46	91.40 ± 0.62	92.71 ± 0.58	87.52 ± 0.67	90.62 ± 0.59
NORM	53.96 ± 33.02	85.46 ± 10.89	92.28 ± 4.92	89.52 ± 4.00	80.19 ± 4.27	82.50 ± 4.93
NORM+	54.99 ± 28.60	87.17 ± 9.12	94.61 ± 2.09	92.92 ± 1.85	85.00 ± 2.61	88.87 ± 2.82
ACT	83.34 ± 9.02	93.32 ± 2.95	94.90 ± 0.70	95.47 ± 0.59	88.77 ± 1.18	92.50 ± 1.08
ACT+	87.68 ± 9.18	94.23 ± 3.50	96.03 ± 1.44	95.93 ± 0.72	88.05 ± 1.53	91.68 ± 1.38
PROJ	85.53 ± 8.09	94.01 ± 2.42	95.61 ± 0.40	95.47 ± 0.58	88.61 ± 1.26	92.05 ± 1.21
ANG	91.78 ± 2.79	93.41 ± 0.09	94.15 ± 0.60	94.76 ± 1.02	88.35 ± 0.51	91.98 ± 0.58
ANG++	99.89 ± 0.12	97.26 ± 0.17	94.25 ± 1.22	93.41 ± 1.70	86.05 ± 0.88	88.43 ± 0.75
IN-DMS	7.85	60.46	98.59	71.94	52.89	49.26
IN-DMS-AOS	52.79	30.41	99.68	96.02	52.55	54.91
IN-DSS	5.13	85.99	36.16	58.53	52.03	42.94
DMS	100.00 ± 0.00	80.29 ± 8.30	93.97 ± 2.47	69.39 ± 6.49	34.21 ± 5.54	22.67 ± 5.33
DMS-AOS	99.25 ± 0.48	4.72 ± 2.26	81.12 ± 9.04	59.42 ± 9.53	25.25 ± 2.65	23.78 ± 2.66
DSS	99.86 ± 0.14	96.51 ± 0.60	70.33 ± 15.30	62.22 ± 3.53	55.01 ± 1.90	47.40 ± 4.40
DSS-EXT	98.24 ± 0.61	97.70 ± 0.34	66.93 ± 1.88	67.64 ± 1.67	66.84 ± 0.88	62.94 ± 1.38
supervised	100.00 ± 0.00	99.75 ± 0.05	100.00 ± 0.00	99.70 ± 0.03	90.82 ± 0.45	96.14 ± 0.19
1C-Sum	97.84 ± 2.70	97.83 ± 0.95	96.47 ± 1.58	95.86 ± 0.63	88.86 ± 0.79	91.61 ± 0.90

Supervised results. We also include supervised results. In that case, half of the ID testing set and half of the OOD data are used to build a linear SVM [8]. The remaining half are used to evaluate the indicators. This means that the training and testing OOD samples are from the *same* distribution. This is clearly an ideal situation, totally outside of our setting. These results are only reported for comparison purpose. It is worth noting that the supervised approach performs almost perfectly on the easy tasks and is almost always best on the hard ones.

3.1. Sample-free indicator analysis

Table 4 shows areas under the ROC curves (auroc) for OOD detection with CIFAR 10 as the ID set on ResNet 50. Detailed tables for the other ID sets and networks are present in Appendix C.1. Table 5 summarizes the average rank (over all the OOD datasets) of each indicator for all settings. Note that the ranking is sensitive to the choice of OOD datasets, although major trends seem stable. For the purpose of this section, the last line can be ignored.

Baseline indicators. ODIN performs well in the case of ImageNet. On CIFARs, it is less clear whether the cost of the backward pass is worth it compared to simply using T1000. As envisioned in the previous section, H is slightly better than MP, although ODIN and T1000 are better suited as single indicators.

Batchnorm indicators. They do not work consistently. For these indicators, the OOD dataset has a high impact on the ranking and results are better understood by looking individually at the datasets (*e.g.* Table 4). They are intuitive, however. Indicators based on the input normalization work only on grey-level datasets. When input statistics are close to the ID's (Tiny ImageNet, LSUN), those indicators do not work better than random. They also fail on the noisy Gaussian dataset, which has individual pixel statistics that are close to ID's. It would be easy to reject such samples if inter-channel information were available, as demonstrates the indicators based on all batchnorm layers for which such information is made available thanks to the convolutions. Overall, it is clear that, in our setting, batchnorm indicators can only discriminate specific OOD sets.

Latent space indicators. As expected, NORM and NORM+ do not convey the appropriate information. The remaining indicators rank well, however. On ImageNet, positive-only indicators seem to work better, while this is not as clear for the other ID tasks. In particular, ANG++ performs better than ANG on ImageNet but ANG works better in the other settings (except for ResNet 50 on CIFAR 100). Once again, the OOD dataset has an impact on the ranking: ACT/ACT+ tend to struggle with (fashion) MNIST on CIFAR 100 and ImageNet (Appendix C.1), while, with ImageNet as ID task, ANG++ comes way ahead of the other indicators against CIFARs as OOD but underperfoms on LSUN. On the hardest cases with CIFARs as ID tasks (*i.e.* rejecting Tiny ImageNet/LSUN samples) ODIN does not perform better than T1000.

Discussion. Batchnorm indicators can capture gross statistical differences but fail on more challenging tasks. For those, optimality-based indicators are more appropriate. In a

Table 5. Average indicator rank (lower is better). These are the average across datasets of the indicator rank per dataset. R50, W and D121 stand for ResNet 50, WideResNet and DenseNet 121, respectively. Shading highlights the 50% best (*i.e.* top most) scores per column (darker is better).

	CIFAR 10			CIFAR 100			ImageNet		
	R50	W	D121	R50	W	D121	R50	W	D121
ODIN	7.30	8.20	10.70	7.20	7.30	6.30	3.40	4.60	4.90
T1000	7.80	7.30	9.30	7.50	8.00	6.70	9.30	9.90	10.30
MP	11.80	11.80	8.80	9.80	13.80	10.00	12.60	11.60	13.40
H	11.00	10.20	7.30	12.50	10.30	11.20	8.90	8.60	9.60
NORM	14.50	13.20	18.00	14.70	15.70	14.70	13.70	18.30	14.90
NORM+	12.30	10.30	15.20	12.80	14.20	13.30	12.30	17.00	13.10
ACT	7.00	6.30	8.50	7.20	8.00	6.70	9.40	10.00	10.10
ACT+	7.00	7.70	12.80	8.00	8.00	8.20	6.70	9.60	9.00
PROJ	7.20	6.00	7.70	9.30	7.30	8.80	9.70	9.90	9.10
ANG	8.20	9.00	4.50	7.80	8.30	7.00	11.30	8.40	10.90
ANG++	8.50	13.70	7.30	4.30	10.70	8.80	4.70	3.10	5.10
IN-DMS	14.50	14.80	14.50	14.80	15.00	14.80	17.00	15.60	16.40
IN-DMS-AOS	12.30	12.20	11.70	11.00	12.70	11.30	14.70	14.00	14.10
IN-DSS	18.50	18.50	17.70	18.00	15.50	17.80	17.60	17.00	17.40
DMS	14.50	12.00	11.00	16.30	9.50	16.20	8.30	10.40	12.60
DMS-AOS	16.70	16.80	17.20	13.30	16.70	13.80	18.30	16.00	17.70
DSS	12.80	12.00	11.30	19.50	7.80	19.30	11.40	9.30	6.70
DSS-EXT	12.20	14.30	10.80	9.20	13.30	9.20	14.30	11.40	9.30
supervised	1.00	1.00	1.00	1.30	1.80	1.70	1.00	1.00	1.00
1C-Sum	4.80	4.70	4.70	5.30	6.00	4.20	5.40	4.40	4.30

few instances, ANG/ANG++ perform extremely well. ODIN is also a strong baseline *if* the cost of the backward pass can be paid. Note however that the gap between ODIN and ANG++ is usually wider when the former underperforms (*e.g.* Gaussian and SVHN on Table 4), suggesting that ANG++ is more robust besides being faster to compute.

Since PROJ and ACT/ACT+ are harder to bound, they are also harder to use in a sample-free context. On that matter, Table 2 displays some statistics about a few indicators. As can be seen, pinpointing where the threshold should be placed is not easy on challenging tasks, at least without data. This will be discussed further in Section 5.

3.2. Additional results

Appendices C.2 to C.5 contain additional experiments related to the complementarity of the indicators (Appendix C.2), the impact of the quality of the base model (Appendix C.3), the joint task of rejecting OOD samples as well as classification errors (Appendix C.4), and the case of semantic anomalies (Appendix C.5). We briefly summarize the key findings related to the former and the two latter questions.

Complementarity/redundancy. We ran a PCA on the indicators for several datasets independently in order to analyze indicator complementarity. The first components accounts for 50% of the total variance, although roughly half of the components are needed to account for 95% of the variance. Analysing the loadings of the first three components shows that the indicators can be partitioned into three categories that follows intuition: the optimality-based ones, the IN- indicators and the remaining batchnorm ones.

Error detection. We wanted to check whether wrongly rejected ID samples correspond to misclassified ones. All indicators are not equal in this respect. MP and H are good at detecting errors, as are ANG, ANG++ and PROJ to a lesser extent. In comparison, T1000 and ODIN lag behind. As a consequence, when tackling both OOD and misclassification detection at the same time, ANG, ANG++ and PROJ tend to perform better on average than T1000 and ODIN.

Semantic anomalies. [2] recently defined semantic anomalies as samples from previously unseen classes with very similar distribution as the ID samples. We carried out an experiment in Section C.5 following the protocol of [2], that shows that identifying these anomalies is challenging in a sample-free setting (batchnorm indicators are, *e.g.*, useless).

4. Summary indicator

The previous section concluded that there is no one-fits-all indicator. In this section, we attempt to remedy this by proposing a summary indicator and evaluating its performance against the other individual indicators.

Whereas combining indicators when ID and OOD data are available is as straightforward as learning a model, it is not an easy task in a sample-free setting. Accordingly, we propose a simple aggregation scheme that consists in summing (a subset of) the previously-introduced indicators. The simple intuition behind this sum is that it will allow to benefit both from the redundancy and complementarity of the individual indicators. We called this aggregation scheme the 1-class sum (1C-Sum).

Since all indicators are such that their values are low (resp. high) for ID (resp. OOD) samples, this is also the case of their sum. However, since indicators have very different distributions (see Table 2), directly summing them would give them largely uneven weights in the aggregated indicator. We thus propose to first rescale their distributions to comparable ranges by standardizing them. In principle, this requires to estimate the mean and variance of these indicators on ID samples, which are unavailable. We propose instead to estimate statistics of the indicators on randomly generated data. Arguably, random data could lead to poor estimates of ID means and variances but will, hopefully, nevertheless allow to rescale the indicators to more comparable ranges.

As normalizing data, we chose uniform noise $\mathcal{U}(0, 1)$, matching the network input size. Samples drawn from this distribution are then standardized according to the ID statistics, as usual. Let h_i, $i = 1, \ldots, N$ be the collection of indicators, $\mu_i = \mathbb{E}_{x\sim\mathcal{U}}\{h_i(x)\}$ be the expectation of the ith indicator under the uniform distribution, and $\sigma_i^2 = \mathbb{V}_{x\sim\mathcal{U}}\{h_i(x)\}$ its variance. The summary indicator H is the following sum:

$$H(x) = \sum_{i=1}^{N} \frac{h_i(x) - \mu_i}{\sigma_i}. \tag{31}$$

We introduced in this sum all indicators, except the IN- indicators and ODIN. The former performed poorly on the Gaussian dataset and are thus expected to result in unsuitable standardization under uniform noise. ODIN was excluded to avoid its costly backward pass and keep the complexity of the 1C-sum as low as possible.

Empirical analysis. To validate 1C-Sum, we tested it in the same experimental conditions as for the other indicators (see Section 3 for more details). From Table 5, one can see that 1C-Sum performs extremely well, being almost always the second best in terms of average ranking (after the supervised approach which is not realistic in our setting). On the few instances where it does not come second, it has a rank close to its challengers (ANG++, ODIN). The contrary cannot be said: ANG++ and ODIN can have far worse rank than 1C-Sum. This is because when 1C-Sum is beaten by an indicator, it is never by far. Overall, 1C-Sum is quite stable.

5. Real-world setting

The empirical analysis highlighted several indicators as adequate, in the sense that they provide a thresholdable quantity capable of separating well ID and OOD samples. The analysis was conducted through the lens of the auroc, a threshold-agnostic metric. In practice, however, a cut point for the indicator must be chosen in order to automatically reject samples. Although some indicators are more interpretable than others, it remains challenging to set a threshold in a sample-free, and also architecture-independent, fashion (See Table 2 and box-plots in Appendix D).

The approach we advocate is to collect a few samples while the model is deployed in real conditions to adapt the threshold (and fine-tune the weights of 1C-Sum). We sketch below a few of such solutions.

Test samples can be labeled. If some (human) effort can be dedicated to labeling observed samples as ID or OOD, setting a threshold is straightforward. Because of the univariate nature of our indicators, we expect that only few samples would be needed to converge to a stable threshold, although it depends on the expected proportion of OOD samples. Obviously, if many labeled samples become available, the problem will stop being sample-free and one could consider supervised approaches. Our experiments show that excellent results can be reached by fitting a simple linear model on all our indicators.

No labeling is possible. Addressing the problem of setting a threshold fully automatically and without any labeling is only possible in our opinion if some assumptions can be made on the OOD data. Let us consider two examples. First, if a good guess could be made regarding the expected proportion of observed OOD samples, one could simply set the threshold so as to isolate that proportion of samples in the stream of data. Second, if the OOD distribution is stable and far away from the base distribution in the indicator space, it is possible possible to isolate both parts of the mix distribution by minimizing the intra-variance along the indicator in a unsupervised way.

6. Conclusion

In this paper, we tackled the challenging task of out-of-distribution (OOD) detection with no data, assuming a white-box access to the network. We firstly introduced several indicators for the task and conducted an empirical analysis of them. We then proposed a summary indicator, since having a single quantity to deal with is much easier in an unsupervised setting.

Provided the indicators can be thresholded appropriately, we have shown them to perform well. In particular, they cover three cases. Some batchnorm indicators are efficient at detecting gross channel-wise statistical differences, while others are good at filtering out noise. On harder tasks, optimality-based indicators were found to be more appropriate. The summary indicator is a good default choice and can be further fine-tuned if data become available.

Finally, we proposed several ways to use these indicators in practical setting, depending on the information that can be gathered when the model is deployed and the assumptions that can be made about the nature of the OOD data. Hopefully, the simplicity of the indicators should render that phase efficient

References

[1] Vahdat Abdelzad, Krzysztof Czarnecki, Rick Salay, Taylor Denounden, Sachin Vernekar, and Buu Phan. Detecting out-of-distribution inputs in deep neural networks using an early-layer output. *arXiv preprint arXiv:1910.10307*, 2019. 3

[2] Faruk Ahmed and Aaron C. Courville. Detecting semantic anomalies. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3154–3162. AAAI Press, 2020. 2, 7

[3] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Discriminative out-of-distribution detection for semantic segmentation. *arXiv preprint arXiv:1808.07703*, 2018. 2

[4] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 13166–13175. IEEE, 2020. 1

[5] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020. 1, 4

[6] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3514–3522, 2019. 1

[7] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Advances in Neural Information Processing Systems*, pages 2902–2913, 2019. 3

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 6

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018. 2

[11] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2151–2159. PMLR, 2019. 2

[12] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pages 9758–9769, 2018. 3

[13] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[15] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1, 2, 3

[16] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 2

[17] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. 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*, pages 10951–10960, 2020. 3

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015. 4

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[22] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 2

[23] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018. 3

[24] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 2, 3

[25] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4743–4751. PMLR, 2019. 1

[26] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[27] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13991–14002, 2019. 2

[28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[29] Igor M Quintanilha, Roberto de ME Filho, José Lezama, Mauricio Delbracio, and Leonardo O Nunes. Detecting out-of-distribution samples using low-order deep features statistics. 2018. 2, 4

[30] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems*, pages 1396–1408, 2019. 3

[31] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pages 14680–14691, 2019. 2

[32] Ryne Roady, Tyler L Hayes, Ronald Kemker, Ayesha Gonzales, and Christopher Kanan. Are out-of-distribution detection methods effective on large-scale datasets? *arXiv preprint arXiv:1910.14034*, 2019. 2

[33] Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with in-distribution examples and gram matrices. *CoRR*, abs/1912.12510, 2019. 3

[34] Alireza Shafaei, Mark Schmidt, and James J Little. Does your model know the digit 6 is not a cat? a less biased evaluation of" outlier" detectors. *arXiv preprint arXiv:1809.04729*, 2018. 2

[35] Kumar Sricharan and Ashok Srivastava. Building robust classifiers through generation of confident out of distribution examples. *arXiv preprint arXiv:1812.00239*, 2018. 2

[36] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 550–564, 2018. 3

[37] Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, A. Taylan Cemgil, S. M. Ali Eslami, and Olaf Ronneberger. Contrastive training for improved out-of-distribution detection. *CoRR*, abs/2007.05566, 2020. 3

[38] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 5

[39] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 4

[40] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5

[41] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press, 2016. 5