# Class-wise Thresholding for Robust Out-of-Distribution Detection

Matteo Guarrera
University of California, Berkeley
matteogu@berkeley.edu

Baihong Jin
University of California, Berkeley

Tung-Wei Lin
University of California, Berkeley

Maria A. Zuluaga
EURECOM

Yuxin Chen
University of Chicago

Alberto Sangiovanni-Vincentelli
University of California, Berkeley

## Abstract

*We consider the problem of detecting Out-of-Distribution (OoD) input data when using deep neural networks, and we propose a simple yet effective way to improve the robustness of several popular OoD detection methods against label shift. Our work is motivated by the observation that most existing OoD detection algorithms consider all training/test data as a whole, regardless of which class entry each input activates (inter-class differences). Through extensive experimentation, we have found that such practice leads to a detector whose performance is sensitive and vulnerable to label shift. To address this issue, we propose a class-wise thresholding scheme that can apply to most existing OoD detection algorithms and can maintain similar OoD detection performance even in the presence of label shift in the test distribution.*

## 1. Introduction

With the recent advancement in deep learning, image classification has shown great performance improvement under well-controlled settings where the test data are clean and sampled from the same distribution as the training data. However, the deployment of deep learning models in the real world is still full of unknowns. More often than not, well-trained models can come across Out-of-Distribution (OoD) data that are sampled from a different distribution than the one used for training. For example, objects that do not belong to any of the classes in the training data (*i.e.*, OoD inputs) can appear at test time. Faced with OoD inputs, deep learning-based classifiers may render unpredictable behaviors and often tend to make *overly confident* decisions [27]. To address this issue, many previous works [11, 12, 20, 22, 25] have been dedicated to detect such OoDs inputs. Therefore, in safety-critical applications such as healthcare and autonomous driving [17], the classifier should have the ability to yield control to humans upon coming across such inputs, instead of making incorrect predictions silently.

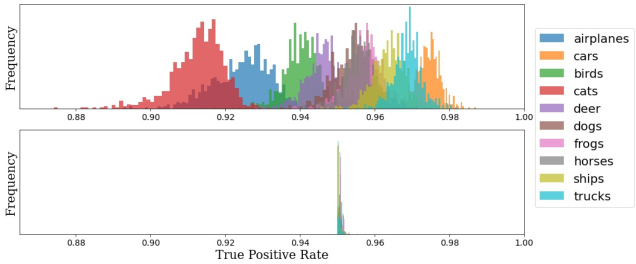


Figure 1. Class-wise True Positive Rate (TPR) variations (*CIFAR10-WideResNet model, max-logit detector*) in simulated scenarios under label shift. Narrower histograms in the proposed thresholding scheme (bottom panel), compared to the state-of-the-art (top panel), suggests that our approach is able to guarantee higher robustness through almost constant false alarm rate (5% in this case).

Among the plethora of works on OoD detection, almost all previous literature focuses on improving the detection performance on various OoD test sets. Their experiment setups *implicitly* assume that the training and the test In-Distribution (ID) data follow the same distribution (no distribution shift), so that the false alarm rates at test time will stay at the same level observed at training. However, it is often not the case in real-world settings, and the distribution shift may result in an increased or decreased number of false alarms on the ID data, which can lead to errors incurring into economic losses (additional costs to address these false alarms). Worse still, malicious attackers can exploit this weakness to launch attacks that may cause an overflow of false alarms for certain classes, thus eventually lowering the sensitivity of the detection system against actual OoD inputs (due to the excessive number of false alarms injected by attackers). The above-mentioned issues are indeed vital for real-world deployment of such systems.

In this paper, we specifically target OoD detection algorithms built upon supervised multi-class classifiers, and address the above-mentioned challenges in the context of label shift, a special type of distribution shift, by using a novel thresholding scheme. Our approach is applicable even for black-box models, where the internal structure and parameters of the classifier are invisible. The contributions of this paper are three-fold:

- We identify a problem that makes many existing OoD detection algorithms vulnerable to test-time label shift.
- We propose a *simple* yet *effective* thresholding scheme to address the challenge, and show empirically that our solution can be used as a plugin amendment to any existing OoD system with a class-wise score function.
- Using our novel thresholding scheme, we also assess the performance limit of several *learning-based* OoD detectors, and compare them with *non-learning-based* ones. The study provides some guidance on how to navigate the design space of OoD detection systems.

## 2. Proposed Approach

### 2.1. Problem formulation

We consider the *OoD detection* problem in *supervised multi-class classification* settings; our goal is to identify whether a data point (image) $\boldsymbol{x}$ comes from the distribution $\mathcal{D}_{\text{in-dist}}$ which the development set data are sampled from. Let us denote a given trained classifier by a function $f_\theta$ whose parameters $\theta$ are learned through a training procedure using data sampled from $D_{\text{in-dist}}$.

The training dataset $D_{\text{train}} = \{(\boldsymbol{x_i}, y_i)\}$ is a collection of image and (categorical) label pairs sampled from $\mathcal{D}_{\text{in-dist}}$, where label $y_i$ for image $\boldsymbol{x}_i$ takes integer values from set $\{1, 2, \dots, K\}$, each corresponding to one of the $K$ ID classes. The trained classifier $f_\theta$ learns to map an image $\boldsymbol{x}_i$ to a *logit* vector $\boldsymbol{\ell} \in \mathbb{R}^K$ that eventually produces a probability vector $\boldsymbol{p}$ after softmax transformation. Through the training procedure, the classifier parameters $\theta$ are updated so as to minimize a given loss function for $D_{\text{train}}$, *e.g.*, the cross-entropy loss.

**Non-learning-based detectors** To detect OoDs, we need to define an *OoD score* function $S$ for each input $\boldsymbol{x}$ given classifier $f_\theta$ to indicate how likely the given input is OoD. Ideally, an OoD detector will always assign higher OoD scores to OoD data than to ID data. In one setup, it is assumed that the OoD detector can only observe the output logit vector $\boldsymbol{\ell}$ of classifier $f_\theta(\boldsymbol{x})$ but not its internal structure [9, 11, 14, 22, 25]; in other words, the model is viewed as a *black box*. This assumption allows OoD detection algorithms to be applicable on almost all classifiers. In an alternative setup [20], the OoD detector also has access to the internal structures or the hidden feature mappings of a classification model, which can serve as additional information that is potentially useful for OoD detection.

For the former setup where only the logit vector $\boldsymbol{\ell}$ is available for OoD detection, two categories of simple statistics on the logit vector $\boldsymbol{\ell}$, the max-logit score (or the very similar max-softmax confidence score) [9, 11] and the energy score [25], are commonly used in literature as the OoD score function. For the latter setup, Lee *et al*. [20] utilized the *Mahalanobis* distance for defining the OoD score.

The max-logit and max-softmax approaches are based on the intuition that OoD data will result in lower softmax confidence scores $\max_j p(y = j \mid \boldsymbol{x})$ as well as the corresponding $j$th logit entry $\ell_j$. Depending on which specific statistic to use, we can define two very related detection methods:

- the *max-logit* approach that uses $S_{\text{max-logit}}(\boldsymbol{x}; f_\theta) = -\max_j \ell_j$ as the OoD score, and
- the *max-softmax* approach that uses $S_{\text{max-softmax}}(\boldsymbol{x}; f_\theta) = -\max_j p(y = j \mid \boldsymbol{x})$ as the OoD score.

Less confident inputs will receive higher OoD scores. The two approaches have demonstrated good detection performance in several empirical studies [9–12], and have been widely used as a baseline method due to their simplicity.

The *energy-based* approach [25] as an alternative to max-logit has also shown good performance in prior literature. Instead of using only the maximum logit entry, the energy-based approach defines the energy function (energy-based OoD score)

$$S_{\text{energy}}(\boldsymbol{x}; f_\theta) = -T \cdot \log \sum_{j=1}^{K} \exp(\ell_j / T) \tag{1}$$

as the OoD score. Here, $\ell_j$ represents the $j$th entry of the logit vector of $\boldsymbol{x}$ and $T$ represents the temperature parameter. It can be proven that the (negative) energy function is a smooth approximation of the maximum logit entry, justifying the similarities among the two statistics for OoD detection. Mathematically, we have the following relation between the max-logit score and the energy score,

$$\max_{1 \le j \le K} \ell_j < -S_{\text{energy}}(\boldsymbol{x}; f_\theta, T) \le \max_{1 \le j \le K} \ell_j + T \cdot \log(K). \tag{2}$$

Because the above-mentioned statistics for calculating the OoD score are pre-defined and no further learning/tuning is needed, we will hereinafter refer to these methods above as *non-learning-based*.

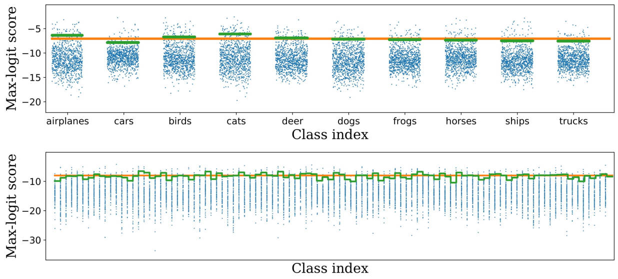


Figure 2. For the pre-trained CIFAR10-WideResNet (top panel) and CIFAR100-WideResNet (bottom panel) networks, we display the activated logit entry values (*i.e.*, one that becomes the maximum logit), and also the corresponding ONE (in orange) and MULTI (in green) threshold values determined by TPR-95.

**Learning-based detectors** As already demonstrated by the energy-based approach [25], more OoD examples can be detected by considering all logit entries, instead of only the maximum one as in max-logit. One may wonder if we can find another candidate function that performs better. In *learning-based* approaches, the OoD score function $S$ is learned or fine-tuned from data with known ID and (optionally) OoD labels. With Machine Learning (ML), the design space for detectors is hugely expanded, which can produce improved results. Different from non-learning-based methods, learning-based methods infer the decision rule by using the data (*i.e.*, the network's responses) instead of a predefined rule.

The actual choice of learning algorithms relies on the availability of labeling information. If only positive-label (*i.e.*, ID) data are available, semi-supervised anomaly detection algorithms can be employed to capture the distribution of ID data and differentiate them from OoD data. Common choices include anomaly/outlier detection models, such as One Class-Support Vector Machine (OC-SVM) and autoencoders. If some negative-label (*i.e.*, OoD) data are available, then not only can they serve as validation data for tuning the hyperparameters of semi-supervised models such as OC-SVM, but they also enable the use of supervised classification models such as fully-connected neural networks for differentiating between ID and OoD data.

**Constant false alarm rate scheme for threshold setting** Since OoD detection is essentially a binary classification problem, a threshold $\tau$ on the OoD score $S(\boldsymbol{x})$ is needed to dichotomize between ID and OoD. In mathematical form, we have

$$\boldsymbol{x} \text{ is} \begin{cases} \textit{out-of-distribution}, \text{ if } S(\boldsymbol{x}) \geq \tau \\ \textit{in-distribution}, \text{ otherwise.} \end{cases} \tag{3}$$

Since there is a natural trade-off between false positives (*i.e.*, OoD misclassified as ID) and false negatives (*i.e.*, ID misclassified as OoD) when we modulate the detection threshold $\tau$, a commonly used method for setting $\tau$ is the "TPR-$\beta$" thresholding scheme, where $1-\beta\%$ is a preset false alarm rate. In practice, $\tau$ is usually determined on a separate validation set $D_{\text{valid}}$ (different from $D_{\text{train}}$ but also sampled from $\mathcal{D}_{\text{in-dist}}$) to meet the pre-defined false alarm rate level $1-\beta\%$. Although the TPR-$\beta$ method seems to ensure a constant false alarm rate on $\mathcal{D}_{\text{in-dist}}$, it still suffers from two subtle problems as to be elaborated below.

**Non-uniform false alarm rates across in-distribution classes** Under the above scheme, the false alarm rate for each class may deviate much from the preset level $1-\beta\%$, due to the misalignment among the distributions of each logit entry. To visually illustrate this issue, we repeated the experiments reported in Liu *et al*. [25] with the same pre-trained network therein, and plotted in Figure 2 the distribution of the (test-time) logit values for each output node. As we can see, the distributions of the output scores (*i.e.*, the maximum logit values) on each output node are not aligned. A single, unified cutoff threshold (shown in orange) for all the logit entries will result in very different false alarm rates (*i.e.*, the ratios of data points above the set threshold) across the classes.

**Sensitivity to label shift** In addition to the above issue, the "constant" overall false alarm rate can still be fragile, *i.e*. not *robust*, under distribution shift. Particularly, let us take *label shift* [24], a common type of distribution shift, for example. In the presence of label shift, the label marginal $p(y)$ changes but the conditional $p(\boldsymbol{x} \mid y)$ does not. If the label distribution $p(y)$ changes for the test data, the overall false alarm rate can easily fluctuate under the single-threshold approach, due to the varying false alarm rate for each class. To illustrate the issue, we did another experiment on top of the one reported above, again using the outputs of a pre-trained network, to simulate the effect of label shift. Let us denote by $p_{\text{train}}(y)$ the label marginal of the $\mathcal{D}_{\text{in-dist}}$ and by $p_{\text{test}}(y)$ the label marginal of the $\mathcal{D}_{\text{in-dist}}$, where both $p_{\text{train}}(y)$ and $p_{\text{test}}(y)$ are $K$-dimension probability vectors with elements summing up to 1. In the presence of label shift, $p_{\text{train}}(y) \neq p_{\text{test}}(y)$, which as explained above will cause varying false alarm rates under the single-threshold scheme at test time. We performed a simulation study with $10000$ pairs of randomly picked $p_{\text{train}}(y)$ and $p_{\text{test}}(y)$ and plotted the resulting false alarm rates against $\Delta p \doteq \|p_{\text{train}}(y) - p_{\text{test}}(y)\|_2$ (Fig. 3). As can be seen, the spread of resulting false alarm rates at test time becomes larger with increasing $\Delta p$ (*i.e.*, worsening label shift phenomenon), even though we select the threshold hoping to control the false alarm rates to be around $0.05$ (a pre-defined level).

The above observations of fluctuating false alarm rates

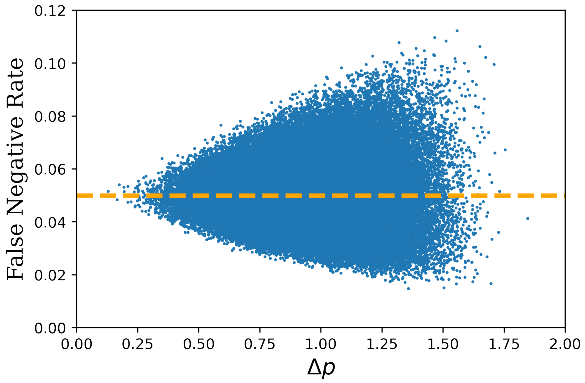


Figure 3. False alarm rate variation under simulated label shift for the pre-trained CIFAR10-WideResNet model. We ran 100000 simulations and each green point represents one trial.

are concerning, since controlling the number of false alarms is highly important for almost all anomaly/outlier/OoD detection tasks. Although the false alarm rate deviation at test time may be small (say about $1\%$ in the above example), the actual increase in the number of false alarms can be significant since ID data (inliers) typically account for the majority of the test data. Worse still, malicious attackers may tamper with the false alarm rates and thereby influence the normal operation of such detection systems. To the best of our knowledge, the above-mentioned issues with existing OoD detection algorithms have never been addressed before in OoD detection literature. In the upcoming section, we will present a simple yet effective solution to this problem.

### 2.2. MULTI: Class-wise thresholding scheme

In the multi-class classification setting, the ID data consist of images of multiple classes. Therefore, we can treat $\mathcal{D}_{\text{in-dist}}$ as a compound distribution of multiple generating processes $\{g_j\}_{j=1}^K$, one for each class. When sampling an image $\boldsymbol{x}$ from $\mathcal{D}_{\text{in-dist}}$, we are actually sampling from the $j$th generating process $g_j$ at probability $p(j)$, *i.e.*, the label marginal for class $j$.

A common assumption in OoD literature is that the training data $D_{\text{train}}$ and the ID test data $D_{\text{test}}$ come from the same distribution $\mathcal{D}_{\text{in-dist}}$, so that we would expect to get similar false alarm rates on both $D_{\text{train}}$ and $D_{\text{test}}$. As we have explained earlier, this assumption may not hold when we deploy the classifier and the OoD detector in real-world settings due to issues such as label shift. In the presence of label shift, the label marginals $\{p(j)\}_{j=1}^K$ change but the underlying generating processes $\{g_j\}_{j=1}^K$ stay the same. This suggest us to devise a way to break down the complexity of $\mathcal{D}_{\text{in-dist}}$ by taking the class label information into account when detecting OoDs. By considering class-dependent scores, we are designing a dedicated decision rule for the case where the $j$th logit becomes the maximum (*i.e.*, the *activated* logit). We are dividing the entire $K$-dimensional space into $K$ **disjoint** (non-overlapping) subspaces $\{\mathbb{R}_i^K\}_{i=1}^K$, and designing a dedicated decision rule for each subspace. Mathematically, let us denote by $\boldsymbol{x}^{(j)}$ an input of class $j$, and $\boldsymbol{\ell}^{(j)}$ the corresponding logit vector. If $f_\theta$ can correctly classify $\boldsymbol{x}^{(j)}$ as class $j$, we have

$$\boldsymbol{\ell}^{(j)} = f_\theta(\boldsymbol{x}^{(j)}) \Rightarrow \boldsymbol{\ell}^{(j)} \in \mathbb{R}_j^K \subset \mathbb{R}^K, \tag{4}$$

where $\mathbb{R}_j^K = \{\text{argmax}_k\, \ell_k = j\}$, $j = 1, 2, \ldots, K$. Here, the $\text{argmax}$ operator finds the first occurrence of the maximum entry (in case of multiple occurrences of the same maximum value).

To consider each subspace separately, we can design a separate detection model (OoD score function) $S_j$ for each logit subspace $\mathbb{R}_j^K$, or to share the same OoD score function $S$ but use different thresholds. By setting a different threshold $\tau_j$ for each class, we can treat each predicted class and its subspace as a separate entity. Essentially, *we are enjoying the benefit of having $K$ models, one for each class, in a much less expensive way.*

Based on the above analysis, we can easily extend the single-threshold approach (3) by using a dedicated threshold $\tau_j$ for each class $j$; this can apply to every detection algorithm that produces a *class-dependent* score, *i.e.*, where each class $j$ is generated by a given $p_j(\boldsymbol{x})$. In general, suppose $S(\boldsymbol{x})$ is the OoD score given by a detection algorithm for input $\boldsymbol{x}$, the decision rule under the class-wise thresholding scheme can be written as follows.

$$\boldsymbol{x} \text{ is} \begin{cases} \textit{out-of-distribution}, \text{ if } S(\boldsymbol{x}) \geq \tau_j \\ \textit{in-distribution}, \text{ otherwise} \end{cases} \qquad j = \underset{k}{\text{argmax}}\, l_k. \tag{5}$$

**Applying the class-wise thresholding scheme to existing OoD detection algorithms** Both non-learning-based and learning-based detectors can easily be extended to apply the class-wise thresholding scheme. To be specific, we simply need to replace the single threshold $\tau$ for cutting off the OoD score $S(\boldsymbol{x}; f_\theta)$ with class-wise thresholds $\tau_j$, one for each *activated* logit entry (*i.e.*, the maximum one). The same TPR-$\beta$ scheme can be used to find the $\tau_j$ for each logit entry. In contrast to the previous practice that uses a single, unified cutoff threshold for all logit entries (later referred to as ONE for brevity), our approach (later referred to as MULTI) is analogous to a "switch-case" statement that uses different thresholds for different activated logits. In our empirical study to be described next, we will compare the performance of ONE and MULTI on several popular detection algorithms.

## 3. Experiment Results

We conducted extensive experiments to compare the performance of MULTI to that of ONE. To achieve a fair and comprehensive comparison between the two thresholding schemes, we performed our experiments using the same or similar settings as in several previous papers. The software implementation can be found as part of the supplementary material.

**Datasets** We used the same ID and OoD datasets as in [20, 25] as benchmarks. In addition, to get a more comprehensive view of the performance of the algorithms, we also included another two popular image databases as ID, the German Traffic Sign Recognition Benchmark (*GTSRB*) [13] and ImageNet [8].

For OoD benchmarks, we included the ones used in prior works [20,25] in our experiments, including *Places365* [43] and *SVHN* [26]. In addition, we added a few more from Kaggle and from other works in the literature [21,37] to our evaluation to cover a larger variety of subjects: *Animals* [1], *Anime Faces* [6], *Fishes* [2], *Fruits* [16], *iSUN* [38], *Jigsaw Training* [33], *LSUN* [39], *Office* [37], *PACS* [21] and *Texture* [7].

To reduce the overlap between ID and OoD datasets, we removed images from OoD datasets that share the same class labels as those in ID datasets. For example, classes *dog*, *horse*, and *cat* were removed from the *Animals* Dataset since they already existed in CIFAR10, an ID dataset in our evaluation.

**Pre-trained classification models** We tested the OoD detection algorithms on pre-trained deep learning models of three network architectures: *DenseNet* [15], *WideResNet* [40], and *AlexNet* [19]. The seven resulting pre-trained models used in our experiments are listed in Table 1. The pre-trained WideResNet models were the same as those used in Liu *et al.* [25]; the DenseNet models were from Liang *et al.* [22]. We used the ImageNet-Densenet model from `torchvision` [28], and trained an AlexNet model on the GTSRB dataset.

**Setup of OoD detectors** We set up the non-learning-based detectors, max-logit, energy-based and Mahalanobis, using the same methods as described in literature. Then we evaluated the two discussed thresholding schemes, ONE and MULTI, by using TPR-95 to determine the respective detection thresholds.

To compare ONE and MULTI, we evaluated two anomaly detection models, $k$-NN and OC-SVM for learning-based OoD detection. In addition, we also tested ODIN [22] where an optimal temperature parameter $T$ needs to be learned. As mentioned earlier, learning-based OoD detectors have the potential to perform better because of their expressiveness in capturing the density distribution of ID data. However, we also noticed the unsatisfactory outcomes from learning-based approaches reported by several previous works. We believe the reported results do not reflect the true potential of learning-based detectors, as it is well-known that hyperparameter settings have profound impacts on the performance of ML models. To gauge the full potential of learning-based methods, in our experiments we assessed the the performance limits of learning-based detectors by conducting a *best-case* analysis to measure their *maximum achievable* performance. To do so, we assumed that we have access to a validation set drawn i.i.d. from the test ID and OoD distribution for tuning the hyperparameters of learning-based detectors. For each detector, a grid-search was performed over the tunable hyperparameters to select the model instance that achieved the best performance on the test set. This analysis helped us understand the performance limits and the potential room for improvement of learning-based OoD detectors.

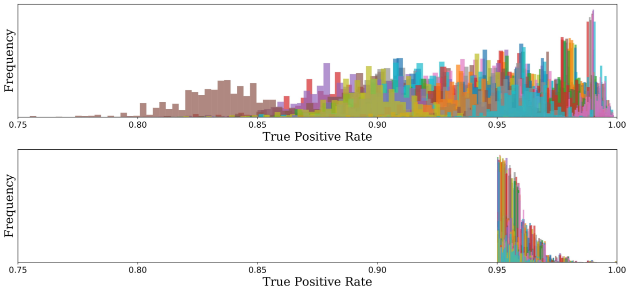


Figure 4. True Positive Rate (TPR) variations (as histograms) of each activated logit entry (shown in different colors) of the CIFAR100-WideResNet model under ONE (top panel) and MULTI (bottom panel) for the max-logit detector.

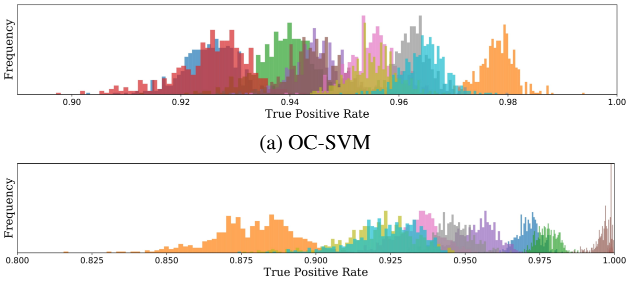
### 3.1. False Alarms (Type-1 Errors/False Negatives)

As described earlier, the single-threshold approach ONE can result in unevenly distributed false alarms. In our empirical evaluation, we tested both ONE and MULTI, and reported in Table 1 the TPR (one minus the false alarm rate) variation across ID classes for all seven pre-trained models. As we can see from the statistics, the baseline scheme ONE resulted in large performance variation. In several cases, the TPR can be as low as 70% under ONE. On the other hand, MULTI does not suffer from such problem. The results suggest that, compared to ONE, MULTI is much more desirable and robust due to its stable TPR performance.

Figure 1 and Figure 4 highlight the problem associated with ONE from a different perspective. Here, we modified the number of test-set ID data samples for each class by oversampling class $i$ with a random factor $\gamma_i \in [1, 10]$, and

Table 1. TPR (unit: %) of OoD Detectors under ONE and MULTI.

| | | *Max-logit Detector* (ONE \| MULTI) | | | *Energy-based Detector* (ONE \| MULTI) | | | *k-NN Detector* (ONE \| MULTI) | | | *OC-SVM Detector* (ONE \| MULTI) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Classifier** | $K$ | **Min. TPR (↑)** | **Max. TPR (↓)** | **Std. TPR (↓)** | **Min. TPR (↑)** | **Max. TPR (↓)** | **Std. TPR (↓)** | **Min. TPR (↑)** | **Max. TPR (↓)** | **Std. TPR (↓)** | **Min. TPR (↑)** | **Max. TPR (↓)** | **Std. TPR (↓)** |
| CIFAR10 - WideResNet | 10 | 91.2 \| **94.9** | 97.5 \| **95.0** | 1.89 \| **0.03** | 91.2 \| **94.9** | 97.6 \| **95.0** | 1.92 \| **0.03** | 88.4 \| **94.9** | 98.8 \| **95.0** | 2.72 \| **0.03** | 90.6 \| **94.9** | 97.8 \| **95.0** | 2.04 \| **0.03** |
| CIFAR10 - DenseNet | 10 | 88.9 \| **94.9** | 98.3 \| **95.0** | 2.52 \| **0.03** | 89.2 \| **94.9** | 98.2 \| **95.0** | 2.45 \| **0.03** | 91.0 \| **94.9** | 98.1 \| **95.0** | 1.85 \| **0.03** | 91.8 \| **94.9** | 97.2 \| **95.0** | 1.49 \| **0.03** |
| SVHN - WideResNet | 10 | 91.0 \| **94.9** | 96.7 \| **95.0** | 1.61 \| **0.03** | 90.8 \| **94.9** | 96.8 \| **95.0** | 1.72 \| **0.03** | 92.4 \| **94.9** | 97.0 \| **95.0** | 1.18 \| **0.03** | 83.4 \| **94.9** | 99.2 \| **95.0** | 4.46 \| **0.03** |
| GTSRB - AlexNet | 43 | 75.9 \| **93.3** | 100.0 \| **95.0** | 5.52 \| **0.39** | 75.0 \| **93.3** | 100.0 \| **95.0** | 5.58 \| **0.39** | 32.5 \| **93.3** | 100.0 \| **95.0** | 13.51 \| **0.39** | 0.8 \| **93.3** | 100.0 \| **95.0** | 22.25 \| **0.39** |
| CIFAR100 - WideResNet | 100 | 83.8 \| **93.9** | 100.0 \| **95.0** | 3.01 \| **0.31** | 83.8 \| **93.9** | 100.0 \| **95.0** | 3.10 \| **0.31** | 72.3 \| **93.9** | 100.0 \| **95.0** | 4.61 \| **0.31** | 84.2 \| **93.9** | 100.0 \| **95.0** | 3.35 \| **0.31** |
| CIFAR100 - DenseNet | 100 | 82.5 \| **93.8** | 100.0 \| **95.0** | 3.43 \| **0.31** | 81.6 \| **93.8** | 100.0 \| **95.0** | 3.41 \| **0.31** | 76.7 \| **93.8** | 100.0 \| **95.0** | 4.90 \| **0.31** | 78.1 \| **93.8** | 100.0 \| **95.0** | 3.86 \| **0.31** |
| ImageNet-DenseNet | 1000 | 72.9 \| **90.0** | 100.0 \| **94.9** | 4.82 \| **0.63** | 69.7 \| **90.0** | 100.0 \| **94.9** | 4.85 \| **0.63** | - | - | - | - | - | - |



(a) OC-SVM

(b) $k$-NN

Figure 5. TPR variations of each each activated logit entry of CIFAR10-WideResNet model under ONE for OC-SVM (top) and $k$-NN (bottom) detectors.

repeated the same experiment for 1000 times. The figures show that ONE not only leads to huge inter-class discrepancies but also induces large intra-class variations among these repeated experiments that simulate different label distribution for the test data. In contrast, MULTI avoids this problem and gives an almost constant false alarm rate (5% in this case) despite the label shift.

Figure 5 shows the variations in TPR of CIFAR10-WideResNet using ONE for $k$-NN and OC-SVM detectors. Similar to Figure 1 and Figure 4, the plots indicate large inter-class and intra-class false alarm rate variations, a not so desirable outcome for OoD detection applications. This finding again motivates the use of MULTI.

**Simulation Study Shown in Figure 3** To produce the results shown in Figure 3, we artificially modified the class ratios of the training and test datasets to match randomly chosen class ratios $p_{\text{train}}(y)$ and $p_{\text{test}}(y)$. For the CIFAR10-WideResNet case as used in this example, the class ratios $p(y) = (p_1, p_2, \dots, p_{10})$ are 10-dimension vectors where $\sum_{k=1}^{10} p_k = 1$. In other words, $p(y) \in \mathbb{S}_{10}$ where $\mathbb{S}_{10}$ is a 10-dimension probability simplex.

Computationally, to randomly sample $p_{\text{train}}(y)$ and $p_{\text{test}}(y)$ from $\mathbb{S}_{10}$ , we made use of a property of the exponential distribution $\exp(1)$. Suppose $X_i \sim \exp(1), i = 1, 2, \dots, K$ are $K$ i.i.d. samples of an exponential distribution $\exp(1)$. It can be proven that random vector

$$\left( \frac{X_1}{\sum_i X_i}, \frac{X_2}{\sum_i X_i}, \dots, \frac{X_K}{\sum_i X_i} \right)$$

is uniformly sampled from $\mathbb{S}_K$.

By using the trick above, we generated $p_{\text{train}}(y)$ and $p_{\text{test}}(y)$, computed the resulting false alarm rate for the test distribution under label shift, and repeated the same experiment for 100000 times to obtain the scatter plot shown in Figure 3.

## 3.2. Missed Detections (Type-2 Errors/False Positives)

Next, let us examine how MULTI impacts the OoD detection performance, in terms of the missed detection rates (ratios of OoDs that are mistaken as IDs). We computed the average missed detection rates for the five different OoD detection methods over the aforementioned OoD benchmarks under both ONE and MULTI, for all seven pre-trained classification models. The results are summarized in Table 2.

As we can see from Table 2, different pre-trained models yield similar results under ONE and MULTI, and we will again take a more detailed look at the results from CIFAR10-WideResNet, shown in Table 3. As can be seen, the False Positive Rate (FPR) performance differences between ONE and MULTI are small, up to a few percentage points. Considering the shortcomings of ONE discussed above, the slight performance trade-off from using MULTI is well worth it.

In Table 3, for a given algorithm, we can also see large performance variations across OoD benchmark datasets. To get a better understanding of the obtained results, we also calculated the statistical distances between the ID training set (*i.e*., CIFAR10 in this case) and each OoD dataset as a measure of their "OoDness". Two statistical distance metrics, the *Wasserstein Distance* [36] and the *Energy Distance* [34], were used in our calculation. We then computed the Pearson Correlation between the performances of an OoD dataset and the corresponding statistical distances (*i.e*., the correlation between two columns in Table 3). The results are shown in Table 4, where we can see negative correlations between the statistical distances and the missed detection rates. This indicates increased difficulties in OoD detection for OoD datasets that are "closer" to the ID test set in terms of statistical distances.

It is also worth noticing in Table 3 that the two learning-based detectors ($k$-NN and OC-SVM) usually give better performance than the non-learning-based ones (max-logit,

Table 2. Average Missed Detection Rates under ONE and MULTI

| | $K$ | **Max-softmax** | **Max-logit** | **Energy** | $k$**-NN** | **OC-SVM** | **ODIN** | **Mahalanobis** |
|---|---|---|---|---|---|---|---|---|
| | | | | | (ONE \| MULTI) | | | |
| **CIFAR10-WideResNet** | 10 | 0.57 \| 0.57 | 0.39 \| 0.42 | 0.38 \| 0.41 | 0.35 \| 0.38 | 0.37 \| 0.41 | 0.40 \| 0.40 | 0.45 \| 0.47 |
| **CIFAR10-DenseNet** | 10 | 0.54 \| 0.54 | 0.39 \| 0.39 | 0.38 \| 0.38 | 0.32 \| 0.34 | 0.35 \| 0.37 | 0.35 \| 0.36 | 0.51 \| 0.52 |
| **SVHN-WideResNet** | 10 | 0.09 \| 0.09 | 0.08 \| 0.08 | 0.08 \| 0.08 | 0.06 \| 0.06 | 0.14 \| 0.08 | 0.17 \| 0.16 | 0.06 \| 0.04 |
| **GTSRB-AlexNet** | 43 | 0.50 \| 0.40 | 0.33 \| 0.35 | 0.32 \| 0.34 | 0.49 \| 0.61 | 0.83 \| 0.66 | 0.25 \| 0.25 | 0.75 \| 0.71 |
| **CIFAR100-WideResNet** * | 100 | 0.79 \| 0.77 | 0.73 \| 0.71 | 0.73 \| 0.71 | 0.77 \| 0.71 | 0.77 \| 0.76 | 0.67 \| 0.67 | 0.70 \| 0.67 |
| **CIFAR100-DenseNet** * | 100 | 0.77 \| 0.75 | 0.72 \| 0.69 | 0.73 \| 0.70 | 0.81 \| 0.74 | 0.79 \| 0.76 | 0.62 \| 0.62 | 0.69 \| 0.68 |
| **ImageNet-DenseNet** | 1000 | 0.64 \| 0.70 | 0.57 \| 0.65 | 0.58 \| 0.66 | - | - | - | - |

*- denotes cases not covered in our experiment due to scalability issues.*
** same hyperparameters of the corresponding CIFAR10 classifiers*

energy and Mahalanobis), under both ONE and MULTI. Despite the fact that we are conducting a best-case analysis here for learning-based algorithms, this finding still highlights the benefits and potential of learning-based detection algorithms that utilize all logit entries for decision-making.

### 3.3. Sensitivity Analysis for Single-Threshold Approach

In addition to the above-mentioned problems with ONE, slight variations of $\tau_{\text{one}}$ can have huge impact on the final OoD detection performance at test time. Many factors, including label shift (*i.e.*, varying class ratios in ID data), can affect $\tau_{\text{one}}$. Therefore, using $\tau_{\text{one}}$ determined on the validation set can yield undesirable detection outcomes. On the contrary, our class-wise thresholding scheme MULTI is naturally robust to label shift because each class is considered separately.

We conducted an experiment to analyze the potential impacts of perturbed $\tau_{\text{one}}$ due to the choice of the validation set. To simulate additive perturbations $\Delta\tau$ on $\tau_{\text{one}}$, we uniformly sampled 50 perturbation values

$$\Delta\tau \in \delta \cdot \left[ - \left| \tau_{\text{one}} - \min_j \tau_j \right|, \left| \max_j \tau_j - \tau_{\text{one}} \right| \right],$$

where $\delta \in [0, 1]$ is a factor that modulates the maximum strength of the perturbation in the validation set. In our experiments, we set $\delta = 0.5$. Figure 6 shows that ONE often produces worse performance than MULTI under perturbed threshold $\tau_{\text{one}} + \Delta\tau$ in terms of missed detection rates, which indicates the risk of using ONE.

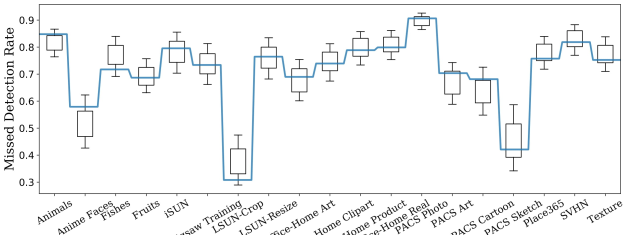


Figure 6. Variations of the missed detection rate for OoD examples from each benchmark dataset, for perturbed $\tau_{\text{one}}$ values (box plot). In blue the value for MULTI thresholding scheme.The experiment is performed on pre-trained model CIFAR100-WideResNet, with max-logit being used as the OoD detection algorithm.

## 4. Related Work

**Pre-trained models as black boxes** Hendrycks *et al.* found out that OoD images tend to have lower maximum softmax scores [11] and maximum logit [9] than ID images. Based on the assumption that neural networks are trained to lower the energy of ID data, Liu *et al.* [25] proposed to use an energy function as a score function. On the other hand, with auxiliary OoD datasets and input preprocessing, *ODIN* [22] can be considered an ML model training the temperature parameter $T$ of the softmax function at the output layer. However, the resulting detector may not generalize well to other unseen OoD samples [14]. Therefore, Hsu *et al.* [14] proposed *Generalized ODIN* without the use of auxiliary datasets. Still, input preprocessing can induce undesirable time delay for real-time applications.

**Pre-trained models with open inner structures** Breaking the black-box assumption, the *Outlier Exposure* (OE) method [12] detects OoDs by modifying the neural network loss function and with additional training. However, the auxiliary OoD datasets that are required are not always accessible [4]. Lee *et al.* [20] used *Mahalanobis* distance, and *MOOD* [23] used energy functions, while incorporating information from early layers.

**OoD Detection with generative modeling** In addition to the above-mentioned works that detect OoDs based on a given classification model, another category of works use a dedicated model for OoD detection. An intuitive approach is to use generative modeling to learn the distribution of the ID dataset and reject samples that have low likelihood of being ID. Although Choi *et al.* [5] showed that image classifiers can assign higher likelihood to OoD data, Ren *et*

Table 3. Missed Detection Rates on OoD Benchmarks for the CIFAR10-WideResNet Model.

| | **Max-softmax** | **Max-logit** | **Energy** | **k-NN** | **OC-SVM** | **ODIN** | **Mahalanobis** | **Statistical Distances** | |
|---|---|---|---|---|---|---|---|---|---|
| **Datasets** | | | | (ONE \| MULTI) | | | | **Wasserstein** | **Energy** |
| **Animals** | 0.68 \| 0.69 | 0.61 \| 0.64 | 0.61 \| 0.64 | 0.63 \| 0.66 | 0.62 \| 0.67 | 0.69 \| 0.69 | 0.80 \| 0.81 | 129 | 70 |
| **Anime Faces** | 0.61 \| 0.68 | 0.38 \| 0.44 | 0.37 \| 0.42 | 0.28 \| 0.36 | 0.32 \| 0.41 | 0.36 \| 0.34 | 0.37 \| 0.45 | 174 | 95 |
| **Fishes** | 0.51 \| 0.54 | 0.30 \| 0.37 | 0.30 \| 0.36 | 0.28 \| 0.37 | 0.29 \| 0.38 | 0.42 \| 0.42 | 0.30 \| 0.35 | 158 | 86 |
| **Fruits** | 0.60 \| 0.63 | 0.42 \| 0.48 | 0.42 \| 0.47 | 0.32 \| 0.39 | 0.38 \| 0.47 | 0.55 \| 0.56 | 0.63 \| 0.67 | 181 | 99 |
| **iSUN** | 0.56 \| 0.51 | 0.35 \| 0.36 | 0.34 \| 0.35 | 0.31 \| 0.32 | 0.33 \| 0.34 | 0.28 \| 0.25 | 0.31 \| 0.27 | 187 | 102 |
| **Jigsaw on Training Set** | 0.60 \| 0.58 | 0.45 \| 0.47 | 0.44 \| 0.47 | 0.36 \| 0.39 | 0.43 \| 0.46 | 0.52 \| 0.52 | 0.65 \| 0.64 | 159 | 88 |
| **LSUN-Crop** | 0.31 \| 0.33 | 0.09 \| 0.12 | 0.08 \| 0.11 | 0.11 \| 0.15 | 0.09 \| 0.14 | 0.10 \| 0.10 | 0.31 \| 0.36 | 178 | 99 |
| **LSUN-Resize** | 0.52 \| 0.46 | 0.29 \| 0.30 | 0.28 \| 0.29 | 0.26 \| 0.26 | 0.28 \| 0.29 | 0.22 \| 0.20 | 0.30 \| 0.26 | 193 | 107 |
| **Office-Home Art** | 0.55 \| 0.56 | 0.37 \| 0.40 | 0.36 \| 0.39 | 0.32 \| 0.35 | 0.34 \| 0.39 | 0.40 \| 0.40 | 0.50 \| 0.51 | 151 | 81 |
| **Office-Home Clipart** | 0.55 \| 0.51 | 0.34 \| 0.38 | 0.34 \| 0.38 | 0.34 \| 0.36 | 0.34 \| 0.37 | 0.20 \| 0.22 | 0.12 \| 0.16 | 182 | 99 |
| **Office-Home Product** | 0.60 \| 0.56 | 0.42 \| 0.46 | 0.42 \| 0.45 | 0.44 \| 0.45 | 0.43 \| 0.44 | 0.34 \| 0.36 | 0.37 \| 0.38 | 170 | 93 |
| **Office-Home Real** | 0.57 \| 0.54 | 0.40 \| 0.43 | 0.39 \| 0.42 | 0.37 \| 0.39 | 0.39 \| 0.41 | 0.37 \| 0.38 | 0.41 \| 0.42 | 163 | 88 |
| **PACS Photo** | 0.72 \| 0.71 | 0.62 \| 0.62 | 0.62 \| 0.62 | 0.60 \| 0.62 | 0.61 \| 0.63 | 0.68 \| 0.67 | 0.83 \| 0.82 | 135 | 75 |
| **PACS Art** | 0.58 \| 0.59 | 0.41 \| 0.43 | 0.41 \| 0.43 | 0.36 \| 0.38 | 0.38 \| 0.41 | 0.48 \| 0.47 | 0.59 \| 0.59 | 138 | 75 |
| **PACS Cartoon** | 0.58 \| 0.59 | 0.37 \| 0.40 | 0.37 \| 0.39 | 0.35 \| 0.36 | 0.36 \| 0.40 | 0.31 \| 0.31 | 0.48 \| 0.52 | 150 | 81 |
| **PACS Sketch** | 0.55 \| 0.53 | 0.26 \| 0.30 | 0.25 \| 0.28 | 0.23 \| 0.26 | 0.27 \| 0.30 | 0.18 \| 0.20 | 0.55 \| 0.60 | 178 | 99 |
| **Place365** | 0.59 \| 0.55 | 0.40 \| 0.41 | 0.40 \| 0.41 | 0.38 \| 0.38 | 0.39 \| 0.40 | 0.47 \| 0.47 | 0.69 \| 0.65 | 134 | 72 |
| **SVHN** | 0.48 \| 0.56 | 0.35 \| 0.41 | 0.35 \| 0.41 | 0.25 \| 0.34 | 0.31 \| 0.37 | 0.44 \| 0.46 | 0.16 \| 0.20 | 214 | 122 |
| **Texture** | 0.60 \| 0.62 | 0.52 \| 0.57 | 0.53 \| 0.57 | 0.44 \| 0.51 | 0.51 \| 0.58 | 0.55 \| 0.54 | 0.17 \| 0.18 | 160 | 85 |
| **Mean Score** | 0.57 \| 0.57 | 0.39 \| 0.42 | 0.38 \| 0.41 | 0.35 \| 0.38 | 0.37 \| 0.41 | 0.40 \| 0.40 | 0.45 \| 0.47 | | |

*The "Statistical Distances" are real-valued scalars measuring the distances between the logits from test-set ID and OoD examples. More details can be found in the appendix.*

Table 4. Pearson Correlations between the Missed Detection Rates on OoD Data and the Corresponding Statistical Distances for CIFAR10-WideResNet Network.

| | **Wasserstein** | | **Energy** | |
|---|---|---|---|---|
| | ONE | MULTI | ONE | MULTI |
| **Max-softmax** | -0.54 | -0.47 | -0.55 | -0.46 |
| **Max-logit** | -0.57 | -0.51 | -0.56 | -0.51 |
| **Energy** | -0.57 | -0.51 | -0.56 | -0.51 |
| $k$**-NN** | -0.66 | -0.60 | -0.65 | -0.59 |
| **OC-SVM** | -0.60 | -0.57 | -0.59 | -0.56 |
| **ODIN** | -0.57 | -0.56 | -0.55 | -0.54 |
| **Mahalanobis** | -0.72 | -0.70 | -0.68 | -0.65 |

*al*. [31] showed how to alleviate this issue by distinguishing between the background and the semantics in the generative model. Zhang *et al*. [41] explained the phenomenon that flow based models can assign higher likelihood to OoD samples, but always generate ID images. Tonin *et al*. [35] proposed an energy-based unsupervised detection method without access to class labels.

## 5. Conclusion

In this paper, we have addressed an issue that relates to the thresholding strategy used in many state-of-the-art OoD detection algorithms. Despite the simplicity of our multi-threshold solution, our main contribution is to identify the problem and to raise people's awareness about the adversarial effect of label shift (and distribution shift in general) in the context of OoD detection. This issue, to our best knowledge, has has never been identified or discussed in previous literature. Our solution has addressed this issue in a simple and effective way, making it amenable to real-world applications.

It is worth pointing out one limitation of our proposed approach: when there are very few samples for a particular class in the training and validation sets, it will be difficult to set a meaningful cutoff threshold $\tau_i$. Since real-world training data can often be long-tailed, it is possible that the data for some classes are very scarce. We leave this challenge to our future work. Beyond label shift, we also plan to study how other types of domain shift (such as concept shift and covariate shift) affect OoD detection algorithms and how to address such challenges.

## Acknowledgements

This research was supported by Singapore SinBerBEST project and C3.ai. The first author would like to thank also Fondazione CEUR (Centro Europeo Università e Ricerca) for supporting his research through the program *Giovani Ricercatori*.

## References

[1] Corrado Alessio. Animals-10, Dec 2019. 5, 11, 12

[2] Kaneswaran Anantharajah, ZongYuan Ge, Christopher McCool, Simon Denman, Clinton B Fookes, Peter Corke, Dian W Tjondronegoro, and Sridha Sridharan. Local inter-session variability modelling for object classification. In *Winter Conference on Applications of Computer Vision (WACV), 2013 IEEE Conference on*, 2014. 5, 12

[3] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 15–27. Springer, 2002. 11

[4] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*, 2020. 7

[5] Hyunsun Choi, Eric Jang, and Alexander A Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018. 7

[6] Spencer Churchill and Brian Chao. Anime face dataset, 2019. 5, 12

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5, 12

[8] 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, 2009. 5, 12

[9] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019. 2, 7

[10] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. 2

[11] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks, 2018. 1, 2, 7

[12] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure, 2019. 1, 2, 7

[13] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013. 5, 12

[14] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized ODIN: Detecting out-of-distribution image without learning from out-of-distribution data, 2020. 2, 7, 11

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 5

[16] Israr Hussain, Qianhua He, Zhuliang Chen, and Wei Xie. Fruit recognition dataset, jul 2018. 5, 12

[17] K J Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection, 2021. 1

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 12

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc. 5

[20] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks, 2018. 1, 2, 5, 7, 11

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization, 2017. 5, 12

[22] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks, 2020. 1, 2, 5, 7, 11

[23] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. MOOD: Multi-level out-of-distribution detection. *arXiv preprint arXiv:2104.14726*, 2021. 7

[24] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018. 3

[25] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection, 2020. 1, 2, 3, 5, 7, 11

[26] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 5, 12

[27] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 1

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 11

[30] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *ACM Sigmod Record*, volume 29, pages 427–438. ACM, 2000. 11

[31] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A DePristo, Joshua V Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *arXiv preprint arXiv:1906.02845*, 2019. 8

[32] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for

novelty detection. NIPS'99, page 582–588, Cambridge, MA, USA, 1999. MIT Press. 11

[33] Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. Negative data augmentation, 2021. 5, 11, 12

[34] Gábor J Székely. E-statistics: The energy of statistical samples. *Bowling Green State University, Department of Mathematics and Statistics Technical Report*, 3(05):1–18, 2003. 6

[35] Francesco Tonin, Arun Pandey, Panagiotis Patrinos, and Johan AK Suykens. Unsupervised energy-based out-of-distribution detection using stiefel-restricted kernel machine. *arXiv preprint arXiv:2102.08443*, 2021. 8

[36] SS Vallender. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, 18(4):784–786, 1974. 6

[37] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 5, 12

[38] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R. Kulkarni, and Jianxiong Xiao. TurkerGaze: Crowdsourcing saliency with webcam based eye tracking, 2015. 5, 12

[39] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop, 2016. 5, 12

[40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017. 5

[41] Yufeng Zhang, Wanwei Liu, Zhenbang Chen, Ji Wang, Zhiming Liu, Kenli Li, Hongmei Wei, and Zuoning Chen. Out-of-distribution detection with distance guarantee in deep generative models. *arXiv preprint arXiv:2002.03328*, 2020. 8

[42] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019. 11

[43] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5, 12