Detecting Out-of-Distribution Examples with Gram Matrices

Chandramouli S. Sastry [1] **Sageev Oore** [1]

Abstract

When presented with Out-of-Distribution (OOD) examples, deep neural networks yield confident, incorrect predictions; detecting OOD examples is challenging, and the potential risks are high. In this paper, we propose to detect OOD examples by identifying inconsistencies between activity patterns and predicted class. We find that characterizing activity patterns by Gram matrices and identifying anomalies in Gram matrix values can yield high OOD detection rates. We identify anomalies in the Gram matrices by simply comparing each value with its respective range observed over the training data. Unlike many approaches, this can be used with any pre-trained softmax classifier and neither requires access to OOD data for fine-tuning hyperparameters, nor does it require OOD access for inferring parameters. We empirically demonstrate applicability across a variety of architectures and vision datasets and, for the important and surprisingly hard task of detecting far out-of-distribution examples, it generally performs better than or equal to state-of-the-art OOD detection methods (including those that do assume access to OOD examples).

1. Introduction

Even when deep neural networks (DNNs) achieve impressive accuracy on challenging tasks, they do not always visibly falter on misclassified examples: in those cases they can often make predictions that are both very confident and completely incorrect. Yet, predictive uncertainty is essential in real-world contexts tolerating minimal error margins such as autonomous vehicle control and medical, financial and legal fields.

In this work, we focus on flagging test examples that do not contain any of the classes modeled in the train distribution. Such examples are often referred to as being *out-of-distribution* (OOD), and while their existence has been well-known for some time, the challenges of identifying them and a baseline method to do so in a variety of tasks such as image classification, text classification, and speech recognition were presented by Hendrycks and Gimpel (2017). Recently, Nalisnick et al. (2019a) identified a similar problem with generative models: they demonstrate that flow-based models, VAEs, and PixelCNNs cannot distinguish images of common objects such as dogs, trucks, and horses (i.e. CIFAR-10) from those of house numbers (i.e. SVHN), assigning a higher likelihood to the latter when the model is trained on the former. They report similar findings across several other pairs of popular image datasets.

While we might expect neural networks to respond differently to OOD examples than to in-distribution (ID) examples, exactly where and how to find these differences in activity patterns is not at all clear. Hendrycks and Gimpel (2017) and others (Nguyen et al., 2015; Yu et al., 2011) showed that looking at the maximal softmax value is insufficient. In Section 2 we describe some other recent approaches to this problem. In this work, we find that characterizing activity patterns with Gram matrices—and their extensions that we introduce—lets us quantify anomalies to allow state-of-the-art (SOTA) detection rates on OOD examples.

Intuition. We identify out-of-distribution examples by jointly considering the class assigned at the output layer and the activity patterns in the intermediate layers. For example, if an image is predicted to be a dog, yet the intermediate activity patterns are somehow atypical of those seen by the network for other dog images during training, then that is a strong indicator of an OOD example. This effectively allows us to detect incongruence between *the prediction made by the network* and *the path by which it arrived at that prediction*. To describe the activation path we need to describe the intermediate feature representations, and as a proxy for describing those representations, we use Gram Matrices as they not only describe the activations at the individual channels but also summarize the pairwise interactions between the channels.

Strengths. Unlike those previous works that assume access to OOD examples and train an auxiliary classifier for

[1]Dalhousie University/ Vector Institute. Correspondence to: Chandramouli Shama Sastry <chandramouli.sastry@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

identifying anomalous activity patterns, our method finds differences in activity patterns without requiring access to any OOD examples, and it works across architectures. We hope this will also help further our understanding of how neural networks respond differently to OOD examples *in general*, not just how a particular network responds to examples coming from a particular distribution.

Contributions. This work includes the following contributions:

1. We extend Gram matrices to compute effective feature correlations.
2. Using the p^{th}-order Gram matrices, we present a new technique for computing class-conditional anomalies in activity patterns.
3. We evaluate this technique on OOD detection, testing on
   - competitive architectures: DenseNet, ResNet;
   - benchmark OOD datasets including: CIFAR-10, CIFAR-100, SVHN, TinyImageNet, LSUN and iSUN.

   Note that no adversarial or other re-training is required; *we can use pre-trained models*.
4. Crucially, *our method does not require access to OOD samples* for tuning hyperparameters or for training auxiliary models.
5. We report results which, for the challenging and important cases of far-from-distribution examples, are generally better than or equal to the state-of-the-art method for OOD detection that does require access to OOD examples. [1]

2. Related Work

Previous work which aims to improve OOD detection can be roughly grouped by several themes:

Bayesian Neural Networks. A nice early Bayesian approach (Gal and Ghahramani, 2016) estimates predictive uncertainty by using an ensemble of sub-networks instantiated by applying dropout at test time. As opposed to implicitly learning a distribution over the predictions by learning a distribution over the weights, Chen et al. (2019) and Malinin and Gales (2018) explicitly parameterize a Dirichlet distribution over the output class distributions using DNNs in order to obtain a better estimate of predictive uncertainty; the main differences between these methods is that Chen et al. (2019) use ELBO, which only requires the in-distribution dataset for training whereas Malinin and Gales (2018) use a contrastive loss which requires access to (optionally synthetic) OOD examples.

Using any pre-trained softmax deep neural network with OOD examples. Lee et al. (2018b)—to the best of our knowledge, the current SOTA technique by a significant margin—compute the Mahalanobis distance between the test sample's feature representations and the class-conditional gaussian distribution at each layer; they then represent each sample as a vector of the Mahalanobis distances, and finally train a logistic regression detector on these representations to identify OOD examples. Another technique in this category is ODIN (Liang et al., 2018): they use a mix of temperature scaling at the softmax layer and input perturbations to achieve better results. In fact, both Lee et al. (2018b) and Liang et al. (2018) add small input perturbations to achieve better results; the former do so to increase the confidence score, while the latter do so to increase the softmax score. Quintanilha et al. (2019) achieve results comparable to that of Lee et al. (2018b) by training a logistic regression detector that looks at the means and standard deviations of various channels activations. Unlike the previous two techniques, Quintanilha et al. (2019) achieves comparable results even without the use of input perturbations, which allows it to be applicable to non-continuous domains. Our work, too, does not involve input perturbations.

Recently, Abdelzad et al. (2019) propose to detect OOD examples by training a one-class detector over the representations of an intermediate layer, *chosen* for each OOD detection task.

All of these techniques depend on OOD examples for fine-tuning hyperparameters (Liang et al., 2018; Abdelzad et al., 2019) or for training auxiliary OOD classifiers (Lee et al. (2018b); Quintanilha et al. (2019)). Furthermore, these classifiers neither transfer between one non-training distribution and another, nor do they transfer between networks, so separate classifiers must be trained for each (In-Distribution, OOD, Architecture) triplet. In many real-world applications, we may not be able to assume advance access to all possible OOD distributions. Motivated by this observation, our work does not require access to OOD samples.

Alternative Training Strategies. Lee et al. (2018a) jointly train a classifier, a generator and an adversarial discriminator such that the classifier produces a more uniform distribution on the boundary examples generated by the generator; they use OOD examples to fine-tune hyperparameters. DeVries and Taylor (2018) train neural networks with a multi-task loss for jointly learning to classify and estimate confidence. Shalev et al. (2018) use multiple semantic dense representations as the target instead of sparse one-hot

[1] The code is open-sourced at `https://github.com/VectorInstitute/gram-ood-detection`

vectors and use a cosine-similarity based measure for detecting OODs. Building on the idea proposed by Lee et al. (2018a), Hendrycks et al. (2019a) propose an *Outlier Exposure* (OE) technique. They regularize a softmax classifier to predict uniform distribution on (any) OOD distribution and show the resulting model can identify examples from unseen OOD distributions; this differs significantly from previous works which used the same OOD distributions for both training and testing. Unlike other methods, they retain the architecture of the classifier and introduce just one additional hyperparameter—the regularization rate—and also demonstrate that their model is quite robust to the choice of OOD examples chosen for the regularization. However, while the OE method is able to generalize across different non-training distributions, it understandably does not achieve the rates of Lee et al. (2018b) on most cases. In the same vein, Vernekar et al. (2019) propose a strategy to generate boundary OOD examples to train a classifer with a reject option. Similarly, Yu and Aizawa (2019) propose to train a two-head CNN on in-distribution data with different decision boundaries by encouraging a higher discrepancy in predictions on unlabeled OOD data. Concurrent with this work, Hsu et al. (2020) build upon Techapanurak and Okatani (2019) and propose to train a neural network which also learns a temperature scaling function that is applied to the logits before computing the softmax values; while also including a probabilistic interpretation, the authors demonstrate remarkable improvements in detecting out-of-distribution examples by using just in-distribution examples. Our technique is complementary to these techniques and can be applied to a classifier trained with any of these techniques to construct a more robust detector; in order to illustrate this, we include results on combining our method with Outlier Exposure in Section 4.2.

Golan and El-Yaniv (2018) show how self-supervised classifiers trained to predict geometrical transformations in the input image can be used for one-class OOD detection. Recently, Hendrycks et al. (2019b) make significant advances in detecting near-distribution outliers without having any knowledge of the exact out-of-distribution examples by using in-distribution examples in a self-supervised training setting.

Generative Models. Ren et al. (2019) hypothesize that stylistic factors might impact the likelihood assignment and propose to detect OOD examples by computing a likelihood ratio which depends on the semantic factors that remain after the dominant stylistic factors are cancelled out. On the other hand, Nalisnick et al. (2019b) argue that samples generated by a generative model reside in the typical set, which might not necessarily coincide with areas of high density. They demonstrate empirically that OOD examples can be identified by checking if an input resides in the typical set of the generative model. Unlike the standard experimental setting, they aim to identify distributional shift, which predicts if a batch of examples are OOD. Several other recent works (Huang et al., 2019; Serrà et al., 2019; Daxberger and Hernández-Lobato, 2019; Song et al., 2019; Choi and Chung, 2020; Serrà et al., 2020) also aim to solve these problems.

3. Extending Gram Matrices for Out-of-Distribution Detection

Overview In light of the above considerations, we are interested in proposing a method that does not require access to any OOD examples, that does not introduce hyperparameters that need tuning, and that works across architectures. Gram matrices can be used to compute pairwise feature correlations, and are often used in DNNs to encode stylistic attributes like textures and patterns (Gatys et al., 2016). We extend these matrices as will be described below, and then use them to compute class-conditional bounds of feature correlations at multiple layers of the network. Starting with a pre-trained network, we compute these bounds over only the training set, and then use them at test time to effectively discriminate between in-distribution samples and out-of-distribution samples. Unlike other SOTA algorithms, we do not need to "look" at any out-of-distribution samples to tune any parameters; the only tuning required is that of a normalizing factor, which we compute using a randomly-selected validation partition of the (in-distribution) test set.

Notation If the considered deep convolutional network has L layers and the l^{th} layer has n_l channels, we consider feature co-occurrences between the $\sum_{1<=l<=L} \frac{n_l*(n_l+1)}{2}$ pairs of feature-maps. (Note that by "layer" we refer to any set of values obtained immediately after applying convolution or activation functions.) We use the following notation:

$F_l(D)$	The feature map at the l-th layer for input image D; when referring to an arbitrary image D, we just write F_l. It can be stored in a matrix of dimensions $n_l \times p_l$, where n_l is the number of channels at the l-th layer and p_l, the number of pixels per channel, is the height times the width of the feature map.
D_c , $f(D)$	The predicted class for input image D
Tr	The set of all train examples
Va	The set of all validation examples. 10% of the examples not used in training are randomly chosen as validation examples.

The set of all test examples is disjoint as usual from Tr and Va. We assume that *only the test set may contain out-of-distribution examples*.

Gram Matrices and Higher order Gram Matrices We summarize the activities at the l-th layer using the Gram matrix:

$$G_l = F_l F_l^\top \tag{1}$$

where F_l is an $n_l \times p_l$ matrix as defined above.

In order to compute Gram Matrices with more prominent activations of the feature maps, we define a *higher-order Gram matrix*, which we write G_l^p, to be a matrix computed identically to the regular Gram matrix, but where, instead of using a raw channel activation a, we use a^p, the p^{th} power of each activation. G_l^p is therefore computed using F_l^p, where the power of F_l is computed element-wise; in an effort to retain uniform scale across all orders of Gram matrices for a given layer, we compute the (element-wise) p-th root. The p-th order Gram matrix is thus computed as:

$$G_l^p = \left(F_l^p F_l^{p^\top}\right)^{\frac{1}{p}} \tag{2}$$

We show in Section 5 that higher p values help significantly in improving the OOD detectability. In our experiments, we limit the value of p to 10, as exponents beyond 10 are not worth the extra computation that is needed to avoid overflow errors[2].

The flattened upper (or lower) triangular matrix along with the diagonal entries is denoted as $\overline{G_l^p}$. The set of all orders of Gram matrices (in our case $\{1, \ldots 10\}$) to be considered is denoted by P. The schematic diaGram of the proposed algorithm is shown in Fig. 1 (in Appendix A).

Notably, while the Mahalanobis method describes each layer with just their channel-wise means, we use the higher-order Gram matrices to also include their higher-order raw moments; specifically, a Gram matrix of order p contains the $2p$-th moments of the individual channels along the diagonal and p-th moment of the hadamard product between the channels in its off-diagonal elements.

Preprocessing If we compute $\overline{G_l^p}$ for every layer l and every order $p \in P$, we obtain a total of $N_S = \sum_{p\in P}\sum_{l=1}^{L} \frac{1}{2} n_l(n_l+1)$ correlations for any image D. The preprocessing involves computing the class-specific minimum and maximum values for the correlations: for every class c, the minimum and maximum values for each of the N_S correlations are computed over all training examples D classified as c. We keep track of the minimum and maximum values of the N_S correlations for all the classes in 4-D arrays Mins and Maxs, each of the order $\left(C \times L \times |P| \times \max_{1\le l\le L} \frac{n_l(n_l+1)}{2}\right)$. Since each layer has different number of channels, the 4-th dimension must be large enough to accommodate the layer with the highest number of channels.

[2]The maximum activation values observed in the convolution layers of a ResNet trained on Cifar-10 (open-sourced by Lee et al. (2018b)) are 6.5 and 6.3 on train and test partitions.

Algorithm 1 Compute the minimum and maximum values of feature co-occurrences for each class, layer and order

Input:
C: Number of output classes
L: Number of Layers in entire network
P: Set of all orders of Gram Matrix to consider
Tr: The train data
Output:
Mins, Maxs

Mins[C][L][P]$\left[\max_{1\le l\le L} \frac{n_l(n_l+1)}{2}\right] \leftarrow \infty$ ▷ Stores the Mins for each class, layer and order
Maxs[C][L][P]$\left[\max_{1\le l\le L} \frac{n_l(n_l+1)}{2}\right] \leftarrow -\infty$ ▷ Stores the Maxs for each class, layer and order
for c in $[1, C]$ **do**
  $\text{Tr}_c = \{D \mid D \in \text{Tr}\ s.t.\ f(D) = c\}$ ▷ All the training examples predicted as c
  for D $\in \text{Tr}_c$ **do**
    for l in $[1, L]$ **do**
      for p in P **do**
        stat = $\overline{G_l^p(D)}$ ▷ The flattened upper triangular matrix
        for i in $\left[1, \frac{n_l(n_l+1)}{2}\right]$ **do**
          Mins$[c][l][p][i]$ = min(Mins$[c][l][p][i]$,stat$[i]$)
          Maxs$[c][l][p][i]$ = max(Maxs$[c][l][p][i]$,stat$[i]$)
return Mins, Maxs

Computing Layerwise Deviations Given the class-specific minimum and maximum values of the N_S feature correlations, we can compute the deviation of the test sample from the images seen at train time with respect to each of the layers. In order to account for the scale of values, we compute the deviation as the percentage change with respect to the maximum or minimum values of feature co-occurrences; the deviation of an observed correlation value from the minimum and maximum correlation values observed during train time can be computed as:

$$\delta(\text{min,max,value}) = \begin{cases} 0 & \text{if min } \le \text{ value} \le \text{max} \\ \frac{\text{min}-\text{value}}{|\text{min}|} & \text{if value} < \text{min} \\ \frac{\text{value}-\text{max}}{|\text{max}|} & \text{if value} > \text{max} \end{cases} \tag{3}$$

The deviation of a test image with respect to a given layer l is the sum total of the deviations with respect to each of the $\sum_{p\in P} \frac{1}{2} n_l(n_l+1)$ correlation values:

$$\delta_l(D) = \sum_{p=1}^{P} \sum_{i=1}^{\frac{1}{2}n_l(n_l+1)} \delta\left(\text{Mins}[D_c][l][p][i], \text{Maxs}[D_c][l][p][i], \overline{G_l^p(D)}[i]\right) \tag{4}$$

Total Deviation of a test image D ($\Delta(D)$), is computed by taking the sum total of the layerwise deviations ($\delta_l(D)$). However, the scale of layerwise deviations (δ_l) varies with each layer depending on the number of channels in the layer, number of pixels per channel and semantic information contained in the layer. Therefore, we normalize the deviations by dividing it by $\mathbb{E}_{\text{Va}}[\delta_l]$, the expected deviation at layer δ_l, computed using the validation data. Note that we use the

same normalizing factor irrespective of the class assigned.

$$\Delta(D) = \sum_{l=1}^{L} \frac{\delta_l(D)}{\mathbb{E}_{\text{Va}}\left[\delta_l\right]} \tag{5}$$

Additionally, we can take into account the maximum softmax probability (MSP) by dividing the above term with the MSP. This marginally improves the detection rate in some cases.

Threshold As is standard (Lee et al., 2018b), a threshold, τ, for discriminating between out-of-distribution data and in-distribution data is computed as the 95th percentile of the total deviations of test data ($\Delta(D)$). In other words, the threshold is computed so that 95% of test examples have deviations lesser than the threshold τ; the threshold-based discriminator can be formally written as:

$$\text{isOOD}(D) = \begin{cases} \text{True} & \text{if } \Delta(D) > \tau, \\ \text{False} & \text{if } \Delta(D) \leq \tau \end{cases} \tag{6}$$

Computational Complexity. For a single example, extracting upper diagonal elements takes $O\left(\sum_l \frac{n_l(n_l+1)}{2}\right)$. Thus, the pre-processing and test-time computation of deviations are both quadratic in the number of channels. In order to reduce computational time and make it linear in the number of channels, we can in fact compute deviations based on row-wise sums rather than individual elements. This would mean that the variable *stat*, defined in line 8 of Algorithm 1, would now contain row-wise sums of G_l^p instead of the flattened upper triangular matrix; the inner loop of Eq. 4 would loop over n_l elements instead of $\frac{1}{2}n_l(n_l+1)$ elements while also reducing the storage required for Mins and Maxs. In practise, we found that computing the anomalies this way yields differences of less than 0.5%, and usually imperceptible, so the results described in the next section were computed in this way.

Relationship to Mahalanobis detection Our algorithm is inspired by the Mahalanobis algorithm and bears several similarities. In fact, both the algorithms compute the total deviation Δ in terms of class-conditional layerwise deviations δ_l as:

$$\Delta = \sum_l \alpha_l \delta_l \tag{7}$$

One immediate difference is that while the Mahalanobis algorithm uses channel-wise means as descriptions of layerwise activations F_l, we use higher-order Gram matrices as a (more tightly coupled) description of F_l. More importantly, however, a core difference between the algorithms is in the way the scaling values α_l and the layerwise-deviations δ_l are computed: while the Mahalanobis algorithm uses Mahalanobis distances to compute the δ_l values and trains a logistic regression classifier to infer the α_l values, we compute the δ_l values as deviations from the extrema and α_l values using the validation dataset as described above.

	Can work with pre-trained Net?	Can work without knowledge of OOD test examples?
DPN (Malinin and Gales, 2018)	✗	✓
Semantic (Shalev et al., 2018)	✗	✓
Variational Dirichlet (Chen et al., 2019)	✗	✓
Mahalanobis (Lee et al., 2018b)	✓	✗
ODIN (Liang et al., 2018)	✓	✗
OE (Hendrycks et al., 2019a)	✗	✓
Baseline (Hendrycks and Gimpel, 2017)	✓	✓
Ours	✓	✓

Table 1: List of closely related methods. Note: OE uses OOD examples during training, but unrelated to test

4. Experiments - Detecting OOD

In this section, we demonstrate the effectiveness of the proposed metric using competitive deep convolutional neural network architectures such as DenseNet and ResNet on various computer vision benchmark datasets such as: CIFAR-10, CIFAR-100, SVHN, TinyImageNet, LSUN and iSUN.

For fair comparison and to aid reproducibility, we use the pretrained ResNet (He et al., 2016) and DenseNet (Huang et al., 2017) models open-sourced by Lee et al. (2018b), i.e. ResNet34 and DenseNet3 models trained on CIFAR-10, CIFAR-100 and SVHN datasets. For each of these models, we considered the corresponding test partitions as the in-distribution (positive) examples. For CIFAR-10 and CIFAR-100, we considered the out-of-distribution datasets used by Lee et al. (2018b): TinyImagenet, LSUN and SVHN. Additionally, we also considered the iSUN dataset. For ResNet and DenseNet models trained on SVHN, we considered CIFAR-10 dataset as the third OOD dataset. Details on these datasets are available in Appendix B.

We benchmark our algorithm with the works listed in Table 1 using the following metrics:

1. **TNR@95TPR** is the probability that an OOD (negative) example is correctly identified when the true positive rate (TPR) is as high as 95%. TPR can be computed as $TPR = TP/(TP + FN)$, where TP and FN denote True Positive and False Negative respectively.
2. **Detection Accuracy** measures the maximum possible classification accuracy over all possible thresholds in discriminating between in-distribution and out-of-distribution examples. For those methods which assign a higher-score to the in-distribution examples, it can be calculated as $\max_\tau \{0.5P_{in}(f(x) \geq \tau) + 0.5P_{out}(f(x) < \tau)\}$; for those methods which assign a lower score to in-distribution examples, it can be calculated as $\max_\tau \{0.5P_{in}(f(x) \leq \tau) + 0.5P_{out}(f(x) > \tau)\}$.

In-dist (model)	OOD	TNR at TPR 95%	AUROC	Detection Acc.
		Baseline / ODIN / Mahalanobis / Ours		
CIFAR-10 (ResNet)	iSUN	44.6 / 73.2 / 97.8 / **99.3**	91.0 / 94.0 / 99.5 / **99.8**	85.0 / 86.5 / 96.7 / **98.1**
	LSUN (R)	49.8 / 82.1 / 98.8 / **99.6**	91.0 / 94.1 / 99.7 / **99.9**	85.3 / 86.7 / 97.7 / **98.6**
	LSUN (C)	48.6 / 62.0 / 81.3 / **89.8**	91.9 / 91.2 / 96.7 / **97.8**	86.3 / 82.4 / 90.5 / **92.6**
	TinyImgNet (R)	41.0 / 67.9 / 97.1 / **98.7**	91.0 / 94.0 / 99.5 / **99.7**	85.1 / 86.5 / 96.3 / **97.8**
	TinyImgNet (C)	46.4 / 68.7 / 92.0 / **96.7**	91.4 / 93.1 / 98.6 / **99.2**	85.4 / 85.2 / 93.9 / **96.1**
	SVHN	50.5 / 70.3 / 87.8 / **97.6**	89.9 / 96.7 / 99.1 / **99.5**	85.1 / 91.1 / 95.8 / **96.7**
	CIFAR-100	33.3 / **42.0** / 41.6 / 32.9	86.4 / 85.8 / **88.2** / 79.0	80.4 / 78.6 / **81.2** / 71.7
CIFAR-100 (ResNet)	iSUN	16.9 / 45.2 / 89.9 / **94.8**	75.8 / 85.5 / 97.9 / **98.8**	70.1 / 78.5 / 93.1 / **95.6**
	LSUN (R)	18.8 / 23.2 / 90.9 / **96.6**	75.8 / 85.6 / 98.2 / **99.2**	69.9 / 78.3 / 93.5 / **96.7**
	LSUN (C)	18.7 / 44.1 / **64.8** / **64.8**	75.5 / 82.7 / 92.0 / **92.1**	69.2 / 75.9 / 84.0 / **84.2**
	TinyImgNet (R)	20.4 / 36.1 / 90.9 / **94.8**	77.2 / 87.6 / 98.2 / **98.9**	70.8 / 80.1 / 93.3 / **95.0**
	TinyImgNet (C)	24.3 / 44.3 / 80.9 / **88.5**	79.7 / 85.4 / 96.3 / **97.7**	72.5 / 78.3 / 89.9 / **92.2**
	SVHN	20.3 / 62.7 / **91.9** / 80.8	79.5 / 93.9 / **98.4** / 96.0	73.2 / 88.0 / **93.7** / 89.6
	CIFAR-10	19.1 / 18.7 / **20.2** / 12.2	77.1 / 77.2 / **77.5** / 67.9	71.0 / 71.2 / **72.1** / 63.4
CIFAR-10 (DenseNet)	iSUN	62.5 / 93.2 / 95.3 / **99.0**	94.7 / 98.7 / 98.9 / **99.8**	89.2 / 94.3 / 95.2 / **97.9**
	LSUN (R)	66.6 / 96.2 / 97.2 / **99.5**	95.4 / 99.2 / 99.3 / **99.9**	90.3 / 95.7 / 96.3 / **98.6**
	LSUN (C)	51.8 / 70.6 / 48.2 / **88.4**	92.9 / 93.6 / 80.2 / **97.5**	86.9 / 86.4 / 75.6 / **92.0**
	TinyImgNet (R)	58.9 / 92.4 / 95.0 / **98.8**	94.1 / 98.5 / 98.8 / **99.7**	88.5 / 93.9 / 95.0 / **97.9**
	TinyImgNet (C)	56.7 / 87.0 / 84.2 / **96.7**	93.8 / 97.6 / 95.3 / **99.3**	88.1 / 92.3 / 89.9 / **96.1**
	SVHN	40.2 / 86.2 / 90.8 / **96.1**	89.9 / 95.5 / 98.1 / **99.1**	83.2 / 91.4 / 93.9 / **95.9**
	CIFAR-100	40.3 / **53.1** / 14.5 / 26.7	89.3 / **90.2** / 58.5 / 72.0	**82.9** / 82.7 / 57.2 / 67.3
CIFAR-100 (DenseNet)	iSUN	14.9 / 37.4 / 87.0 / **95.9**	69.5 / 84.5 / 97.4 / **99.0**	63.8 / 76.4 / 92.4 / **95.6**
	LSUN (R)	17.6 / 41.2 / 91.4 / **97.2**	70.8 / 85.5 / 98.0 / **99.3**	64.9 / 77.1 / 93.9 / **96.4**
	LSUN (C)	28.6 / 57.8 / 42.1 / **65.5**	80.2 / **91.4** / 81.7 / **91.4**	72.7 / **83.3** / 74.0 / **83.6**
	TinyImgNet (R)	17.6 / 42.6 / 86.6 / **95.7**	71.7 / 85.2 / 97.4 / **99.0**	65.7 / 77.0 / 92.2 / **95.5**
	TinyImgNet (C)	24.6 / 51.0 / 60.1 / **89.0**	76.2 / 88.3 / 88.8 / **97.7**	69.0 / 80.2 / 81.6 / **92.5**
	SVHN	26.7 / 70.6 / 82.5 / **89.3**	82.7 / 93.8 / 97.2 / **97.3**	75.6 / 86.6 / 91.5 / **92.4**
	CIFAR-10	**18.9** / 16.8 / 7.7 / 10.6	**75.9** / 74.2 / 60.1 / 64.2	**69.7** / 68.6 / 57.8 / 60.4
SVHN (DenseNet)	iSUN	78.3 / 82.2 / **99.9** / **99.4**	94.4 / 94.7 / **99.9** / **99.8**	89.6 / 89.7 / **99.2** / 98.3
	LSUN (R)	77.1 / 81.1 / **99.9** / **99.5**	94.1 / 94.5 / **99.9** / **99.8**	89.1 / 89.2 / **99.3** / 98.6
	TinyImgNet (R)	79.8 / 84.1 / **99.9** / **99.1**	94.8 / 95.1 / **99.9** / **99.7**	90.2 / 90.4 / **98.9** / 97.9
	CIFAR-10	69.3 / 71.7 / **96.8** / 80.4	91.9 / 91.4 / **98.9** / 95.5	86.6 / 85.8 / **95.9** / 89.1
SVHN (ResNet)	iSUN	77.1 / 79.1 / **99.7** / **99.4**	92.2 / 91.4 / **99.8** / **99.8**	89.7 / 89.2 / **98.3** / **98.1**
	LSUN (R)	74.3 / 77.3 / **99.9** / **99.6**	91.6 / 89.4 / **99.9** / **99.8**	89.0 / 87.2 / **99.5** / 98.5
	TinyImgNet (R)	79.0 / 82.0 / **99.9** / **99.3**	93.5 / 92.0 / **99.9** / **99.7**	90.4 / 89.4 / **99.1** / 97.9
	CIFAR-10	78.3 / 79.8 / **98.4** / 85.8	92.9 / 92.1 / **99.3** / 97.3	90.0 / 89.4 / **96.9** / 92.0

Table 2: Comparison of OOD Detection Performance for all combinations of model architecture and training dataset are shown. The hyperparameters of *ODIN* and the hyperparameters and parameters of *Mahalanobis* are tuned using a random sample of the OOD dataset.

3. **AUROC** is the measure of the area under the plot of TPR vs FPR. For example, for those methods which assign a higher score to the in-distribution examples, this measures the probability that an OOD example is assigned a lower score than an in-distribution example.

Experimental setup: We use a pre-trained network to extract class-specific minimum and maximum correlation values for all pairs of features across all orders of Gram matrices. Subsequently, the total deviation is computed for each example following Eq. 5. Since the total deviation values depend on the randomly selected validation examples, we repeat the experiment 10 times to get a reliable estimate of the performance. The OOD detection performance for several combinations of model architecture, in-distribution dataset and out-of-distribution dataset are shown in Table 2. The results for Outlier Exposure (OE) are available in Table 3; some more results for OE and the results for DPN, Variational Dirichlet and Semantic are available in Appendix E.1 and Appendix E.2 respectively.

The results of Table 2 show that at a glance, over a total of 32 combinations of model architecture/in-distribution-dataset/far-out-of-distribution-datasets, the proposed method outperforms the previous competing methods in 22 of them, is on par in 7 of them, and gives second highest results on 3 of them[3]. Furthermore, it does so *without requiring access to samples from the OOD dataset*. If the hyperparameters and/or parameters of *Mahalanobis* and *ODIN* algorithms are fine-tuned using FGSM adversarial examples instead of the real OOD examples, their performance decreases. We also observe that our performance is similar for both architectures.

We also performed experiments with fully-connected networks by using three different MLP architectures trained on MNIST; Fashion-MNIST (Xiao et al., 2017) and KMNIST (Clanuwat et al., 2018) were considered as the out-of-distribution datasets (Results are provided in Appendix E.3).

4.1. Ablation Tests

The above results are obtained when we consider all elements of Gram matrix (Algorithm 1), compute deviations from extrema (Eq 3) and finally, compute the total deviation with normalized layerwise deviations (Eq 5). Additionally,

[3]This is based on the TNR at TPR 95% value; AUROC and Detection Accuracy results are comparable.

In-dist	Mean TNR @ TPR95			
	OE Base	OE	Ours (Base)	Ours
CIFAR-10	65.1	90.5	51.7625	**98.7**
CIFAR-100	37.3	61.5	19.15	**93.4**
SVHN	93.7	**99.9**	76.65	95.2

Table 3: Comparison of results with OE (Hendrycks et al., 2019a). Since OE uses a different model from ours, we also report the corresponding baseline accuracy. We extract the mean TNR @ TPR95 for our technique by considering both ResNet and DenseNet models. Some more results are available in Appendix E.1.

all layers and all orders were considered. We perform ablation experiments that aim to answer the following questions:

Q1 What if strictly diagonal elements or strictly off-diagonal elements are considered instead of complete Gram Matrix?

Q2 What happens if the deviation is computed from the mean instead of the extrema?

Q3 What happens if we do not normalize the layerwise deviations when aggregating them?

Q4 Which layer representations are most useful?

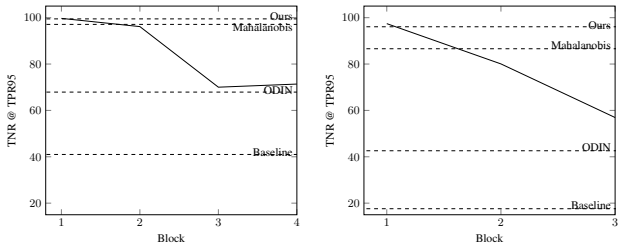
Q5 Which orders of Gram matrices are most useful?

As the answers to Q4 and Q5 were found to be independent of each other and Q1-Q3, we consider all layers and orders when answering Q1-Q3.

Q1, Q2, Q3. **Matrix feature set / deviation metric / aggregation scheme.** In order to answer Q1-Q3, we conduct 12 experiments: 3 choices for Q1 $\times$ 2 choices for Q2 $\times$ 2 choices for Q3. As a broad summary, we find that, while there is no single rule that is unbroken by an exception, our proposed combination—i.e. using the complete Gram matrix, using the min/max metric, and using normalization as in Eq 5—is generally more robust than any of the other combinations that we tried. More details on the ablation tests and discussions are available in Appendix C.

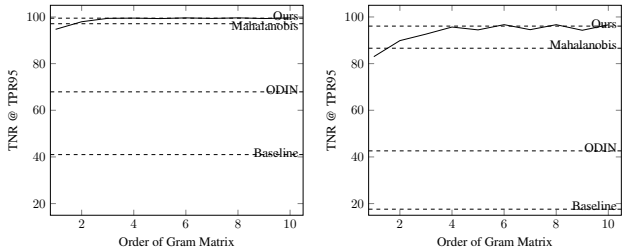
In answering Q4 and Q5, we consider the most robust setting that we identified by answering Q1-Q3.

Q4 **Which layer representations are most useful?** To explore this, we construct detectors which use correlations derived from just one residual or dense block at a time while considering all orders of Gram matrices. Representative results are shown in Figure 1. We consider all combinations of model/in-distribution/out-of-distribution-dataset, and generally find that the lower level representations are much more informative in discriminating between in-distribution and out-of-distribution datasets. However, the difference in detective power depends on the in-distribution dataset considered: for example, the difference in detective power between higher-level representations and lower-level representations is bigger for Cifar-100 than for Cifar-10. More graphs are available in Appendix C.2.



(a) ResNet/CIFAR-10 vs Tiny Imagenet (b) DenseNet/CIFAR-100 vs Tiny ImageNet

Figure 1: Significance of depth: The TNR@TPR95 is computed by constructing detectors which make use of all the Gram matrices but consider only one residual or dense block at a time. ResNet32 has 4 residual blocks and DenseNet3 has 3 dense blocks.



(a) ResNet/CIFAR-10 vs Tiny Imagenet (b) DenseNet/CIFAR-100 vs Tiny ImageNet

Figure 2: The importance of higher order Gram matrices: The TNR@TPR95 is computed by constructing detectors which make use of only one of the Gram matrices but consider all layers.

Q5 **Which orders of Gram matrices are most useful?** Here we construct detectors which make use of only one order of Gram matrix at a time, while computing correlations based on the representations of all residual and dense blocks. Representative results are shown in Figure 2. We again consider all combinations of model/in-distribution/out-of-distribution-dataset, and usually find that the higher order Gram matrices are more informative in discriminating between in-distribution and out-of-distribution datasets. Ignoring the variations at orders greater than 4, we find that the TNR @ 95TPR increases with higher orders and finally saturates. More graphs are available in Appendix C.1.

4.2. Limitations and Future Work

Near-distribution Outliers Traditionally, OOD detection algorithms have been evaluated by considering far-out-

In-dist (WRN 40-2)	OOD	TNR at TPR95		
		MSP	Ours	Ours + MSP
CIFAR-10	iSUN	98.3	98.9	**99.8**
	LSUN (R)	98.5	99.4	**99.8**
	LSUN (C)	98.0	89.5	**98.6**
	TinyImgNet (R)	93.9	98.5	**99.5**
	TinyImgNet (C)	95.2	95.9	**99.1**
	SVHN	98.0	97.6	**99.3**
	CIFAR-100	**73.9**	38.9	72.9
CIFAR-100	iSUN	50.9	**96.3**	95.6
	LSUN (R)	58.3	**98.4**	97.4
	LSUN (C)	69.5	69.7	**83.1**
	TinyImgNet (R)	36.1	**96.3**	92.8
	TinyImgNet (C)	41.6	**90.1**	87.1
	SVHN	56.2	84.8	**85.6**
	CIFAR-10	**17.4**	7.5	16.5

Table 4: Table shows results when our method is combined with OE. The experiment was conducted with WideResNet trained with outlier-exposure, open-sourced by Hendrycks et al. (2019a). MSP uses Maximum Softmax Probability; "Ours" refers to the metric Δ (Eq. 5); "Ours+MSP" is obtained by using $\Delta'(x) = \frac{\Delta(x)}{\max_{x \in \text{Va}} \Delta(x)} -$MSP. For each model/in-distribution/out-distribution triplet, the highest AUROC and DTACC numbers also correspond to the bold-faced columns; detailed results are available in Appendix D.

of-distribution datasets (e.g. as introduced in Hendrycks and Gimpel (2017) and expanded in Liang et al. (2018)). Hendrycks et al. (2019b) point out that the task of detecting near-distribution outliers is much more challenging than that of detecting far-distribution outliers. Following the convention in Hendrycks et al. (2019b), we consider CIFAR10 vs CIFAR100 tasks as near-distribution outliers and the rest as far-distribution outliers. We note that while near- and far- distribution outliers is an intuitive and potentially useful concept, the distinction between them is not well-defined. "Near-OOD" tends to be implicitly used to refer to those examples that are sufficiently similar-looking to the training set that it is not possible to detect them without learning more about the structure of in-distribution examples(Hendrycks et al., 2019b). Surprisingly, coming up with a generic algorithm for detecting these far-out-of-distribution examples have been challenging, prompting earlier papers to make assumptions about the source of out-of-distribution examples.

Our method is aimed at detecting far distribution classifiers, and, like nearly all other current far-OOD detectors, it does not perform very well for near OOD examples (refer to CIFAR-10 vs CIFAR-100 results in Table 2). To the best of our knowledge, Outlier Exposure Hendrycks et al. (2019a) and Classification augmented with rotation prediction Hendrycks et al. (2019b) yield the state-of-the-art detection rates on the CIFAR-10 vs CIFAR-100 tasks; impressively, the self-supervised technique achieves results comparable to OE. Since our method is complementary to modifications in training approaches, we show the result of combining our method with OE in Table 4: summarily, the combination of the two methods is more robust than either of the methods individually.

Appreciating the challenges and practical importance of identifying near-distribution outliers, Ahmed and Courville (2019) propose new benchmark tasks. In our experiments on these tasks, we found that our metric performed comparably to the other baseline methods and did not yield exceptional detection rates as observed when applied to far-distribution outlier detection. It remains to be explored if it is possible to identify the near-distribution outliers using any pre-trained network without explicit knowledge of the source of out-of-distribution examples – as is the case with our metric.

Connection to Style-transfer. The outstanding performance on the far-distribution outlier detection tasks and limited performance on near-distribution (and stylistically similar) outlier detection task of CIFAR-10 vs CIFAR-100 may appear related to the use of Gram matrices, which are themselves known to capture stylistic features. However, detailed experimental results do not clearly support this, as explained below.

First, we note that while style transfer using Gram matrices does not work well with ResNets and DenseNets, we get slightly better detection rate with ResNets and DenseNets than with VGGNets. Second, it has been empirically demonstrated that two images having similar textures have similar order-1 Gram matrices at lower layers. In contrast, we found that higher-order Gram matrices—key to the performance of our OOD system as indicated in Figure 2—can be different even when the corresponding order-1 Gram matrices are similar. Third, we consider Gram matrices at all layers and do not specifically compute and compare those typically associated with "style".

Gram matrices might contain more useful information than what the results portray: we suspect that the limited performance on this task is related to the choice of the metric used to compute δ_l and discuss it further below.

Improved estimation of δ_l Higher-order Gram matrices, in combination with an improved estimation of δ_l may—in future—help in obtaining better results than those contained in Table 2. For example, while we found the deviation from extrema gave overall strongest results, considering deviation from means gave an advantage on some specific cases. E.g., considering deviation from means improves the TNR@TPR95 for the SVHN vs CIFAR-10 task to 94% for both ResNet and DenseNet architectures (see Appendix C). It is possible that other density models could be used to obtain improved estimates of δ_l.

Implicit Confidence Prediction An important observation from Table 4 is that the softmax predictions (when trained with OE) and layerwise deviations contain mutually exclusive information; this is best observed in the CIFAR-100 vs LSUN (C) case. Additionally, the performance of

our metric using a network trained with outlier-exposure is comparable to the results obtained using a network trained with cross-entropy loss. Given the demonstrated feasibility of reliably detecting OOD examples using internal feature representations without making any assumptions over the data source[4], a natural future research question is whether modifications in training procedures and/or network architectures could implicitly yield reliable confidence estimates that are more sensitive to anomalies in intermediate feature representations.

5. Discussion and Conclusion

Beyond explicit OOD detection, this line of work may ultimately help better interpret neural networks' responses to OOD examples. As an initial exploratory example, we use our method to examine certain elements of the dataset. In Figure 3, we show some images from the standard MNIST dataset that had among the highest class-conditional deviation values, i.e. that were flagged as being potentially out-of-distribution. In Figure 4, we also explore the in-class variation among images that get flagged as OOD at individual layers of the network (pre-aggregation), by showing the top-10 anomalous plane images at each of four distinct layers. This suggests that different kinds of outlying examples might be detected at different layers. Finally, Figure 5 shows some (non-plane) CIFAR-10 images that we selected from among those flagged with unusual correlations. In Appendix F, we include additional images from SVHN.

Conclusion. Out-of-distribution detection is a challenging and important problem. We have proposed and reported on a relatively simple OOD detection method based on pairwise feature correlations. We conduct a variety of ablation tests and also show examples of unusual images detected in the standard data sets. Our method gives new state of the art detection results on far out-of-distribution examples without requiring access to anything other than the training data itself.

Figure 3: Some images from train and test partitions of MNIST which have unusual feature correlations as determined by our method.

[4]We would like to remind the reader that Lee et al. (2018b) and Quintanilha et al. (2019) originally demonstrated that internal feature representations do contain information about out-of-distribution examples; however, their technique involved training auxiliary classifiers, which in turn required them to make assumptions over the source of out-of-distribution data.

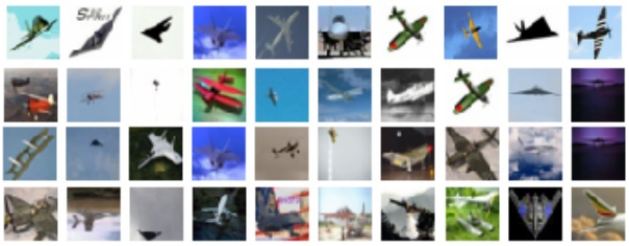


Figure 4: Each row shows 10 images, predicted to be planes, but having the highest deviations as identified by detectors constructed using 10th order Gram matrices computed over the representations of exactly one of the four ResNet blocks. I.e. the first row corresponds to the first ResNet block and the fourth row corresponds to the fourth ResNet block. In each row, images are sorted by their deviation values with the right-most images having the highest deviation values. Qualitatively, we have sometimes observed a progression wherein the outliers at the early layers tend to have a clearly defined shape, over a clear background, while the outliers at the later layers tend to be harder to discern.

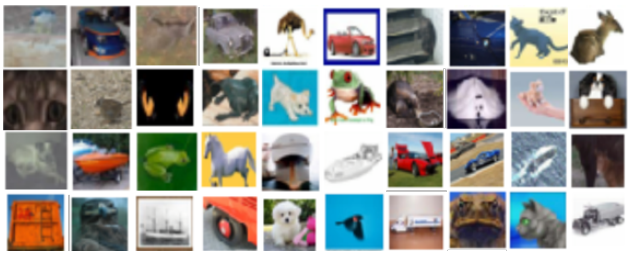


Figure 5: Some images selected from the test partition of CIFAR-10 which have unusual feature correlations as determined by our method. All classes apart from planes were considered.

Acknowledgements

We would like to sincerely acknowledge the insightful remarks by Scott Lowe, Jackson Wang, and Richard Zemel, that helped in the preparation of this manuscript. We thank Evangelos Milios for their support. We also thank our anonymous reviewers.

We thank the Canadian Institute for Advanced Research (CIFAR) and MITACS for their support. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute `www.vectorinstitute.ai/#partners`.

References

V. Abdelzad, K. Czarnecki, R. Salay, T. Denounden, S. Vernekar, and B. Phan. Detecting out-of-distribution inputs in deep neural networks using an early-layer output. *arXiv preprint arXiv:1910.10307*, 2019.

F. Ahmed and A. Courville. Detecting semantic anomalies.

arXiv preprint arXiv:1908.04388, 2019.

W. Chen, Y. Shen, W. Wang, and H. Jin. A variational dirichlet framework for out-of-distribution detection, 2019. URL https://openreview.net/forum?id=ByxmXnA9FQ.

S. Choi and S.-Y. Chung. Novelty detection via blurring. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByeNra4FDB.

T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep learning for classical japanese literature, 2018.

E. Daxberger and J. M. Hernández-Lobato. Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint arXiv:1912.05651*, 2019.

T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

Y. Gal and Z. Ghahramani. Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning - Volume 48*, ICML'16, pages 1050–1059, New York, NY, USA, June 19–24, 2016. JMLR.org. URL http://dl.acm.org/citation.cfm?id=3045390.3045502.

L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2414–2423, 2016. doi: 10.1109/CVPR.2016.265. URL https://doi.org/10.1109/CVPR.2016.265.

I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 9781–9791, 2018. URL http://papers.nips.cc/paper/8183-deep-anomaly-detection-using-geometric-transformations.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

D. Hendrycks and K. 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*, 2017. URL https://openreview.net/forum?id=Hkg4TI9xl.

D. Hendrycks, M. Mazeika, and T. 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*, 2019a. URL https://openreview.net/forum?id=HyxCxhRcY7.

D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019b.

Y.-C. Hsu, Y. Shen, H. Jin, and Z. 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 (CVPR)*, June 2020.

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 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269, 2017. doi: 10.1109/CVPR.2017.243. URL https://doi.org/10.1109/CVPR.2017.243.

Y. Huang, S. Dai, T. Nguyen, R. G. Baraniuk, and A. Anandkumar. Out-of-distribution detection using neural rendering generative models. *arXiv preprint arXiv:1907.04572*, 2019.

K. Lee, H. Lee, K. Lee, and J. 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*, 2018a. URL https://openreview.net/forum?id=ryiAv2xAZ.

K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 7167–7177, 2018b. URL http://papers.nips.cc/paper/7947-a-simple-unified-framework-for-detecting-out-of-distribution-samples-and-adversarial-attacks.

S. Liang, Y. 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, 2018. URL https://openreview.net/forum?id=H1VGkIxRZ.

A. Malinin and M. J. F. Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 7047–7058, 2018. URL http://papers.nips.cc/paper/7936-predictive-uncertainty-estimation-via-prior-networks.

E. T. Nalisnick, A. Matsukawa, Y. W. Teh, D. Görür, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019a. URL https://openreview.net/forum?id=H1xwNhCcYm.

E. T. Nalisnick, A. Matsukawa, Y. W. Teh, and B. Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using a test for typicality. *CoRR*, abs/1906.02994, 2019b. URL http://arxiv.org/abs/1906.02994.

A. M. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 427–436, 2015. doi: 10.1109/CVPR.2015.7298640. URL https://doi.org/10.1109/CVPR.2015.7298640.

I. M. Quintanilha, R. de M. E. Filho, J. Lezama, M. Delbracio, and L. O. Nunes. Detecting out-of-distribution samples using low-order deep features statistics, 2019. URL https://openreview.net/forum?id=rkgpCoRctm.

J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. A. DePristo, J. V. Dillon, and B. Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *CoRR*, abs/1906.02845, 2019. URL http://arxiv.org/abs/1906.02845.

J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. *arXiv preprint arXiv:1909.11480*, 2019.

J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SyxIWpVYvr.

G. Shalev, Y. Adi, and J. Keshet. Out-of-distribution detection using multiple semantic label representations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 7386–7396, 2018. URL http://papers.nips.cc/paper/7967-out-of-distribution-detection-using-multiple-semantic-label-representations.

J. Song, Y. Song, and S. Ermon. Unsupervised out-of-distribution detection with batch normalization. *arXiv preprint arXiv:1910.09115*, 2019.

E. Techapanurak and T. Okatani. Hyperparameter-free out-of-distribution detection using softmax of scaled cosine similarity. *arXiv:1905.10628*, 2019.

S. Vernekar, A. Gaurav, V. Abdelzad, T. Denouden, R. Salay, and K. Czarnecki. Out-of-distribution detection in classifiers via generation. *arXiv preprint arXiv:1910.04241*, 2019.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

D. Yu, J. Li, and L. Deng. Calibration of confidence measures in speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2461–2473, Nov 2011. doi: 10.1109/TASL.2011.2141988.

Q. Yu and K. Aizawa. Unsupervised out-of-distribution detection by maximum classifier discrepancy. *CoRR*, abs/1908.04951, 2019. URL http://arxiv.org/abs/1908.04951.