Deep Anomaly Detection with Outlier Exposure

Dan Hendrycks[*]
University of California, Berkeley
hendrycks@berkeley.edu

Mantas Mazeika
University of Chicago
mantas@ttic.edu

Thomas G. Dietterich
Oregon State University
tgd@oregonstate.edu

Abstract

When machine learning systems are deployed, it is important that anomalous inputs can be detected and controlled. Advances in deep learning have led to larger input spaces and more complex data distributions, magnifying the difficulty of distinguishing between in-distribution and out-of-distribution examples. At the same time, diverse data commonly used by deep learning systems, such as images or text, are available in enormous quantities. We propose leveraging this data to improve deep anomaly detection by training anomaly detectors against massive, diverse datasets of outliers, an approach we call Outlier Exposure (OE). In extensive experiments, we find that OE significantly improves the performance of existing anomaly detectors. We also find that OE improves classifier calibration in the presence of anomalous inputs.

1 Introduction

Detecting anomalous data is important in many machine learning system applications [Emmott et al., 2013]. At deployment time, when a data point comes from a different distribution than the learned distribution, it may be the case that this anomalous data is of interest to the operators of the system. This can occur in discovering novel astronomical phenomena, encountering unknown diseases, or detecting sensor failure. Also, proper handling of the anomalous data by the system is often not guaranteed. This is especially true of deep neural networks, as these systems often lack robustness to novel inputs.

Deep learning systems [Krizhevsky et al., 2012] can provide high performance in a variety of applications so long as the data seen at test time is similar to the training distribution. However, when this is not the case, deep neural network classifiers tend to give high-confidence predictions on anomalous test examples [Nguyen et al., 2015]. This opens the door to silent errors, and makes detecting the presence of anomalous data doubly important.

Several previous works seek to address these problems by giving deep neural network classifiers a means of assigning anomaly scores to inputs. These scores can then be used for detecting *out-of-distribution* examples [Hendrycks and Gimpel, 2017, DeVries and Taylor, 2018, Liang et al., 2018]. Intuitively, this approach models the distinction between *in-distribution* and out-of-distribution examples without necessarily modeling the full data distribution, and has been demonstrated to work surprisingly well for complex input spaces, such as images, text, and speech.

In this paper, we investigate the complementary research avenue of exposing anomaly detectors to the data that we would like them to mark as anomalous. The basic intuition is that exposing anomaly detectors to negative examples rather than solely positive examples can enable narrowing down the hypothesis class of what counts as in-distribution. We expose anomaly detectors to a diverse dataset of real, anomalous examples that we would like to be marked as anomalies. While the dataset of in-distribution examples may be limited in size, one can typically find massive quantities of out-of-distribution data. Moreover, it is not required that these data be labeled in any way, enabling the use of web-scraping as an effectively unlimited source of such data. We call this approach Outlier Exposure (OE).

[*] Work done while at OSU.

Through extensive experiments with a range of anomaly detectors, we find that OE consistently provides a substantial boost in performance. We explore the use of several large datasets for the use of OE. We also find that OE improves the calibration of neural network classifiers in the realistic setting where a fraction of the data is anomalous. Code is available at https://github.com/hendrycks/outlier-exposure.

2 Related Work

Deep Anomaly Detection. Hendrycks and Gimpel [2017] demonstrate that a deep pre-trained classifier has a lower maximum softmax probability on anomalous examples than in-distribution examples, so a classifier can double as a consistently useful anomaly detector. Building on this work, DeVries and Taylor [2018] attach an auxiliary branch onto a pre-trained classifier and derive a new anomaly score from this branch. Liang et al. [2018] present a method which can be used to improve performance of anomaly detectors using a softmax distribution. In particular, they make the the maximum softmax probability more effective by pre-processing input data with adversarial perturbations [Goodfellow et al., 2015]. The pre-processing parameters are tailored to each source of anomalies, while in this work we train our method without tuning parameters to fit specific types of anomalies. Lee et al. [2018] trains classifier with a GAN [Radford et al., 2016] concurrently, and the classifier is trained to have lower confidence on GAN samples. For each distribution of anomalies, they tune their classifier and GAN using samples from that anomaly distribution, as discussed in Appendix B of their work. de Vries et al. [2016], Bendale and Boult [2016], Subramanya et al. [2017] also encourage the model to have lower confidence on anomalous examples.

Utilizing Massive Datasets. Outlier Exposure uses a large database of examples to give the network better representations to detect anomalies. Torralba et al. [2011] pre-trains unsupervised deep models on a large database of web images for stronger features. Radford et al. [2017] train an unsupervised network on a large corpus of Amazon reviews for a month in order to obtain quality sentiment representations. Zeiler and Fergus [2014] find that pre-training a network on the large ImageNet database [Russakovsky et al., 2015] endows the network with general representations useful in myriad fine-tuning applications. In webly-supervised learning, Chen and Gupta [2015], Mahajan et al. [2018] show that representations learned from images scraped from the nigh unlimited source of search engines and photo-sharing websites improve object detection performance.

3 Outlier Exposure

We consider the problem of distinguishing whether a sample is anomalous or from a learned distribution called $\mathcal{D}_{\text{in}}$. Samples from $\mathcal{D}_{\text{in}}$ are called "in-distribution," and anomalous examples are said to be "out-of-distribution" or sampled from $\mathcal{D}_{\text{out}}$. In real applications, it may be difficult to know the distribution of anomalies. Thus we *expose* a given model to a large, diverse dataset of outliers $\mathcal{D}_{\text{out}}^{\text{OE}}$ so that the model learns how to process outliers. This method is called Outlier Exposure (OE). During OE we fine-tune a model with a refined optimization optimization objective so that there is a gap between the scores of in-distribution samples and anomalies. Once fine-tuned, the model should then be capable of detecting unseen anomalies from novel distributions $\mathcal{D}_{\text{out}}^{\text{test}}$.

We utilize OE in various contexts. For MNIST, SVHN, CIFAR-10, and CIFAR-100 as $\mathcal{D}_{\text{in}}$, we use the 80 Million Tiny Images dataset as $\mathcal{D}_{\text{out}}^{\text{OE}}$. We remove the subset of 80 Million Tiny Images [Torralba et al., 2008] corresponding to the CIFAR datasets. For Tiny ImageNet as $\mathcal{D}_{\text{in}}$, we use ImageNet-22K as $\mathcal{D}_{\text{out}}^{\text{OE}}$ with ImageNet-1K classes removed. For NLP experiments with Penn Treebank as $\mathcal{D}_{\text{in}}$, we use WikiText-2 [Merity et al., 2016] as $\mathcal{D}_{\text{out}}^{\text{OE}}$. For all OE experiments, we sample in-distribution samples from $\mathcal{D}_{\text{in}}$ and equally many samples from $\mathcal{D}_{\text{out}}^{\text{OE}}$. OE usually requires less than 15 minutes of fine-tuning on a single GPU.

4 Experiments

4.1 Evaluating Anomaly Detection Methods

To evaluate different anomaly detection methods, use three metrics: area under the receiver operating characteristic curve (AUROC), area under the precision-recall curve (AUPR), and the false positive rate at $N\%$ recall (FPRN). Since we aim to detect anomalous examples, we define anomalous examples as positive class in these measures. The AUROC and AUPR are holistic metrics since they summarize the performance of a detection method across several different anomaly score thresholds. The AUROC can be thought of as the probability that an anomalous example is given a higher anomaly score than a ordinary example [Davis and Goadrich, 2006]. Thus, a higher AUROC is better, and an uninformative detector has an AUROC of 50%. The AUPR is useful when anomalous examples are infrequent [Manning and Schütze, 1999], as it takes depends on the base rate of anomalies. During evaluation with these metrics, the base rate of anomalous to in-distribution examples in all our experiments is 1:5.

Whereas the previous two metrics represent the detection performance at various thresholds, the FPRN metric represents performance at one strict threshold. The FPRN metric is the probability that a negative (in-distribution) example is misclassified as positive (anomalous) when $N\%$ of anomalous examples are detected. Therefore, a lower FPRN is better. This desired recall level may be thought of as a safety threshold. Moreover, anomalous inputs may require human intervention, so a detector capturing most anomalies with few false positives is of high practical value.

4.2 Multiclass Classification

4.2.1 In-Distribution Datasets

We evaluate anomaly detectors on a wide range of datasets. Each evaluation consists of an in-distribution dataset $\mathcal{D}_{\text{in}}$ used to train a multiclass classifier a dataset of anomalous examples $\mathcal{D}_{\text{out}}^{\text{OE}}$. Below, we list the five in-distribution datasets used in the upcoming multiclass experiments.

MNIST. The MNIST dataset contains 28×28 grayscale images of the digits 0-9. The training set has $60,000$ images and the test set has $10,000$ images. We rescale the pixels to the interval $[0, 1]$.

SVHN. The SVHN dataset [Netzer et al., 2011] contains 32×32 color images of house numbers. There are ten classes comprised of the digits 0-9. The training set has 604,388 images, and the test set has $26,032$ images. For preprocessing, we rescale the pixels to be in the interval $[0, 1]$.

CIFAR. The two CIFAR [Krizhevsky and Hinton, 2009] datasets contain 32×32 natural color images. CIFAR-10 has ten classes while CIFAR-100 has 100. CIFAR-10 and CIFAR-100 classes are mutually exclusive but have similiarities. Both datasets have trucks, CIFAR-10 has "automobiles" and "trucks" but not CIFAR-100's "pickup truck" class. Both datasets have $50,000$ training images and 10,000 test images. Pixels are standardized by their channel's mean and standard deviation.

Tiny ImageNet. The Tiny ImageNet dataset [Johnson et al.] is a subset of the ImageNet [Russakovsky et al., 2015] dataset, restricted to 200 classes, and resized and cropped to 64×64 resolution. The dataset's images were cropped using bounding box information, unlike Downsampled ImageNet [Chrabaszcz et al., 2017]. The training set has $100,000$ images and test sets have $10,000$ images each. Pixels are standardized by the channel mean and standard deviation.

4.2.2 Anomalous Data

For each in-distribution dataset $\mathcal{D}_{\text{in}}$, we comprehensively evaluate anomaly detectors on synthetic and realistic anomalous distributions $\mathcal{D}_{\text{out}}^{\text{test}}$. We briefly describe these datasets below and leave SVHN and MNIST outlier dataset descriptions to the Supplementary Materials.

Gaussian anomalies have each pixel i.i.d. sampled from an isotropic Gaussian distribution. *Rademacher* anomalies are images where each pixel is -1 or 1 with equal probability, so each pixel is sampled from a symmetric Rademacher distribution. *Blobs* data consist in algorithmically generated amorphous shapes with definite edges. *Textures* is a dataset of describable textural images [Cimpoi et al., 2014]. *Places365* consists in images for scene recognition rather than object recognition [Zhou et al., 2017]. *LSUN* is another scene understanding dataset with fewer classes than Places365 [Yu et al., 2015]. *ImageNet* anomalous examples are taken from the 800 ImageNet-1K classes disjoint from Tiny ImageNet's 200 classes, and each image is cropped with bounding box information as in Tiny ImageNet. For *CIFAR-10* as $\mathcal{D}_{\text{in}}$, we use also *CIFAR-100* as $\mathcal{D}_{\text{out}}^{\text{test}}$ and vice versa; recall that the CIFAR-10 and CIFAR-100 classes do not overlap.

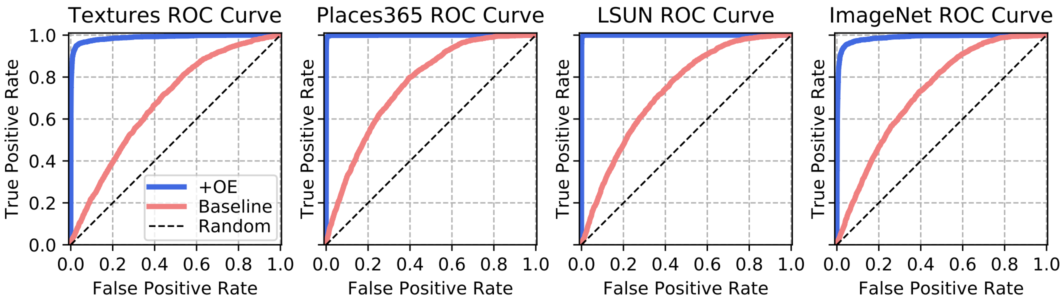

Figure 1: Tiny ImageNet ROC curves showing the baseline [Hendrycks and Gimpel, 2017] and OE.

4.2.3 Improving Multiclass Anomaly Detection Methods With Outlier Exposure

In what follows, we using existing anomaly detection techniques and enhance their performance with Outlier Exposure. For brevity, we supply MNIST and SVHN results in the Supplementary Materials. To ground the following discussion, we now establish notation. Let $x \in \mathcal{X}$ be a model input, and $y \in \mathcal{Y} = \{1, 2, \dots, k\}$ be a label. Represent the classifier with the function $p : \mathcal{X} \to \mathbb{R}^k$, such that $1^\mathsf{T} p(x) = 1$ and $p(x) \succeq 0$.

		FPR95 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	Baseline	+OE	Baseline	+OE	Baseline	+OE
CIFAR-10	Gaussian	29.6	0.7	86.6	99.7	42.1	97.1
	Rademacher	36.0	0.6	84.6	99.8	39.1	97.6
	Blobs	19.9	4.0	92.9	99.0	67.5	94.1
	Textures	36.8	8.0	89.4	98.5	60.8	93.7
	SVHN	18.1	8.3	93.3	98.0	68.8	88.7
	Places365	50.0	11.1	86.9	97.7	56.3	91.3
	LSUN	42.7	6.8	87.1	98.7	54.6	93.5
	CIFAR-100	42.7	26.4	85.4	94.3	50.8	80.1
	Mean	34.5	**8.2**	88.3	**98.2**	55.0	**92.0**
CIFAR-100	Gaussian	53.0	1.6	68.5	99.5	21.7	95.7
	Rademacher	43.6	0.1	77.7	100.0	29.2	99.9
	Blobs	34.4	3.9	88.9	99.1	54.8	95.0
	Textures	61.7	51.6	76.5	84.0	35.3	52.1
	SVHN	55.8	47.3	78.9	83.3	39.1	42.7
	Places365	66.8	46.0	74.8	88.1	34.1	62.6
	LSUN	66.3	50.9	73.6	86.6	31.1	60.1
	CIFAR-10	64.7	54.4	75.1	78.6	33.7	35.4
	Mean	55.8	**32.0**	76.7	**89.9**	34.9	**67.9**
Tiny ImageNet	Gaussian	48.3	18.0	67.8	93.9	21.3	66.0
	Rademacher	69.0	57.4	42.2	57.1	13.5	16.9
	Blobs	66.9	0.0	52.7	100.0	15.7	99.6
	Textures	77.3	13.8	68.0	97.7	28.3	94.6
	SVHN	47.1	0.2	83.3	99.9	40.3	99.3
	Places365	63.0	0.1	76.3	99.9	35.4	99.6
	LSUN	69.9	0.1	73.1	100.0	31.3	99.8
	ImageNet	66.5	20.0	73.7	96.9	30.9	91.5
	Mean	63.5	**13.7**	67.1	**93.2**	27.1	**83.4**

Table 1: Anomaly detection results for the maximum softmax probability (MSP) baseline and the MSP + OE detector. All values are percentages.

Maximum Softmax Probability. In this section, we improve on the maximum softmax probability baseline [Hendrycks and Gimpel, 2017] by exposing the classifier to outliers. To accomplish this, we fine-tune a pre-trained classifier p with outlier distribution $\mathcal{D}_{\text{out}}^{\text{OE}}$ by incentivizing the model to have a uniform posterior distribution on samples from $\mathcal{D}_{\text{out}}^{\text{OE}}$. In consequence, the training objective becomes $\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{in}}}[-\log p_y(x)] + 0.5\mathbb{E}_{x\sim\mathcal{D}_{\text{out}}^{\text{OE}}}[\text{KL}[U \| p(x)]]$, where U is the uniform distribution over k classes. The fine-tuning OE term has a coefficient of 0.5, which was determined early in experimentation with validation anomaly distributions on MNIST that are not included in the results. Like previous anomaly detection methods involving network fine-tuning, we choose a coefficient so impact on classification accuracy is negligible. This coefficient is kept constant, unlike [Liang et al., 2018, Lee et al., 2018] who, for each $\mathcal{D}_{\text{out}}^{\text{test}}$ tune a coefficient. Further architectural and training details are left to the Supplementary Materials. In the case of CIFAR-10, $\mathcal{D}_{\text{out}}^{\text{test}}$ is either "Gaussian," "Rademacher," . . ., or "CIFAR-100". We average the anomaly detection measurements across each possible each possible $\mathcal{D}_{\text{out}}^{\text{test}}$ setting to obtain the results in Table 1. The "Baseline" values are obtained using the maximum softmax probabilities (MSP) from the pre-trained network. The network's maximum softmax probabilities become more useful for anomaly detection after the network is fine-tuned with OE, as shown in Table 1 and Figure 1.

	Mean FPR95 $\downarrow$		Mean AUROC $\uparrow$		Mean AUPR $\uparrow$	
$\mathcal{D}_{\text{in}}$	$\max_c p_c$	KL$[U\|p]$	$\max_c p_c$	KL$[U\|p]$	$\max_c p_c$	KL$[U\|p]$
CIFAR-10	8.2	7.3	98.2	98.2	92.0	91.9
CIFAR-100	32.0	27.2	89.9	91.2	67.9	69.8
Tiny ImageNet	13.7	4.6	93.2	98.2	83.4	90.6

Table 2: Comparison between the Maximum Softmax Probability and KL$[U\|p]$ anomaly scoring methods fine-tuned with OE. Values are percentages and averaged across all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets.

Performance with Outlier Exposure can be improved further by using KL$[U\|p(x)]$ as the anomaly score instead of the maximum softmax probability, after the model is trained with OE. In Table 2, each detection measure across all $\mathcal{D}_{\text{out}}^{\text{test}}$ distributions are averaged into one summary statistic. This general performance improvement is most pronounced on datasets with larger classes. For instance, when $\mathcal{D}_{\text{out}}^{\text{test}}$ = Tiny ImageNet and $\mathcal{D}_{\text{out}}^{\text{test}}$ = Rademacher, swapping the maximum softmax probability score with the KL$[U\|p(x)]$ score increases the AUROC 57.1% to 89.5%.

	Mean FPR95 $\downarrow$			Mean AUROC $\uparrow$			Mean AUPR $\uparrow$		
$\mathcal{D}_{\text{in}}$	Baseline	Branch	+OE	Baseline	Branch	+OE	Baseline	Branch	+OE
CIFAR-10	49.3	38.7	20.8	84.4	86.9	93.7	51.9	48.6	66.6
CIFAR-100	55.6	47.9	42.0	77.6	81.2	85.5	36.5	44.4	54.7
Tiny ImageNet	64.3	66.9	20.1	65.3	63.4	90.6	30.3	25.7	75.2

Table 3: Comparison between the MSP, Confidence Branch, and Confidence Branch + OE anomaly detectors. The same network architecture is used for all three detectors. All values are percentages, and averaged across all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets.

Confidence Branch. A recently proposed anomaly detection technique DeVries and Taylor [2018] appends a logistic anomaly scoring branch $b : \mathcal{X} \to [0, 1]$ onto a classifier. This branch, trained only with samples from $\mathcal{D}_{\text{in}}$, estimates the network's confidence on any in-distribution examples and anomalies. The creators of this technique made their code publicly available, so we use their code to train new 40-4 Wide Residual Network classifiers. With these classifiers, we can fine-tune the confidence branch with Outlier Exposure. To fine-tune with OE, we simply add $0.5\mathbb{E}_{x\sim\mathcal{D}_{\text{out}}^{\text{OE}}}[\log b(x)]$ to the network's original optimization objective. In Table 3, the baseline values are derived from the maximum softmax probabilities produced by the classifier trained with DeVries and Taylor [2018]'s training code. The confidence branch improves over the MSP baseline, and after OE the confidence branch becomes even stronger at anomaly detection.

	Mean FPR95 $\downarrow$			Mean AUROC $\uparrow$			Mean AUPR $\uparrow$		
$\mathcal{D}_{\text{in}}$	Baseline	+GAN	+OE	Baseline	+GAN	+OE	Baseline	+GAN	+OE
CIFAR-10	32.3	37.3	11.8	88.1	89.6	97.2	51.1	59.0	88.5
CIFAR-100	66.6	66.2	49.0	67.2	69.3	77.9	27.4	33.0	44.7

Table 4: Comparison between the maximum softmax probability (MSP), MSP + GAN, and MSP + GAN + OE anomaly detectors. The same network architecture is used for all three detectors. All values are percentages and averaged across all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets.

Synthetic Outliers. Outlier Exposure leverages the simplicity of gathering large quantities of real data from the Internet. An important question to address is how this compares to using synthetic data for the outlier dataset. Lee et al. [2018] propose training a GAN to generate examples near a classifier's decision boundary. The classifier is encouraged to have a low maximum softmax probability on these synthetic examples. For CIFAR classifiers, they briefly mention that a GAN can be a more effective source of anomalies than datasets like SVHN. In contrast, we find that drawing anomalies from a massive, diverse database is sufficient for marked improvements in anomaly detection.

We train a 40-4 Wide Residual Network using Lee et al. [2018]'s the publicly available code and use its maximum softmax probabilities as our baseline. Another network trains with a GAN which produces anomalies, "+GAN" in the Table 4. While we use their code to train the network and GAN concurrently, we do not train and tune a network for each $\mathcal{D}_{\text{out}}^{\text{test}}$ distribution as they state in Appendix B of their paper. Instead, we use their code's default hyperparameters and one model encounters all tested $\mathcal{D}_{\text{out}}^{\text{test}}$ distributions. We do not evaluate on Tiny ImageNet since DCGANs cannot effectively model images of that scale and diversity. Last, we take the network pre-trained with a GAN and fine-tune it with OE to demonstrate the large gains from a real, massive, diverse database.

4.3 Density Estimation

Density estimation for the purpose of anomaly detection provides an alternative to the baseline maximum softmax probability detector. We show the potential for OE to greatly improve the performance of unsupervised anomaly detectors and its ability to make density estimates on anomalous data reasonable.

		FPR95 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	Baseline	+OE	Baseline	+OE	Baseline	+OE
CIFAR-10	Gaussian	0.0	0.0	100.0	100.0	100.0	99.6
	Rademacher	61.4	50.3	44.2	56.5	14.2	17.3
	Blobs	17.2	1.3	93.2	99.5	60.0	96.2
	Textures	96.8	48.9	69.4	88.8	40.9	70.0
	SVHN	98.8	86.9	15.8	75.8	9.7	60.0
	Places365	86.1	50.3	74.8	89.3	38.6	70.4
	LSUN	76.9	43.2	76.4	90.9	36.5	72.4
	CIFAR-100	96.1	89.8	52.4	68.5	19.0	41.9
Mean		66.6	**46.4**	65.8	**83.7**	39.9	**66.0**

Table 5: Anomaly detection results with a PixelCNN++ anomaly detector, and the same detector after applying OE. All values are percentages.

PixelCNN++. Autoregressive neural density estimation provides a powerful way to parametrize the probability density of image data. Although sampling from these architectures is slow, they allow for evaluating the probability density with a single forward pass through a CNN, making them promising candidates for anomaly detection. We use PixelCNN++ [Salimans et al., 2017] as a baseline anomaly detector, and we train it on CIFAR-10. This is then fine-tuned for 1 epoch using OE on 80 Million Tiny Images. OE is implemented with a margin loss over the log-likelihood difference between anomalous and in-distribution examples. This loss uses a margin of 1 per pixel. Results are shown in Table 5. For all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets, OE significantly improves results.

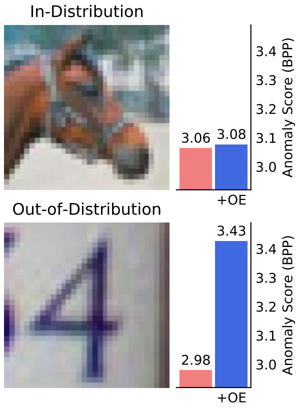

Figure 2: Anomaly scores (in bits per pixel) from PixelCNN++ on an image from CIFAR-10 and SVHN.

Language Modelling. We next explore using OE for detecting anomalous text. Natural language datasets are more amenable to density estimation than image datasets. We leverage this by using language models as baseline anomaly detectors. Outlier Exposure is implemented by adding the KL divergence to the uniform distribution to the loss on tokens from sequences in $\mathcal{D}_{\text{out}}^{\text{OE}}$.

For $\mathcal{D}_{\text{in}}$, we convert Penn Treebank into a language modeling corpus, split into sequences of length 70 for backpropagation in word-level models, and 150 in character-level models. We do not train with BPTT, in order to preserve consistency with the evaluation setting, in which retaining the hidden state would greatly simplify the task of anomaly detection. Accordingly, the anomaly detection task is to provide a score for each such 70- or 150-token sequence, with no prior information about the corpus of anomalous data. The $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets come from the English Web Treebank [Bies et al., 2012], which contains text from five different domains: Yahoo Answers, emails, newsgroups, product reviews, and weblogs.

		FPR90 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	Baseline	+OE	Baseline	+OE	Baseline	+OE
PTB Char	Answers	99.0	93.0	79.8	87.3	81.7	89.0
	Email	98.5	76.9	81.7	89.2	83.8	91.2
	Newsgroup	99.6	98.0	77.9	89.3	77.5	87.7
	Reviews	98.6	72.2	81.0	89.3	82.4	91.1
	Weblog	100.0	100.0	68.3	78.7	70.0	81.8
Mean		99.1	**88.0**	77.7	**86.8**	79.1	**88.2**
PTB Word	Answers	40.6	3.48	82.6	98.1	55.0	96.3
	Email	73.4	0.17	73.5	99.2	52.6	98.8
	Newsgroup	58.1	0.17	75.6	99.5	41.7	98.6
	Reviews	29.6	0.85	88.1	99.0	67.0	98.0
	Weblog	47.8	0.08	80.8	99.9	51.4	99.8
Mean		49.9	**0.95**	80.1	**99.1**	53.6	**98.3**

Table 6: Anomaly detection results on Penn Treebank examples and English Web Treebank outliers. All values are percentages.

We train word-level models for 300 epochs, and character-level models for 50 epochs [Merity et al., 2017, 2018]. We then fine-tune using OE on WikiText-2 for 5 epochs. For the character-level language model, we create a character-level version of WikiText-2 by converting words to lowercase, and leaving out characters that do not appear in PTB. Anomaly detection results for the word-level and character-level language models are shown in Table 6. In all cases, OE improves over the baseline, and the improvement is especially large for the word-level model.

4.4 Confidence Calibration

Models integrated into a decision making process should indicate when they are trustworthy and not have inordinate confidence in its prediction. In an effort to combat an false sense of certainty from overconfident models, we aim to calibrate the model confidence. A model is calibrated if confidence estimates represent a true correctness likelihood. Thus if a calibrated model predicts an event with 30% confidence, then 30% of the time the event transpires. Prior research [Guo et al., 2017, Nguyen and O'Connor, 2015, Kuleshov and Liang, 2015] studies calibrating systems where all inputs are in $\mathcal{D}_{\text{in}}$, but such systems should also ascribe low confidence to inputs from $\mathcal{D}_{\text{out}}^{\text{test}}$. This motivates us to apply OE for better calibration. Confidence calibration metrics are in the Supplementary Material.

4.4.1 Setup and Results

There are many ways to compute a classifier's confidence estimate. One way is bind a logistic regression branch onto the network, so that confidence values are in $[0, 1]$. Other confidence estimates use the model's logits, $l \in \mathbb{R}^k$. Another confidence estimate is $\max_i \big[\exp(l_i)/\sum_{j=1}^{k}\exp(l_j)\big]$ which has confidence values in $[1/k, 1]$, k the number of classes. A third possible confidence estimate is $\sigma(\max_i l_i) \in [0, 1]$.

Softmax Temperature Tuning. A confidence estimation technique shown to work consistently well requires simply adjusting the softmax temperature [Guo et al., 2017]. Specifically, they compute their confidence estimate with the formula $c_T := \max_i \big[\exp(l_i/T)/\sum_{j=1}^{k}\exp(l_j/T)\big]$, T a constant. To obtain T, they first hold out a validation set. Then, using convex optimization software, they find the T which minimizes the cross entropy (negative average log-likelihood).

0-1 Posterior Rescaling. While temperature tuning improves calibration, the confidence estimate c_T cannot be less than $1/k$. For an out-of-distribution example like Gaussian Noise, a good model should have no confidence in its prediction over k classes. One option is to add a possibility option, or a $(k + 1)$st class, which we cover in the Discussion section. A simpler option is to perform an affine transformation of $c_T \in [1/k, 1]$ with the formula $(c_T - 1/k)/(1 - 1/k) \in [0, 1]$. This simple transformation makes it possible for a network to express no confidence on an out-of-distribution input and improves calibration performance.

Results. In this calibration experiment, the baseline is simply confidence estimation with softmax temperature tuning. Therefore, we train MNIST, SVHN, CIFAR-10, CIFAR-100, and Tiny

	Mean RMS Calib. Error ↓			Mean MAV Calib. Error ↓			Mean Soft F1 Score ↑		
$\mathcal{D}_{\text{in}}$	Baseline	0-1 Rescale	+OE	Baseline	0-1 Rescale	+OE	Baseline	0-1 Rescale	+OE
MNIST	13.5	12.7	5.9	7.1	6.8	3.0	53.9	57.3	74.4
SVHN	16.0	14.8	2.7	6.7	6.2	1.0	50.8	54.4	88.1
CIFAR-10	20.3	18.6	6.2	13.2	12.1	3.8	41.1	44.4	73.6
CIFAR-100	14.1	13.8	10.2	12.0	11.3	7.9	58.8	59.3	66.5
Tiny ImageNet	8.4	8.3	4.2	6.6	6.5	3.0	65.5	65.7	73.0

Table 7: Calibration results for the softmax temperature tuning baseline, the same baseline after adding 0-1 Posterior Rescaling, and temperature tuning + 0-1 Rescaling + OE.

ImageNet classifiers with 6000, 5000, 5000, 5000, and 10000 training examples held out. A copy of this classifier is fine-tuned with Outlier Exposure, like in section 4.2.3. Then we determine the optimal temperature of the original and fine-tuned classifier on the held out examples. Calibration measures are in the Supplementary Material. To measure calibration, we take examples from the test distribution and equally many examples from a distribution $\mathcal{D}_{\text{out}}^{\text{test}}$. Out-of-distribution points are understood to be incorrectly classified since their label is not in the model's output space. Results are in Table 7. Full results are in the Supplementary Materials. Every model used temperature tuning. Notably, the simple 0-1 posterior rescaling technique consistently improves calibration, and the model fine-tuned with OE using temperature tuning and 0-1 posterior rescaling achieved large calibration improvemetns. On this problem Outlier Exposure markedly improves model calibration.

4.5 Discussion

Extensions to Multilabel Classifiers and the Reject Option. Outlier Exposure can work in more classification regimes than just those considered above. For example, a multilabel classifier trained on CIFAR-10 has obtains an 88.8% mean AUROC when the prediction probability is the anomaly score. By depressing the classifier's output probabilities with OE, the mean AUROC increases to 97.1%. An alternative formulation for anomaly detection is to include a "reject class" to a classifier [Bartlett and Wegkamp, 2008]. Outlier Exposure can also work in this setting, but classifiers with the reject option or multilabels are not as competitive in anomaly detection as multiclass classifiers with OE.

$\mathcal{D}_{\text{out}}^{\text{OE}}$ Diversity and Closeness to $\mathcal{D}_{\text{in}}$. The choice of outlier distribution $\mathcal{D}_{\text{out}}^{\text{OE}}$ in Outlier Exposure is central. If the outliers are only Gaussian noise samples, then the detector will hardly improve on novel anomalies. Clearly the outlier diversity matters. Furthermore, a CIFAR-10 classifier exposed to 10 CIFAR-100 outlier classes corresponds to an average AUPR of 78.5% (while excluding CIFAR-100 detection performance from the average). Exposed to 30 such classes, the classifier's average AUPR becomes 85.1%. 50 classes corresponds to 85.3%, and from thereon additional CIFAR-100 classes barely improve performance.

The closeness of $\mathcal{D}_{\text{out}}^{\text{OE}}$ to $\mathcal{D}_{\text{in}}$ is also worthy of analysis. In the supplementary materials, we observe that MNIST and SVHN performance improve with Outlier Exposure to 80 Million Tiny Images, even though the outliers are images of natural scenes not digits. In a separate experiment, we used Online Hard Example Mining so that difficult outliers have more weight in Outlier Exposure. Although this improves performance on the hardest anomalies, anomalies without plausible local statistics like noise are detected less effectively than before. Consequently, outliers need not be close-to-distribution for anomaly detection performance gains.

5 Conclusion

In this paper, we propose leveraging vast quantities of diverse data for improving anomaly detection with Outlier Exposure. In extensive experiments, we show that Outlier Exposure significantly improves the performance of a range of recently proposed baseline anomaly detectors, and can be applied with fine-tuning at low computational cost. We also show that Outlier Exposure improves the calibration of neural network classifiers when anomalous data is present. Our results suggest that Outlier Exposure is a promising approach for improving future anomaly detection systems, and can be applied with low overhead and substantial gains in performance to existing systems.

References

Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(Aug):1823–1840, 2008. 8

Abhijit Bendale and Terrance Boult. Towards open set deep networks. *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. English web treebank, 2012. 6

Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1431–1439, 2015. 2

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint*, 2017. 3

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *International Conference on Machine Learning (ICML)*, 2006. 3

Harm de Vries, Roland Memisevic, and Aaron Courville. Deep learning vector quantization. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016. 2

Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018. 1, 2, 5

Andrew F Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, pages 16–21. ACM, 2013. 1

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 2

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *International Conference on Machine Learning*, 2017. 7

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint*, 2018. 11

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. ICLR, 2017. 1, 2, 4, 12

Johnson et al. Tiny imagenet visual recognition challenge. URL `https://tiny-imagenet.herokuapp.com`. 3

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 3

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NIPS 2012*, 2012. 1

Volodymyr Kuleshov and Percy Liang. Calibrated structured prediction. *NIPS*, 2015. 7

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *International Conference on Learning Representations*, 2018. 2, 5, 6

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *International Conference on Learning Representations*, 2018. 1, 2, 5

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. *arXiv preprint*, 2018. 2

Chris Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. 3

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. 2

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and Optimizing LSTM Language Models. *arXiv preprint arXiv:1708.02182*, 2017. 7

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. An Analysis of Neural Language Modeling at Multiple Scales. *arXiv preprint arXiv:1803.08240*, 2018. 7

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. 3

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

Khanh Nguyen and Brendan O'Connor. Posterior calibration and exploratory analysis for natural language processing models. *arXiv preprint arXiv:1508.05154*, 2015. 7

Joan Pastor-Pellicer, Francisco Zamora-Martínez, Salvador España-Boquera, and María José Castro-Bleda. F-measure as the error function to train neural networks. In *International Work-Conference on Artificial Neural Networks (IWANN)*, 2013. 12

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016. 2

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017. 2

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 3

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 6

Akshayvarun Subramanya, Suraj Srinivas, and R.Venkatesh Babu. Confidence estimation in deep neural networks via density modelling. *arXiv preprint*, 2017. 2

Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30 (11):1958–1970, 2008. 2

Antonio Torralba, Joshua B Tenenbaum, and Ruslan R Salakhutdinov. Learning to learn with compound hd models. In *Advances in Neural Information Processing Systems*, pages 2061–2069, 2011. 2

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. 3

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. 11

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2

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. 3

A MNIST and SVHN Multiclass Results

Bernoulli images have each pixel sampled from a Bernoulli distribution if the input range is $[0, 1]$. *Icons-50* is a dataset of icons; icons from the "Number" class are removed Hendrycks and Dietterich [2018]. *Fashion-MNIST* is a drop-in replacement for MNIST, where the images and classes correspond to apparel rather than digits. *Negative MNIST* is a dataset obtained by taking $(1 - x)$ for each data point x in the MNIST dataset. *notMNIST* is a dataset of black and white images of the letters A through J rendered in a variety of stylized fonts. *Omniglot* is a dataset of black and white images of handwritten characters from multiple languages, and character resembling digits are removed. *Chars74K* is a dataset of photographed characters in various styles; letters like like "O" and "l" were removed since they can look like numbers.

		FPR99 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	Baseline	+OE	Baseline	+OE	Baseline	+OE
MNIST	Gaussian	1.0	0.0	99.9	100.0	99.4	100.0
	Bernoulli	3.3	0.0	99.6	100.0	97.0	100.0
	CIFAR-10	7.7	0.0	99.5	100.0	97.4	100.0
	Icons-50	6.4	0.0	99.5	100.0	97.7	99.9
	Fashion-MNIST	12.1	4.7	98.2	99.9	91.4	99.3
	Negative MNIST	17.8	0.4	98.0	100.0	90.4	99.9
	notMNIST	32.4	12.8	96.2	98.8	76.8	92.6
	Omniglot	45.2	48.5	95.8	95.5	77.8	77.6
	Mean	15.7	**8.3**	98.3	**99.3**	91.0	**96.2**
SVHN	Gaussian	48.3	0.0	96.6	100.0	75.2	99.3
	Bernoulli	34.9	0.0	97.1	100.0	78.1	99.2
	Blobs	11.6	0.0	98.4	100.0	85.1	100.0
	Icons-50	66.6	0.2	96.8	100.0	87.7	99.9
	Textures	53.8	0.5	97.0	100.0	83.3	99.6
	Places365	42.7	0.1	97.5	100.0	83.5	99.9
	LSUN	48.4	0.1	97.2	100.0	81.1	99.9
	CIFAR-10	53.1	0.1	97.3	100.0	89.3	100.0
	Chars74K	80.5	17.6	95.1	98.7	71.5	90.1
	Mean	48.9	**2.1**	97.0	**99.8**	81.6	**98.6**

Table 8: Anomaly detection results for the maximum softmax probability (MSP) baseline, and MSP+OE. All values are percentages.

	Mean FPR99 ↓		Mean AUROC ↑		Mean AUPR ↑	
$\mathcal{D}_{\text{in}}$	$\max_c p_c$	KL$[U \| p]$	$\max_c p_c$	KL$[U \| p]$	$\max_c p_c$	KL$[U \| p]$
MNIST	8.3	7.0	99.3	99.4	96.2	97.0
SVHN	2.1	1.6	99.8	99.9	98.6	98.8

Table 9: Comparison between the maximum softmax probability and KL$[U \| p]$ anomaly scoring techniques. Both scoring techniques are for a model fine-tuned with OE. All values are percentages and averaged across all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets.

B Architecture and Training Details

For CIFAR-10 and CIFAR-100 classification experiments, we use a 40-4 Wide Residual Network [Zagoruyko and Komodakis, 2016]. The network trains for 100 epochs with a dropout rate of 0.3. The initial learning rate of 0.1 decays following a cosine learning rate schedule. We use standard flipping and data cropping augmentation. Tiny ImageNet training is similar, except that we use a 40-2 Wide ResNet. SVHN architectures are 16-4 Wide ResNets trained for 20 epochs with an initial learning rate of 0.01 and no data augmentation. OE fine-tuning occurs for 10 epochs (of the length of the in-distribution dataset) on all classifiers except SVHN. SVHN is fine-tuned with OE for 5 epochs.

C Calibration Evaluation Metrics

In order to evaluate a multiclass classifier's calibration, we present three metrics. First we establish context. Let $Y \in \{1, 2, \ldots, k\}$ be the ground truth class. Let $\widehat{Y}$ be the model's class prediction, and let C be the corresponding model confidence or prediction probability. Denote the set of prediction-label pairs made by the model with $S = \{(\widehat{y}_1, c_1), (\widehat{y}_2, c_2), \ldots, (\widehat{y}_n, c_n)\}$.

RMS and MAV Calibration Error. The Root Mean Square Calibration Error measures the square root of the expected squared difference between confidence and accuracy at a confidence level. It has the formula $\sqrt{\mathbb{E}_C[\mathbb{P}(Y = \widehat{Y}|C = c) - c]^2]}$. A similar formulation which penalizes large confidence-accuracy deviations less severely is the Mean Absolute Value Calibration error, written $\mathbb{E}_C[|\mathbb{P}(Y = \widehat{Y}|C = c) - c|]$. The MAV Calibration Error is a lower bound of the RMS Calibration Error. To empirically estimate these miscalibration measures, we partition the n samples of S into m bins $\{B_1, \ldots, B_m\}$ with approximately 100 samples in each bin. Unlike [Weinberger], bins are not equally spaced since the distribution of confidence values is not uniform but dynamic. Concretely, the RMS Calibration Error is estimated with

$$\sqrt{\sum_{i=1}^{m} \frac{|B_i|}{n} \left(\frac{1}{|B_i|} \sum_{k \in B_i} \mathbb{1}(y_k = \widehat{y}_k) - \frac{1}{|B_i|} \sum_{k \in B_i} c_k \right)^2}.$$

Along similar lines, the MAV Calibration Error is estimated with

$$\sum_{i=1}^{m} \frac{|B_i|}{n} \left| \frac{1}{|B_i|} \sum_{k \in B_i} \mathbb{1}(y_k = \widehat{y}_k) - \frac{1}{|B_i|} \sum_{k \in B_i} c_k \right|.$$

Soft F1 Score. If a classifier produces few mistakes, then most examples should have high confidence. But if the classifier gives all predictions high confidence, including its mistakes, then the previous metrics will indicate that the model is calibrated on the vast majority of instances, despite having systematic miscalibration. The Soft F1 score [Pastor-Pellicer et al., 2013, Hendrycks and Gimpel, 2017] is suited for measuring the calibration of a system where there is an acute imbalance between mistakes and correct decisions. Its first component is the true positive rate, written $\text{tp} = \sum_{i=1}^{n}(1 - c_i)\mathbb{1}(y_i \neq \widehat{y}_i)$. The false positive rate is $\text{fp} = \sum_{i=1}^{n}(1 - c_i)\mathbb{1}(y_i = \widehat{y}_i)$, and the false negative rate is $\text{fn} = \sum_{i=1}^{n} c_i \mathbb{1}(y_i \neq \widehat{y}_i)$. The precision is defined as $\text{pr} = \text{tp}/(\text{tp} + \text{fp})$, and the recall is $\text{rc} = \text{tp}/(\text{tp} + \text{fn})$. These formulas define the Soft F1 score which is

$$2\text{pr} \cdot \text{rc}/(\text{pr} + \text{rc}).$$