DEEP ANOMALY DETECTION WITH OUTLIER EXPOSURE

Anonymous authors
Paper under double-blind review

ABSTRACT

It is important to detect and handle anomalous inputs when deploying machine learning systems. The use of larger and more complex inputs in deep learning magnifies the difficulty of distinguishing between anomalous and in-distribution examples. At the same time, diverse image and text data commonly used by deep learning systems are available in enormous quantities. We propose leveraging these data to improve deep anomaly detection by training anomaly detectors against an auxiliary dataset of outliers, an approach we call Outlier Exposure (OE). In extensive experiments in vision and natural language processing settings, we find that Outlier Exposure significantly improves the performance of existing anomaly detectors, including detectors based on density estimation, and that OE improves classifier calibration in the presence of anomalous inputs. We also analyze the flexibility and robustness of Outlier Exposure, and identify characteristics of the auxiliary dataset that improve performance.

1 INTRODUCTION

Machine Learning systems in deployment often encounter data that is unlike the model's training data. This can occur in discovering novel astronomical phenomena, finding unknown diseases, or detecting sensor failure. In these situations, models that can detect anomalies (Liu et al., 2018; Emmott et al., 2013) are capable of correctly flagging unusual examples for human intervention, or carefully proceeding with a more conservative fallback policy.

Behind many machine learning systems are deep learning models (Krizhevsky et al., 2012) which can provide high performance in a variety of applications, so long as the data seen at test time is similar to the training data. However, when there is a distribution mismatch, deep neural network classifiers tend to give high confidence predictions on anomalous test examples (Nguyen et al., 2015). This can invalidate the use of prediction probabilities as calibrated confidence estimates (Guo et al., 2017), and makes detecting anomalous examples 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* (OOD) examples (Hendrycks & Gimpel, 2017; DeVries & Taylor, 2018; Liu et al., 2018). These approaches have been demonstrated to work surprisingly well for complex input spaces, such as images, text, and speech. Moreover, they do not require modeling the full data distribution, but instead can use heuristics for detecting OOD, unmodeled phenomena. Several of these methods detect unmodeled phenomena by using representations from only in-distribution data.

In this paper, we investigate a complementary method where we train models to detect unmodeled data by learning cues for whether an input is unmodeled. While it is difficult to model the full data distribution, we can learn effective heuristics for detecting out-of-distribution inputs by *exposing* the model to OOD examples, thus learning a more conservative concept of the inliers. We propose leveraging diverse, realistic datasets for this purpose, with a method we call Outlier Exposure (OE). Akin to fine-tuning a classifier with pretrained features, OE provides a simple and effective way to consistently improve existing methods for OOD detection.

Through numerous experiments, we extensively evaluate the broad applicability of Outlier Exposure. For multiclass neural networks, we provide thorough results on Computer Vision and Natural Language Processing tasks which show that Outlier Exposure can help anomaly detectors generalize to and perform well on unseen distributions of outliers. Also, we demonstrate that Outlier Exposure

provides gains over several existing approaches to out-of-distribution detection. Our results also show the flexibility of Outlier Exposure, as we can train models with different sources of outlier distributions. Additionally, we establish that Outlier Exposure can make density estimates of OOD samples significantly more useful for OOD detection. Finally, we demonstrate that Outlier Exposure improves the calibration of neural network classifiers in the realistic setting where a fraction of the data is OOD. Our code is made publicly available at [anonymized].

2 Related Work

Out-of-Distribution Detection with Deep Networks. Hendrycks & 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 conveniently double as a consistently useful out-of-distribution detector. Building on this work, DeVries & Taylor (2018) attach an auxiliary branch onto a pre-trained classifier and derive a new OOD score from this branch. Liang et al. (2018) present a method which can improve performance of OOD detectors that use a softmax distribution. In particular, they make the maximum softmax probability more discriminative between anomalies and in-distribution examples by pre-processing input data with adversarial perturbations (Goodfellow et al., 2015). Unlike in our work, their hyperparameters are tailored to each source of anomalies.

Lee et al. (2018) train a classifier concurrently with a GAN (Radford et al., 2016), and the classifier is trained to have lower confidence on GAN samples. For each testing distribution of anomalies, they tune the classifier and GAN using samples from that out-distribution, as discussed in Appendix B of their work. Unlike Liang et al. (2018); Lee et al. (2018), in this work we train our method *without* tuning parameters to fit specific types of anomaly test distributions, so our results are not directly comparable with their results. Many other works (de Vries et al., 2016; Bendale & Boult, 2016; Subramanya et al., 2017) also encourage the model to have lower confidence on anomalous examples. Recently, Liu et al. (2018) provide theoretical guarantees for detecting out-of-distribution examples.

Utilizing Auxiliary Datasets. Outlier Exposure uses an auxiliary dataset to give the network better representations in order to detect anomalies. Goodfellow et al. (2015) train on adversarial examples to increased robustness. Torralba et al. (2011) pre-train unsupervised deep models on a database of web images for stronger features. Radford et al. (2017) train an unsupervised network on a corpus of Amazon reviews for a month in order to obtain quality sentiment representations. Zeiler & Fergus (2014) find that pre-training a network on the large ImageNet database (Russakovsky et al., 2015) endows the network with general representations that are useful in many fine-tuning applications. Chen & 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 task of deciding whether or not a sample is from a learned distribution called $\mathcal{D}_{\text{in}}$. Samples from $\mathcal{D}_{\text{in}}$ are called "in-distribution," and otherwise are said to be "out-of-distribution" (OOD) or samples from $\mathcal{D}_{\text{out}}$. In real applications, it may be difficult to know the distribution of outliers one will encounter in advance. Thus, we consider the realistic setting where $\mathcal{D}_{\text{out}}$ is unknown. Given a parametrized OOD detector and an Outlier Exposure (OE) dataset $\mathcal{D}_{\text{out}}^{\text{OE}}$, we train the model to discover signals and learn heuristics to detect whether a query is sampled from $\mathcal{D}_{\text{in}}$ or $\mathcal{D}_{\text{out}}^{\text{OE}}$. We find that these heuristics generalize to unseen distributions $\mathcal{D}_{\text{out}}$.

Deep parametrized anomaly detectors typically leverage learned representations from an auxiliary task, such as classification or density estimation. Given a representation learner f, we can thus formalize Outlier Exposure as minimizing the objective

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{in}}}[-\mathcal{L}(f(x),y) + \lambda\mathbb{E}_{x'\sim\mathcal{D}_{\text{out}}^{\text{OE}}}[\mathcal{L}_{\text{OE}}(f(x'),f(x),y)]]$$

over the parameters of f. In cases where labeled data is not required or available, then y can be ignored. The primary loss function $\mathcal{L}$ is the original representation learning objective, and $\mathcal{L}_{\text{OE}}$ is a loss incorporating OE.

Outlier Exposure can be applied with many types of data and original tasks. Hence, the specific formulation of $\mathcal{L}_{\text{OE}}$ is a design choice, and depends on the task at hand and the OOD detector used. For example, when using the maximum softmax probability baseline detector (Hendrycks & Gimpel, 2017), we set $\mathcal{L}_{\text{OE}}$ to the cross-entropy between $f(x')$ and the uniform distribution. When the original objective $\mathcal{L}$ is density estimation and labels are not available, we set $\mathcal{L}_{\text{OE}}$ to a margin ranking loss on the log probabilities $f(x')$ and $f(x)$. When performing Outlier Exposure, we fine-tune a pre-trained f by sampling equally many examples from $\mathcal{D}_{\text{in}}$ and $\mathcal{D}_{\text{out}}^{\text{OE}}$ per batch. The entire Outlier Exposure process usually requires only a few minutes, even for large datasets.

4 EXPERIMENTS

4.1 EVALUATING OUT-OF-DISTRIBUTION DETECTION METHODS

To evaluate different out-of-distribution detection methods, we use three metrics: area under the receiver operating characteristic curve (*AUROC*), area under the precision-recall curve (*AUPR*), and the false positive rate at $N\%$ true positive rate (*FPR*N). Since we aim to detect anomalous examples, we define anomalous examples as the positive class in these measures. The AUROC and AUPR are holistic metrics, since they summarize the performance of a detection method across several different OOD score thresholds. The AUROC can be thought of as the probability that an anomalous example is given a higher OOD score than a in-distribution example (Davis & 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 & Schütze, 1999), as it takes the base rate of anomalies into account. During evaluation with these metrics, the base rate of $\mathcal{D}_{\text{out}}^{\text{test}}$ to $\mathcal{D}_{\text{in}}^{\text{test}}$ test examples in all our experiments is 1:5.

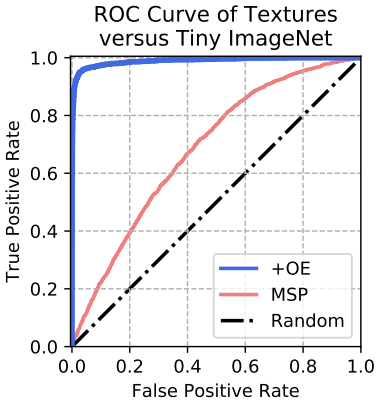

Figure 1: ROC curve with Tiny ImageNet ($\mathcal{D}_{\text{in}}$) and Textures ($\mathcal{D}_{\text{out}}^{\text{test}}$).

Whereas the previous two metrics represent the detection performance across various thresholds, the FPRN metric represents performance at one strict threshold. By observing performance at a strict threshold, we can make clear comparisons among strong detectors. The FPRN metric (Liu et al., 2018; Kumar et al., 2016; Balntas et al., 2016) is the probability that an in-distribution example (negative) raises a false alarm when $N\%$ of anomalous examples (positive) are detected, so a lower FPRN is better. When anomalies require human intervention, capturing nearly all anomalies with few false alarms is of high practical value.

4.2 MULTICLASS CLASSIFICATION

We evaluate OOD detectors on a wide range of datasets with multiclass classification as the original task. Each evaluation consists of an in-distribution dataset $\mathcal{D}_{\text{in}}$ used to train a multiclass classifier, a dataset of anomalous examples $\mathcal{D}_{\text{out}}^{\text{OE}}$, and a baseline detector to which we apply OE.

4.2.1 IN-DISTRIBUTION DATASETS

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 & 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 disjoint but have similiarities. For example, CIFAR-10 has "automobiles" and "trucks" but not CIFAR-100's "pickup truck" class. Both have $50,000$ training images and $10,000$ test images. Pixels are standardized by the average training image's channel means and standard deviations.

Tiny ImageNet. The Tiny ImageNet dataset (Johnson et al.) is a 200-class subset of the ImageNet (Russakovsky et al., 2015) dataset where images are 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 the test set has $10,000$ images. Pixels are standardized by the average training image's channel means and standard deviations.

20 Newsgroups. 20 Newsgroups is a text classification dataset of newsgroup documents with 20 classes, and approximately $20,000$ examples split evenly among the classes. We use the standard 60/40 train/test split.

TREC. TREC is a question classification dataset with 50 fine-grained classes, and $5,952$ individual questions. We reserve 500 examples for the test set, and use the rest for training.

SST. The Stanford Sentiment Treebank dataset (Socher et al., 2013) consists of movie reviews categorized by their positive or negative sentiment. The dataset has $8,544$ reviews for training and $2,210$ for testing.

4.2.2 Outlier Exposure Datasets

80 Million Tiny Images. 80 Million Tiny Images (Torralba et al., 2008) is a large-scale, diverse dataset of 32×32 natural images scrapped form the web. We use this dataset as $\mathcal{D}_{\text{out}}^{\text{OE}}$ for experiments with SVHN, CIFAR-10, and CIFAR-100 as $\mathcal{D}_{\text{in}}$. We remove all examples of 80 Million Tiny Images which appear in the CIFAR datasets. In the Discussion we note that only a small fraction of this dataset is necessary for successful OE.

ImageNet-22K. We use the ImageNet dataset with images from approximately 22 thousand classes as $\mathcal{D}_{\text{OE}}$ for experiments with Tiny ImageNet as $\mathcal{D}_{\text{in}}$; images in ImageNet-1K are removed.

WikiText-2. WikiText-2 is a corpus of Wikipedia articles typically used for language modeling. We use WikiText-2 as $\mathcal{D}_{\text{out}}^{\text{OE}}$ for language modeling experiments with Penn Treebank as $\mathcal{D}_{\text{in}}$. For classification tasks on 20 Newsgroups, TREC, and SST, we treat each sentence of WikiText-2 as an individual example, and use simple filters to remove low-quality sentences. Code for reproducing this dataset from the raw WikiText-2 corpus is provided with the main experiment code.

In what follows, we use Outlier Exposure to enhance the performance of existing OOD detection techniques. Throughout the following experiments, we let $x \in \mathcal{X}$ be a classifier's input and $y \in \mathcal{Y} = \{1, 2, \ldots, k\}$ be a class. We also represent the classifier with the function $p : \mathcal{X} \to \mathbb{R}^k$, such that for any x, $1^\top p(x) = 1$ and $p(x) \succeq 0$ so that p lies on a probability simplex.

Maximum Softmax Probability. Consider the maximum softmax probability baseline (Hendrycks & Gimpel, 2017) which gives an input x the OOD score $-\max_c p_c(x)$. Out-of-distribution samples are drawn from various unseen distributions (Appendix A). For each task, we test with approximately twice the number of $\mathcal{D}_{\text{out}}^{\text{test}}$ distributions compared to most other papers, and we also test on NLP tasks. The quality of the OOD example scores are judged with the metrics described in Section 4.1. For this multiclass setting, we can perform Outlier Exposure by fine-tuning a pre-trained classifier p so that it has a uniform posterior on $\mathcal{D}_{\text{out}}^{\text{OE}}$ samples. Formally, the fine-tuning objective is $\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{in}}}[-\log p_y(x)] + \lambda \mathbb{E}_{x\sim\mathcal{D}_{\text{out}}^{\text{OE}}}[H(\mathcal{U}; p(x))]$, where H is the cross entropy and $\mathcal{U}$ is the uniform distribution over k classes. When there is class imbalance, we could encourage $p(x)$ to match $(P(y=1), \ldots, P(y=k))$; yet for the datasets we consider, matching $\mathcal{U}$ works well enough. Also, note that training from scratch with OE can result in even better performance than fine-tuning (Appendix F). This approach works on different architectures as well (see Appendix G).

	FPR95 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	MSP	+OE	MSP	+OE	MSP	+OE
SVHN	6.3	0.1	98.0	100.0	91.1	99.9
CIFAR-10	34.9	9.5	89.3	97.8	59.2	90.5
CIFAR-100	62.7	38.5	73.1	87.9	30.1	58.2
Tiny ImageNet	66.3	14.0	64.9	92.2	27.2	79.3

Table 1: Out-of-distribution image detection for the maximum softmax probability (MSP) baseline detector and the MSP detector after fine-tuning with Outlier Exposure (OE). Results are percentages and also an average of 10 runs.

Unlike Liang et al. (2018); Lee et al. (2018) and like Hendrycks & Gimpel (2017); DeVries & Taylor (2018), we do not tune our hyperparameters for each $\mathcal{D}_{\text{out}}^{\text{test}}$ distribution, so that $\mathcal{D}_{\text{out}}^{\text{test}}$ is kept unknown like with real-world anomalies. Instead, the λ coefficients were determined early in experimentation with validation $\mathcal{D}_{\text{out}}$ distributions that are not included in the results. For vision

	FPR90 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	MSP	+OE	MSP	+OE	MSP	+OE
20 Newsgroups	42.4	4.9	82.7	97.7	49.9	91.9
TREC	43.5	0.8	82.1	99.3	52.2	97.6
SST	74.9	27.3	61.6	89.3	22.9	59.4

Table 2: Comparisons between the MSP baseline and the MSP of the natural language classifier fine-tuned with OE. Results are percentages and averaged over 10 runs.

experiments we use $\lambda = 0.5$, and for NLP $\lambda = 1.0$. Like previous OOD detection methods involving network fine-tuning, we chose λ so that impact on classification accuracy is negligible.

For vision experiments, we train Wide Residual Networks (Zagoruyko & Komodakis, 2016) and then fine-tune network copies with OE for 10 epochs. For NLP experiments, we train 2-layer GRUs (Cho et al., 2014) for 5 epochs, then fine-tune network copies with OE for 2 epochs. During each epoch of OE, we use the full training set, and equally many outliers sampled from $\mathcal{D}_{\text{out}}^{\text{OE}}$. This way OE requires only a few minutes on a single GPU. Networks trained on CIFAR-10 or CIFAR-100 are exposed to outliers from 80 Million Tiny Images, and the Tiny ImageNet classifier is exposed to ImageNet-22K. NLP classifiers are exposed to outliers from WikiText-2. Further architectural and training details are in Appendix B. For all tasks, OE improves average performance by a large margin. Averaged results are shown in Tables 1 and 2. Sample ROC curves are shown in Figures 1 and 3. Detailed results on individual $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets are in Table 8 and Table 9 in Appendix A. Notice that the SVHN classifier with OE can be used to detect new anomalies such as emojis and street view alphabet letters, even though $\mathcal{D}_{\text{OE}}^{\text{test}}$ is a dataset of natural images. Thus, Outlier Exposure allows models to generalize to unseen $\mathcal{D}_{\text{out}}^{\text{test}}$ distributions far better than the baseline.

Confidence Branch. A recently proposed OOD detection technique (DeVries & Taylor, 2018) involves appending an OOD scoring branch $b : \mathcal{X} \to [0, 1]$ onto a deep network. Trained with samples from only $\mathcal{D}_{\text{in}}$, this branch estimates the network's confidence on any input. The creators of this technique made their code publicly available, so we use their code to train new 40-4 Wide Residual Network classifiers. We fine-tune the confidence branch with Outlier Exposure by simply adding $0.5\mathbb{E}_{x \sim \mathcal{D}_{\text{out}}^{\text{OE}}}[\log b(x)]$ 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 & Taylor (2018)'s publicly available training code. The confidence branch improves over this MSP detector, and after OE the confidence branch detects anomalies more effectively.

	FPR95 ↓			AUROC ↑			AUPR ↑		
$\mathcal{D}_{\text{in}}$	MSP	Branch	+OE	MSP	Branch	+OE	MSP	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 among the maximum softmax probability, Confidence Branch, and Confidence Branch + OE OOD detectors. The same network architecture is used for all three detectors. All results are percentages, and averaged across all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets.

Synthetic Outliers. Outlier Exposure leverages the simplicity of downloading real datasets, but it is possible to generate synthetic outliers. Note that we made an attempt to distort images with noise and use these as outliers for OE, but the classifier quickly memorized this statistical pattern and did not detect new OOD examples any better than before (Hafner et al., 2018). A method with better success is from Lee et al. (2018). They carefully train a GAN to generate synthetic examples near the classifier's decision boundary. The classifier is encouraged to have a low maximum softmax probability on these synthetic examples. For CIFAR classifiers, they mention that a GAN can be a better source of anomalies than datasets such as SVHN. In contrast, we find that the simpler approach of drawing anomalies from a diverse dataset is sufficient for marked improvements in OOD detection.

We train a 40-4 Wide Residual Network using Lee et al. (2018)'s publicly available code, and use the network's maximum softmax probabilities as our baseline. Another classifier trains concurrently

with a GAN so that the classifier assigns GAN-generated examples a high OOD score. We want each $\mathcal{D}_{\text{out}}^{\text{test}}$ to be novel. Consequently we use their code's default hyperparameters, and exactly one model encounters all tested $\mathcal{D}_{\text{out}}^{\text{test}}$ distributions. This is unlike their work since, for each $\mathcal{D}_{\text{out}}^{\text{test}}$ distribution, they train and tune a new network. We do not evaluate on Tiny ImageNet, since DCGANs cannot stably generate images of that scale and diversity. Lastly, we take the network trained in tandem with a GAN and fine-tune it with OE. Table 4 shows the large gains from using OE with a real and diverse dataset over using synthetic samples from a GAN.

	FPR95 ↓			AUROC ↑			AUPR ↑		
$\mathcal{D}_{\text{in}}$	MSP	+GAN	+OE	MSP	+GAN	+OE	MSP	+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 among the maximum softmax probability (MSP), MSP + GAN, and MSP + GAN + OE OOD detectors. The same network architecture is used for all three detectors. All results are percentages and averaged across all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets.

4.3 Density Estimation

		FPR95 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	BPP	+OE	BPP	+OE	BPP	+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: OOD detection results with a PixelCNN++ density estimator, and the same estimator after applying OE. The model's bits per pixel (BPP) scores each sample. All results are percentages. Test distributions $\mathcal{D}_{\text{out}}^{\text{test}}$ are described in Appendix A.

Density estimators learn a probability density function over the data distribution $\mathcal{D}_{\text{in}}$. Anomalous examples should have low probability density, as they are scarce in $\mathcal{D}_{\text{in}}$ by definition. Consequently, density estimates are another means by which to score anomalies (Zong et al., 2018). We show the ability of OE to improve density estimates on low-probability, outlying data.

PixelCNN++. Autoregressive neural density estimators provide 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 OOD detection. We use PixelCNN++ (Salimans et al., 2017) as a baseline OOD detector, and we train it on CIFAR-10. The bits-per-pixel (BPP) is used as an OOD score. This is then fine-tuned for 1 epoch using OE. Here 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 bit per pixel. Results are shown in Table 5. For all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets, OE significantly improves results.

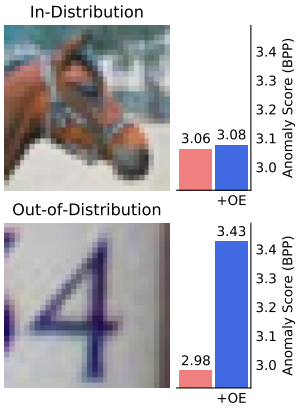

Figure 2: OOD scores from PixelCNN++ on images from CIFAR-10 and SVHN.

Language Modeling. We next explore using OE on language models. We use QRNN (Merity et al., 2018a;b) language models as baseline OOD detectors. The bits per character (BPC) or bits per word (BPW) is used as an OOD score. Outlier Exposure is implemented by adding the cross entropy to the uniform distribution on tokens from sequences in $\mathcal{D}_{\text{out}}^{\text{OE}}$ as an additional loss term.

$\mathcal{D}_{\text{in}}$	FPR90 ↓		AUROC ↑		AUPR ↑	
	BPC/BPW	+OE	BPC/BPW	+OE	BPC/BPW	+OE
PTB Char	99.0	89.4	77.5	86.3	76.0	86.7
PTB Word	48.5	0.98	81.2	99.2	44.0	97.8

Table 6: OOD detection results on Penn Treebank language models. Results are percentages averaged over the $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets. Expanded results are in Appendix I.

For $\mathcal{D}_{\text{in}}$, we convert use the language-modeling version of Penn Treebank, split into sequences of length 70 for backpropagation for word-level models, and 150 for character-level models. We do not train or evaluate with preserved hidden states as in BPTT. This is because retaining hidden states would greatly simplify the task of OOD detection. Accordingly, the OOD detection task is to provide a score for 70- or 150-token sequences in the unseen $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets.

We train word-level models for 300 epochs, and character-level models for 50 epochs. 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 which do not appear in PTB. OOD detection results for the word-level and character-level language models are shown in Table 6; expanded results and $\mathcal{D}_{\text{out}}^{\text{test}}$ descriptions are in Appendix I. 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 such models should not have inordinate confidence in their predictions. In an effort to combat a false sense of certainty from overconfident models, we aim to calibrate 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 & O'Connor, 2015; Kuleshov & Liang, 2015) considers calibrating systems where test-time queries are samples from $\mathcal{D}_{\text{in}}$, but systems also encounter samples from $\mathcal{D}_{\text{out}}^{\text{test}}$ and should also ascribe low confidence to these samples. Hence, we use OE to control the confidence on these samples.

$\mathcal{D}_{\text{in}}$	RMS Calib. Error ↓		MAD Calib. Error ↓		Soft F1 Score ↑	
	Temperature	+OE	Temperature	+OE	Temperature	+OE
SVHN	22.3	4.8	10.9	2.4	52.1	87.9
CIFAR-10	19.4	9.3	12.6	5.6	39.9	69.7
CIFAR-100	14.7	8.2	11.3	6.5	52.8	65.8
Tiny ImageNet	12.0	6.9	9.0	4.8	62.9	72.2

Table 7: Calibration results for the temperature tuned baseline and temperature tuning + OE. Calibration metric descriptions are in Appendix C.

4.4.1 Setup and Results

There are many ways to estimate a classifier's confidence. One way is to bind a logistic regression branch onto the network, so that confidence values are in $[0, 1]$. Other confidence estimates use the model's logits $l \in \mathbb{R}^k$, such as the estimate $\sigma(\max_i l_i) \in [0, 1]$. Another common confidence estimate is $\max_i \left[\exp(l_i)/\sum_{j=1}^k \exp(l_j)\right]$. A modification of this estimate is our baseline.

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

Results. In this calibration experiment, the baseline is confidence estimation with softmax temperature tuning. Therefore, we train SVHN, CIFAR-10, CIFAR-100, and Tiny ImageNet classifiers with 5000, 5000, 5000, and 10000 training examples held out, respectively. A copy of this classifier

is fine-tuned with Outlier Exposure. Then we determine the optimal temperatures of the original and fine-tuned classifiers on the held-out examples. To measure calibration, we take equally many examples from a given in-distribution dataset $\mathcal{D}_{\text{in}}^{\text{test}}$ and OOD dataset $\mathcal{D}_{\text{out}}^{\text{test}}$. Confidence calibration metrics are in Appendix C. Out-of-distribution points are understood to be incorrectly classified since their label is not in the model's output space, so calibrated models should assign these out-of-distribution points low confidence. See Appendix E for an approach to allow models to assign no confidence to OOD samples. Results are in Table 7. Here Outlier Exposure noticeably improves model calibration.

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 multi*label* classifier trained on CIFAR-10 obtains an 88.8% mean AUROC when using the MSP detector. By using OE to control the classifier's output probabilities, the mean AUROC increases to 97.1%. An alternative OOD detection formulation is to give classifiers a "reject class" (Bartlett & Wegkamp, 2008). Outlier Exposure can also improve performance in this setting, but even with OE classifiers with the reject option or multilabel outputs are not as competitive in OOD detection as multiclass classifiers.

Flexibility in Choosing $\mathcal{D}_{\text{out}}^{\text{OE}}$. Size is not necessarily crucial in collecting a $\mathcal{D}_{\text{out}}^{\text{OE}}$ dataset. Concretely, 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%. Next, 50 classes corresponds to 85.3%, and from thereon additional CIFAR-100 classes barely improve performance. Moreover, experiments in this paper often used around 1% of the images in the 80 Million Tiny Images dataset since we only briefly fine-tuned the models, but we also found that using only 50,000 examples from this dataset had a negligible impact on performance. Sun et al. (2017) show a logarithmic relation between performance and dataset size. Additionally, $\mathcal{D}_{\text{out}}^{\text{OE}}$ datasets with significantly different statistics can perform similarly. For instance, using the Project Gutenberg dataset in lieu of WikiText-2 for $\mathcal{D}_{\text{out}}^{\text{OE}}$ in the SST experiments gives an average AUROC of 90.1% instead of 89.3%.

Closeness of $\mathcal{D}_{\text{out}}^{\text{test}}$, $\mathcal{D}_{\text{out}}^{\text{OE}}$, and $\mathcal{D}_{\text{in}}^{\text{test}}$. Our experiments show several interesting effects of the closeness of the datasets involved. Firstly, we find that $\mathcal{D}_{\text{out}}^{\text{test}}$ and $\mathcal{D}_{\text{out}}^{\text{OE}}$ need not be close for training with OE to improve performance on $\mathcal{D}_{\text{out}}^{\text{test}}$. In Appendix A, we observe that an OOD detector for SVHN has its performance improve with Outlier Exposure even though (1) $\mathcal{D}_{\text{out}}^{\text{OE}}$ samples are images of natural scenes not digits, and (2) $\mathcal{D}_{\text{out}}^{\text{test}}$ includes unnatural examples such as emojis. We observed the same in our preliminary experiments with MNIST; using 80 Million Tiny Images as $\mathcal{D}_{\text{out}}^{\text{OE}}$, OE increased the AUPR from 94.2% to 97.0%.

Secondly, we find that the closeness of $\mathcal{D}_{\text{out}}^{\text{OE}}$ to $\mathcal{D}_{\text{in}}^{\text{test}}$ can be an important factor in the success of OE. In the NLP experiments, preprocessing $\mathcal{D}_{\text{out}}^{\text{OE}}$ to be closer to $\mathcal{D}_{\text{in}}$ improves OOD detection performance significantly. Without preprocessing, the OE training objective can be satisfied in unintended ways, which can result in weaker detectors. In a separate experiment, we use 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 slightly less effectively than before. Real-world applications of OE could use the method of Sun et al. (2018) to refine a scraped $\mathcal{D}_{\text{out}}^{\text{OE}}$ auxiliary dataset to be appropriately close to $\mathcal{D}_{\text{in}}^{\text{test}}$.

6 Conclusion

In this paper, we proposed Outlier Exposure, a simple technique that enhances many current OOD detectors across various settings. It uses out-of-distribution samples to teach a network heuristics to detect new, unmodeled, out-of-distribution examples. We showed that this method is broadly applicable in vision and natural language settings and can improve model calibration and various anomaly detection techniques such as confidence branches, density estimation, and more. Such broad applicability suggests that OE may enable future methods for out-of-distribution detection. Moreover, the high performance of OE on standard datasets suggests the need to move to more challenging datasets in future works. Finally, Outlier Exposure is computationally efficient, and can be applied with low overhead to existing systems. In summary, Outlier Exposure is an effective and complementary approach for enhancing out-of-distribution detection systems.

REFERENCES

Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. *British Machine Vision Conference*, 2016.

Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 2008.

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

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

Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *International Conference on Computer Vision*, 2015.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Empirical Methods in Natural Language Processing*, 2014.

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

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. *Computer Vision and Pattern Recognition*, 2014.

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

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.

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

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*. ACM, 2013.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Machine Learning*, 2015.

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

Danijar Hafner, Dustin Tran, Alex Irpan, Timothy Lillicrap, and James Davidson. Reliable uncertainty estimates in deep neural networks using noise contrastive priors. *International Conference on Machine Learning Workshop*, 2018.

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

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations*, 2017.

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

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 2012.

Volodymyr Kuleshov and Percy Liang. Calibrated structured prediction. *Neural Information Processing Systems*, 2015.

Vijay Kumar, Gustavo Carneiro, and Ian Reid. Learning local image descriptors with deep siamese and triplet convolutional networks by minimizing global loss functions. *Computer Vision and Pattern Recognition*, 2016.

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.

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.

Si Liu, Risheek Garrepalli, Thomas Dietterich, Alan Fern, and Dan Hendrycks. Open category detection with PAC guarantees. In *Proceedings of International Conference on Machine Learning*, 2018.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017.

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.

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

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and Optimizing LSTM Language Models. *International Conference on Learning Representations*, 2018a.

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

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.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Computer Vision and Pattern Recognition*, 2015.

Khanh Nguyen and Brendan O'Connor. Posterior calibration and exploratory analysis for natural language processing models. *Empirical Methods in Natural Language Processing*, 2015.

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.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Machine Learning*, 2016.

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

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*, 2015.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *International Conference on Learning Representations*, 2017.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, 2013.

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

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *International Conference on Computer Vision*, 2017.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Large scale fine-grained categorization and the effectiveness of domain-specific transfer learning. *Yin Cui and Yang Song and Chen Sun and Andrew Howard and Serge Belongie*, 2018.

Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence*, 2008.

Antonio Torralba, Joshua B Tenenbaum, and Ruslan R Salakhutdinov. Learning to learn with compound hd models. In *Neural Information Processing Systems*, 2011.

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*, 2015.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *British Machine Vision Conference*, 2016.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*. Springer, 2014.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *Pattern Analysis and Machine Intelligence*, 2017.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. *International Conference on Learning Representations*, 2018.

A EXPANDED MULTICLASS RESULTS

Expanded mutliclass out-of-distribution detection results are in Table 8 and Table 9. These strong results suggest that future work should focus on new and harder datasets.

$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	FPR95 ↓		AUROC ↑		AUPR ↑	
		MSP	+OE	MSP	+OE	MSP	+OE
SVHN	Gaussian	5.4 ± 0.2	0.0 ± 0.0	98.2 ± 0.1	100.0 ± 0.0	90.5 ± 0.2	100.0 ± 0.0
	Bernoulli	4.4 ± 0.1	0.0 ± 0.0	98.6 ± 0.0	100.0 ± 0.0	91.9 ± 0.1	100.0 ± 0.0
	Blobs	3.7 ± 0.1	0.0 ± 0.0	98.9 ± 0.0	100.0 ± 0.0	93.5 ± 0.1	100.0 ± 0.0
	Icons-50	11.4 ± 1.0	0.3 ± 0.0	96.4 ± 0.2	99.8 ± 0.0	87.2 ± 0.3	99.2 ± 0.0
	Textures	7.2 ± 0.1	0.2 ± 0.0	97.5 ± 0.0	100.0 ± 0.0	90.9 ± 0.1	99.7 ± 0.0
	Places365	5.6 ± 0.2	0.1 ± 0.0	98.1 ± 0.1	100.0 ± 0.0	92.5 ± 0.1	99.9 ± 0.0
	LSUN	6.4 ± 0.2	0.1 ± 0.0	97.8 ± 0.1	100.0 ± 0.0	91.0 ± 0.1	99.9 ± 0.0
	CIFAR-10	6.0 ± 0.2	0.1 ± 0.0	98.0 ± 0.0	100.0 ± 0.0	91.2 ± 0.1	99.9 ± 0.0
	Chars74K	6.4 ± 0.3	0.1 ± 0.0	97.9 ± 0.1	100.0 ± 0.0	91.5 ± 0.2	99.9 ± 0.0
	Mean	6.28	**0.07**	97.95	**99.96**	91.12	**99.85**
CIFAR-10	Gaussian	14.4 ± 0.5	0.7 ± 0.0	94.7 ± 0.1	99.6 ± 0.0	70.0 ± 0.5	94.3 ± 0.1
	Rademacher	47.6 ± 1.0	0.5 ± 0.0	79.9 ± 0.3	99.8 ± 0.0	32.3 ± 0.4	97.4 ± 0.0
	Blobs	16.2 ± 0.3	0.6 ± 0.0	94.5 ± 0.1	99.8 ± 0.0	73.7 ± 0.3	98.9 ± 0.1
	Textures	42.8 ± 1.2	12.2 ± 0.7	88.4 ± 0.2	97.7 ± 0.1	58.4 ± 0.5	91.0 ± 0.2
	SVHN	28.8 ± 1.1	4.8 ± 0.3	91.8 ± 0.2	98.4 ± 0.0	66.9 ± 0.5	89.4 ± 0.2
	Places365	47.5 ± 3.0	17.3 ± 1.3	87.8 ± 0.4	96.2 ± 0.2	57.5 ± 0.8	87.3 ± 0.4
	LSUN	38.7 ± 1.5	12.1 ± 0.7	89.1 ± 0.3	97.6 ± 0.1	58.6 ± 0.7	89.4 ± 0.4
	CIFAR-100	43.5 ± 1.2	28.0 ± 1.2	87.9 ± 0.2	93.3 ± 0.2	55.8 ± 0.4	76.2 ± 0.4
	Mean	34.94	**9.50**	89.27	**97.81**	59.16	**90.48**
CIFAR-100	Gaussian	54.3 ± 0.5	12.1 ± 0.4	64.7 ± 0.3	95.7 ± 0.1	19.7 ± 0.1	71.1 ± 0.3
	Rademacher	39.0 ± 0.5	17.1 ± 0.4	79.4 ± 0.2	93.0 ± 0.1	30.1 ± 0.2	56.9 ± 0.2
	Blobs	58.0 ± 0.6	12.1 ± 0.5	75.3 ± 0.3	97.2 ± 0.1	29.7 ± 0.3	86.2 ± 0.4
	Textures	71.5 ± 1.6	54.4 ± 1.3	73.8 ± 0.5	84.8 ± 0.2	33.3 ± 0.6	56.3 ± 0.5
	SVHN	69.3 ± 0.4	42.9 ± 1.0	71.4 ± 0.4	86.9 ± 0.2	30.7 ± 0.5	52.9 ± 0.6
	Places365	70.4 ± 1.4	49.8 ± 1.1	74.2 ± 0.5	86.5 ± 0.3	33.8 ± 0.5	57.9 ± 0.8
	LSUN	74.0 ± 1.4	57.5 ± 1.5	70.7 ± 0.5	83.4 ± 0.2	28.8 ± 0.7	51.4 ± 0.5
	CIFAR-10	64.9 ± 1.1	62.1 ± 0.9	75.4 ± 0.3	75.7 ± 0.4	34.3 ± 0.3	32.6 ± 0.4
	Mean	62.66	**38.50**	73.11	**87.89**	30.05	**58.15**
Tiny ImageNet	Gaussian	72.6 ± 0.1	45.4 ± 0.3	33.7 ± 0.1	76.5 ± 0.2	12.3 ± 0.0	28.6 ± 0.2
	Rademacher	51.7 ± 0.1	49.0 ± 0.3	62.0 ± 0.2	65.1 ± 0.2	18.8 ± 0.1	20.0 ± 0.1
	Blobs	79.4 ± 0.3	0.0 ± 0.0	48.2 ± 0.3	100.0 ± 0.0	14.4 ± 0.1	99.9 ± 0.0
	Textures	76.4 ± 1.1	4.8 ± 1.0	70.4 ± 0.3	98.5 ± 0.2	31.4 ± 0.5	95.8 ± 0.2
	SVHN	52.3 ± 0.6	0.4 ± 0.0	80.8 ± 0.4	99.8 ± 0.0	42.8 ± 0.6	98.2 ± 0.1
	Places365	63.6 ± 1.1	0.4 ± 0.0	76.9 ± 0.3	99.8 ± 0.0	36.3 ± 0.7	99.3 ± 0.1
	LSUN	67.0 ± 1.5	0.4 ± 0.0	74.2 ± 0.4	99.9 ± 0.0	31.2 ± 0.4	99.5 ± 0.0
	ImageNet	67.3 ± 0.9	11.6 ± 1.1	72.8 ± 0.4	97.9 ± 0.1	30.0 ± 0.4	92.9 ± 0.3
	Mean	66.27	**13.99**	64.86	**92.18**	27.15	**79.26**

Table 8: Vision OOD example detection for the maximum softmax probability (MSP) baseline detector and the MSP detector after fine-tuning with Outlier Exposure (OE). All results are percentages and the result of 10 runs. Values are rounded so that 99.95% rounds to 100%. More results are in Appendix H.

Anomalous Data. For each in-distribution dataset $\mathcal{D}_{\text{in}}$, we comprehensively evaluate OOD detectors on artificial and real anomalous distributions $\mathcal{D}_{\text{out}}^{\text{test}}$ following Hendrycks & Gimpel (2017). For each learned distribution $\mathcal{D}_{\text{in}}$, the number of test distributions that we compare against is approximately double that of most previous works.

Gaussian anomalies have each dimension i.i.d. sampled from an isotropic Gaussian distribution. *Rademacher* anomalies are images where each dimension is -1 or 1 with equal probability, so each dimension is sampled from a symmetric Rademacher distribution. *Bernoulli* images have each pixel sampled from a Bernoulli distribution if the input range is $[0, 1]$. *Blobs* data consist in algorithmi-

$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	FPR90 ↓		AUROC ↑		AUPR ↑	
		MSP	+OE	MSP	+OE	MSP	+OE
20 Newsgroups	SNLI	38.2 ± 0.0	12.5 ± 0.0	87.7 ± 0.0	95.1 ± 0.0	71.3 ± 0.0	86.3 ± 0.0
	IMDB	45.0 ± 0.0	18.6 ± 0.0	79.9 ± 0.0	93.5 ± 0.0	42.6 ± 0.0	74.5 ± 0.0
	Multi30K	54.5 ± 1.6	3.2 ± 0.4	78.3 ± 0.7	97.3 ± 0.3	45.8 ± 0.9	93.7 ± 0.5
	WMT16	45.8 ± 0.7	2.0 ± 0.2	80.2 ± 0.3	98.8 ± 0.1	43.7 ± 0.6	96.1 ± 0.2
	Yelp	45.9 ± 0.7	3.9 ± 0.2	78.7 ± 0.5	97.8 ± 0.1	38.1 ± 1.0	87.9 ± 0.5
	EWT-A	36.1 ± 0.4	1.2 ± 0.1	86.2 ± 0.1	99.2 ± 0.0	58.2 ± 0.4	97.3 ± 0.1
	EWT-E	31.9 ± 0.7	1.4 ± 0.1	87.8 ± 0.2	99.2 ± 0.0	60.3 ± 0.5	97.2 ± 0.1
	EWT-N	41.7 ± 0.4	1.8 ± 0.0	83.1 ± 0.1	98.7 ± 0.0	46.2 ± 0.3	95.7 ± 0.0
	EWT-R	40.7 ± 0.7	1.7 ± 0.1	83.5 ± 0.2	98.9 ± 0.1	53.4 ± 0.6	96.6 ± 0.1
	EWT-W	44.5 ± 0.2	2.4 ± 0.1	81.1 ± 0.1	98.5 ± 0.0	39.0 ± 0.2	93.8 ± 0.1
	Mean	42.44	**4.86**	82.66	**97.71**	49.85	**91.91**
TREC	SNLI	18.2 ± 0.0	4.2 ± 0.0	94.0 ± 0.0	98.1 ± 0.0	81.9 ± 0.0	91.6 ± 0.0
	IMDB	49.6 ± 0.0	0.6 ± 0.0	78.0 ± 0.0	99.4 ± 0.0	44.2 ± 0.0	97.8 ± 0.0
	Multi30K	44.2 ± 5.5	0.3 ± 0.1	81.6 ± 1.2	99.7 ± 0.2	44.9 ± 2.5	99.0 ± 0.3
	WMT16	50.7 ± 1.7	0.2 ± 0.0	78.2 ± 1.4	99.8 ± 0.2	42.2 ± 2.4	99.4 ± 0.3
	Yelp	50.9 ± 1.5	0.4 ± 0.1	75.1 ± 1.8	99.7 ± 0.1	37.7 ± 2.4	96.1 ± 0.4
	EWT-A	45.7 ± 5.7	0.9 ± 0.4	82.4 ± 1.5	97.7 ± 1.2	53.1 ± 2.8	96.1 ± 1.4
	EWT-E	36.8 ± 7.6	0.4 ± 0.2	85.7 ± 0.7	99.5 ± 0.4	60.8 ± 3.4	99.1 ± 0.5
	EWT-N	44.3 ± 8.6	0.3 ± 0.1	84.2 ± 1.9	99.6 ± 0.4	58.8 ± 2.4	99.2 ± 0.5
	EWT-R	46.1 ± 2.7	0.4 ± 0.1	82.5 ± 1.0	99.5 ± 0.2	51.1 ± 3.0	98.8 ± 0.4
	EWT-W	50.1 ± 2.9	0.2 ± 0.0	79.8 ± 1.7	99.7 ± 0.2	47.8 ± 3.5	99.4 ± 0.3
	Mean	43.46	**0.78**	82.14	**99.28**	52.23	**97.64**
SST	SNLI	57.3 ± 0.0	33.4 ± 0.0	75.7 ± 0.0	86.8 ± 0.0	36.2 ± 0.0	52.0 ± 0.0
	IMDB	83.1 ± 0.0	32.6 ± 0.0	54.3 ± 0.0	85.9 ± 0.0	19.0 ± 0.0	51.5 ± 0.0
	Multi30K	81.3 ± 1.8	33.0 ± 3.0	58.5 ± 1.6	88.3 ± 0.8	21.4 ± 1.1	58.9 ± 1.8
	WMT16	76.0 ± 2.0	17.1 ± 1.7	60.2 ± 0.9	92.9 ± 0.3	21.4 ± 0.7	68.8 ± 1.2
	Yelp	82.0 ± 1.0	11.3 ± 0.7	54.2 ± 1.2	92.7 ± 0.3	18.8 ± 0.7	60.0 ± 1.2
	EWT-A	72.6 ± 2.6	33.6 ± 2.8	62.7 ± 1.6	87.2 ± 0.8	21.4 ± 1.5	53.8 ± 2.5
	EWT-E	68.1 ± 1.9	26.5 ± 2.0	68.5 ± 0.6	90.4 ± 0.5	27.0 ± 0.8	63.7 ± 1.0
	EWT-N	73.8 ± 2.3	27.2 ± 2.1	63.8 ± 1.1	90.1 ± 0.4	22.6 ± 1.0	62.0 ± 0.9
	EWT-R	79.6 ± 2.0	41.4 ± 2.8	58.1 ± 1.2	85.6 ± 0.5	20.3 ± 0.8	54.7 ± 1.2
	EWT-W	74.8 ± 1.6	17.2 ± 1.3	60.3 ± 0.8	92.8 ± 0.3	21.2 ± 0.6	66.9 ± 1.1
	Mean	74.86	**25.31**	61.61	**90.07**	22.93	**64.37**

Table 9: NLP OOD example detection for the maximum softmax probability (MSP) baseline detector and the MSP detector after fine-tuning with Outlier Exposure (OE). All results are percentages and the result of 10 runs. Values are rounded so that 99.95% rounds to 100%.

cally generated amorphous shapes with definite edges. *Icons-50* is a dataset of icons and emojis (Hendrycks & Dietterich, 2018); icons from the "Number" class are removed. *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 when possible each image is cropped with bounding box information as in Tiny ImageNet. With *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. *Chars74K* is a dataset of photographed characters in various styles; letters such as "O" and "l" were removed since they can look like numbers.

SNLI is a dataset of predicates and hypotheses for natural language inference. We use the hypotheses for $\mathcal{D}_{\text{out}}^{\text{OE}}$. *IMDB* is a sentiment classification dataset of movie reviews, with similar statistics to those of SST. *Multi30K* is a dataset of English-German image descriptions, of which we use the English descriptions. *WMT16* is the English portion of the test set from WMT16. *Yelp* is a dataset of restaurant reviews. *English Web Treebank (EWT)* consists of five individual datasets: Answers (A), Email (E), Newsgroups (N), Reviews (R), and Weblog (W). Each contains examples from the indicated domain.

B ARCHITECTURES AND TRAINING DETAILS

For CIFAR-10, CIFAR-100, and Tiny ImageNet classification experiments, we use a 40-2 Wide Residual Network (Zagoruyko & 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 (Loshchilov & Hutter, 2017). During fine-tuning of the entire network, we again use a cosine learning rate schedule but with an initial learning rate of 0.001. We use standard flipping and data cropping augmentation, Nesterov momentum, and ℓ_2 weight decay with a coefficient of 5×10^{-4}. SVHN architectures are 16-4 Wide ResNets trained for 20 epochs with an initial learning rate of 0.01 and no data augmentation. Outlier Exposure fine-tuning occurs for 10 epochs with each epoch being the length of in-distribution epoch, so that Outlier Exposure completes quickly and does involve reading the entire $\mathcal{D}_{\text{out}}^{\text{OE}}$ dataset.

C CALIBRATION EVALUATION METRICS

In order to evaluate a multiclass classifier's calibration, we present three metrics. First we establish context. For input example $X \in \mathcal{X}$, let $Y \in \mathcal{Y} = \{1, 2, \ldots, k\}$ 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 MAD 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 less severely penalizes large confidence-accuracy deviations is the Mean Absolute Value Calibration error, written $\mathbb{E}_C[|\mathbb{P}(Y = \widehat{Y}|C = c) - c|]$. The MAD 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 b bins $\{B_1, B_2, \ldots, B_b\}$ with approximately 100 samples in each bin. Unlike Guo et al. (2017), bins are not equally spaced since the distribution of confidence values is not uniform but dynamic. Concretely, the RMS Calibration Error is estimated with the numerically stable formula

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

Along similar lines, the MAD Calibration Error is estimated with

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

Soft F1 Score. If a classifier makes only a 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 & Gimpel, 2017) is suited for measuring the calibration of a system where there is an acute imbalance between mistaken and correct decisions. Since we treat mistakes a positive examples, we can write the model's confidence that the examples are anomalous with $c_a = (1 - c_1, 1 - c_2, \ldots, 1 - c_n)$. To indicate that an example is positive (mistaken), we use the vector $m \in \{0, 1\}^n$ such that $m_i = \mathbb{1}(y_i \neq \widehat{y}_i)$ for $1 \leq i \leq n$. Then the Soft F1 score is

$$\frac{c_a^{\mathsf{T}} m}{\mathbf{1}^{\mathsf{T}}(c_a + m)/2}.$$

D ADDITIONAL ROC AND PR CURVES

In Figure 3, we show additional PR and ROC Curves using the Tiny ImageNet dataset and various anomalous distributions.

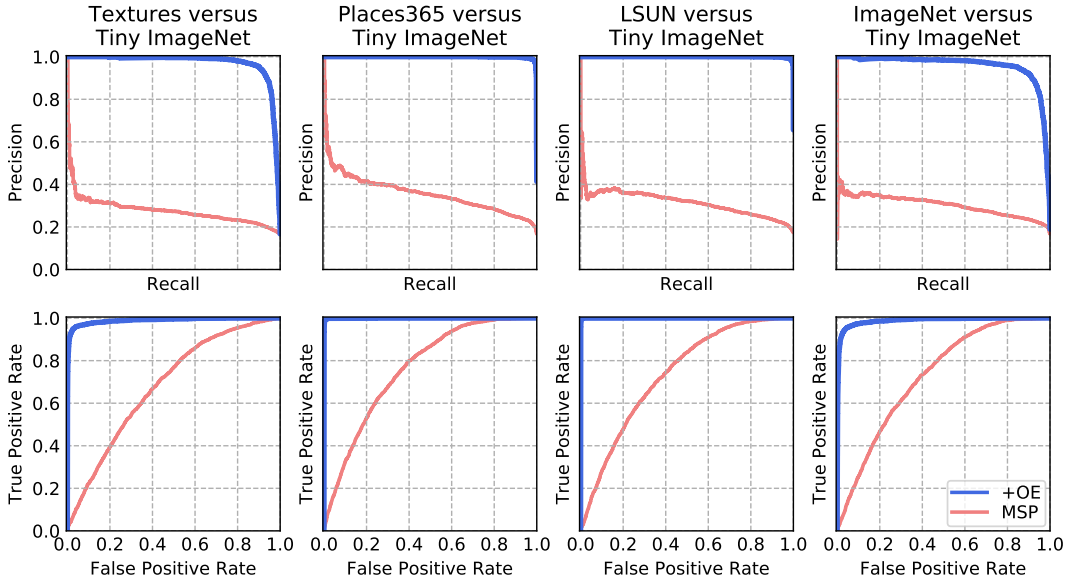

Figure 3: ROC curves with Tiny ImageNet as $\mathcal{D}_{\text{in}}$ and Textures, Places365, LSUN, and ImageNet as $\mathcal{D}_{\text{out}}^{\text{test}}$. Figures show the curves corresponding to the maximum softmax probability (MSP) baseline detector and the MSP detector with Outlier Exposure (OE).

E Posterior Rescaling

While temperature tuning improves calibration, the confidence estimate c_T cannot be less than $1/k$, k the number of classes. For an out-of-distribution example like Gaussian Noise, a good model should have no confidence in its prediction over k classes. One possibility is to add a reject option, or a $(k+1)$st class, which we cover in Section 5. A simpler option we found 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. As Table 10 shows, this simple 0-1 posterior rescaling technique consistently improves calibration, and the model fine-tuned with OE using temperature tuning and posterior rescaling achieved large calibration improvements.

	RMS Calib. Error ↓			MAD Calib. Error ↓			Soft F1 Score ↑		
$\mathcal{D}_{\text{in}}$	Temp	+Rescale	+OE	Temp	+Rescale	+OE	Temp	+Rescale	+OE
SVHN	22.3	20.8	3.0	10.9	10.1	1.0	52.1	56.1	92.7
CIFAR-10	19.4	17.8	6.7	12.6	11.7	4.1	39.9	42.8	73.9
CIFAR-100	14.7	14.4	8.1	11.3	11.1	6.4	52.8	53.1	66.1
Tiny ImageNet	12.0	11.9	6.9	9.0	8.8	4.8	62.9	63.1	72.3

Table 10: Calibration results for the softmax temperature tuning baseline, the same baseline after adding Posterior Rescaling, and temperature tuning + Posterior Rescaling + OE.

F Training from Scratch with Outlier Exposure Usually Improves Performance

Elsewhere we show results for pre-trained networks that are fine-tuned with OE. However, a network trained from scratch which simultaneously trains with OE tends to give superior results. For example, a CIFAR-10 Wide ResNet trained normally obtains a classification error rate of 5.16% and an FPR95 of 34.94%. Fine-tuned, this network has an error rate of 5.27% and an FPR95 of 9.50%. Yet if we instead train the network from scratch and expose it to outliers as it trains, then the error rate is 4.26% and the FPR95 is 6.15%. This architecture corresponds to a 9.50% RMS calibration error with OE fine-tuning, but by training with OE from scratch the RMS calibration error is 6.15%.

Compared to fine-tuning, training a network in tandem with OE tends to produce a network with a better error rate, calibration, and OOD detection performance. The reason why we use OE for fine-tuning is because training from scratch requires more time and sometimes more GPU memory than fine-tuning.

G OE WORKS ON OTHER VISION ARCHITECTURES

Outlier Exposure also improves vision OOD detection performance for more than just Wide ResNets. Table 11 shows that Outlier Exposure also improves vision OOD detection performance for "All Convolutional Networks."

	FPR95 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	MSP	+OE	MSP	+OE	MSP	+OE
SVHN	6.84	0.08	98.1	100.0	90.9	99.8
CIFAR-10	28.4	14.0	90.1	96.7	58.9	87.3
CIFAR-100	57.5	43.3	76.7	85.3	33.9	51.3
Tiny ImageNet	75.5	25.0	55.4	82.9	25.6	75.3

Table 11: Results using an All Convolutional Network architectures. Results are percentages and an average of 10 runs.

H OUTLIER EXPOSURE WITH $H(\mathcal{U}; p)$ SCORES DOES BETTER THAN WITH MSP SCORES

While $-\max_c p_c(x)$ tends to be a discriminative OOD score for example x, models with OE can do better by using $-H(\mathcal{U}; p(x))$ instead. This alternative accounts for classes with small probability mass rather than just the class with most mass. Additionally, the model with OE is trained to give anomalous examples a uniform posterior not just a lower MSP. This simple change roundly aids performance as shown in Table 12. This general performance improvement is most pronounced on datasets with many classes. For instance, when $\mathcal{D}_{\text{out}}^{\text{test}}$ = Tiny ImageNet and $\mathcal{D}_{\text{out}}^{\text{test}}$ = Gaussian, swapping the MSP score with the $H(\mathcal{U}; p(x))$ score increases the AUROC 76.5% to 97.1%.

	FPR95 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	MSP	$H(\mathcal{U}; p)$	MSP	$H(\mathcal{U}; p)$	MSP	$H(\mathcal{U}; p)$
CIFAR-10	9.50	9.04	97.81	97.92	90.48	90.85
CIFAR-100	38.50	33.31	87.89	88.46	58.15	58.30
Tiny ImageNet	13.99	7.45	92.18	95.45	79.26	85.71

Table 12: Comparison between the maximum softmax probability (MSP) and $H(\mathcal{U}; p)$ OOD scoring methods on a network fine-tuned with OE. Results are percentages and an average of 10 runs. For example, CIFAR-10 results are averaged over "Gaussian," "Rademacher," . . ., or "CIFAR-100" measurements.

I EXPANDED LANGUAGE MODELING RESULTS

Detailed OOD detection results with language modeling datasets are shown in Table 13.

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. Other NLP $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets we consider do not satisfy the language modeling assumption of continuity in the examples, so we do not evaluate on them.

$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	FPR90 ↓ BPC	FPR90 ↓ +OE	AUROC ↑ BPC	AUROC ↑ +OE	AUPR ↑ BPC	AUPR ↑ +OE
PTB Char	Answers	96.9	49.93	82.1	89.6	81.0	89.3
	Email	99.5	90.64	80.6	88.6	79.4	89.1
	Newsgroup	99.8	99.39	75.2	85.0	73.3	85.5
	Reviews	99.0	74.64	80.8	89.0	79.2	89.6
	Weblog	100.0	100.0	68.9	79.2	67.3	80.1
Mean		99.0	**89.4**	77.5	**86.3**	76.0	**86.7**
PTB Word	Answers	41.4	3.65	81.4	98.0	40.5	94.7
	Email	64.9	0.17	78.1	99.6	44.5	98.9
	Newsgroup	54.9	0.17	77.8	99.5	39.8	98.3
	Reviews	30.5	0.85	88.0	98.9	53.6	96.8
	Weblog	50.8	0.08	80.7	99.9	41.5	99.7
Mean		48.5	**0.98**	81.2	**99.2**	44.0	**97.8**

Table 13: OOD detection results on Penn Treebank examples and English Web Treebank outliers. All results are percentages.