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

It is important to detect and handle anomalous inputs in 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 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, and OE improves the performance of density estimation based detectors. We also find that OE improves classifier calibration in the presence of anomalous inputs.

1 Introduction

Detecting anomalous data is important in many applications of machine learning systems [23, 11]. At deployment time, when a data point comes from a different distribution than the learned distribution, the anomalous data may be of interest to the system operators. This can occur in discovering novel astronomical phenomena, encountering unknown diseases, or detecting sensor failure. Also, typical inputs that have been corrupted through noise or other distortions may require special handling. This is especially true of deep neural networks, as these systems have been demonstrated to lack robustness to input noise and distortions [1, 14].

Deep learning systems [18] 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 [31]. This silent error invalidates the use of prediction probabilities as calibrated confidence estimates [13]. This 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* examples [15, 10, 22, 23]. Intuitively, distinguishing between *in-distribution* and out-of-distribution examples is possible without necessarily modeling the full data distribution. These approaches have been demonstrated to work surprisingly well for complex input spaces, such as images, text, and speech.

In this paper, we investigate a complementary method where we *expose* anomaly detectors to data that we would like them to deem anomalous. The basic intuition is that exposing anomaly detectors to outliers, rather than just to training examples, can enable learning a more conservative concept of the class structure and teach the network to treat unusual inputs differently. Thus we expose anomaly detectors to a diverse dataset of real outliers that are outside of the training distribution. These examples do not need to be labeled in any way. Thus, while the dataset of in-distribution examples may be limited in size, one can typically find massive quantities of out-of-distribution data to train against. We call this approach Outlier Exposure (OE).

Through extensive experiments using a range of anomaly detectors, we find that OE consistently provides a substantial boost in anomaly detection performance. We use several large datasets for OE in different settings. We also find that OE makes density estimators such as PixelCNN++

*Work done while at OSU.

competitive with anomaly detectors with a classifier backbone, unlike before. Finally, we 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 conveniently 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, Li, and Srikant [2018] present a method which can improve performance of anomaly 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 [12]. 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 [35] 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. Many other works [41, 4, 38] 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 in order to detect anomalies. Torralba, Tenenbaum, and Salakhutdinov [2011] pre-trains unsupervised deep models on a large database of web images for stronger features. Radford, Jozefowicz, and Sutskever [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 [36] endows the network with general representations useful in myriad fine-tuning applications. Chen and Gupta, Mahajan et al. [2015, 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 that encourages a gap between the scores of in-distribution samples and anomalies. Once fine-tuned with OE, 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. With 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 [39] overlapping with the CIFAR datasets. With 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 [29] as $\mathcal{D}_{\text{out}}^{\text{OE}}$. When performing OE, we sample in-distribution samples from $\mathcal{D}_{\text{in}}$ and equally many samples from $\mathcal{D}_{\text{out}}^{\text{OE}}$; the entire OE process usually requires only a few minutes.

4 Experiments

4.1 Evaluating Anomaly Detection Methods

To evaluate different anomaly 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 anomaly score thresholds. The AUROC can be thought of as the probability that an anomalous example is given a higher anomaly score than a in-distribution example [9]. Thus, a higher AUROC

is better, and an uninformative detector has an AUROC of 50%. Now, the AUPR is useful when anomalous examples are infrequent [26], 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 $\mathcal{D}_{\text{in}}$ test examples in all our experiments is $1{:}5$.

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 [23, 20, 2] 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

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 [30] 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 [17] 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. For example, 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 the average training image's channel means and standard deviations.
Tiny ImageNet. The Tiny ImageNet dataset [16] is a subset of the ImageNet [36] 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 [7]. 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.

4.2.2 Anomalous Data

For each in-distribution dataset $\mathcal{D}_{\text{in}}$, we comprehensively evaluate anomaly detectors on artificial and real anomalous distributions $\mathcal{D}_{\text{out}}^{\text{test}}$ following Hendrycks and Gimpel [2017].

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. *Blobs* data consist in algorithmically generated amorphous shapes with definite edges. *Textures* is a dataset of describable textural images [8]. *Places365* consists in images for scene recognition rather than object recognition [45]. *LSUN* is another scene understanding dataset with fewer classes than Places365 [42]. *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.

4.2.3 Adding Outlier Exposure to Popular Anomaly Detection Methods

In what follows, we using existing anomaly detection techniques and enhance their performance with Outlier Exposure. For brevity, we place MNIST and SVHN results in the Supplementary Materials. Throughout the following experiments, we let $x \in \mathcal{X}$ be a model 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, $1^\mathsf{T} p(x) = 1$ and $p(x) \succeq 0$ so that p lies on a probability simplex.

Maximum Softmax Probability. Consider the maximum softmax probability baseline [15] which gives an input x the anomaly score $-\max_c p_c(x)$. By exposing the classifier to outliers, its anomaly scoring significantly improves. We perform Outlier Exposure by fine-tuning a pre-trained classi-

fier p so that it has a uniform posterior on $\mathcal{D}_{\text{out}}^{\text{OE}}$ samples. Formally, the fine-tuning objective is $\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{in}}}[-\log p_y(x)] + 0.5\mathbb{E}_{x\sim\mathcal{D}_{\text{out}}^{\text{OE}}}[H(U; p(x))]$, where H is the cross entropy and 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 U works well enough.

$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	FPR95 ↓		AUROC ↑		AUPR ↑	
		MSP	+OE	MSP	+OE	MSP	+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 detector and the MSP detector after fine-tuning with Outlier Exposure (OE). All results are percentages.

The fine-tuning objective has its OE term weighted by 0.5. Unlike Liang, Li, and Srikant [22] and Lee et al. [21] and like Hendrycks and Gimpel [2017], we do not tune our hyperparameters for each $\mathcal{D}_{\text{out}}^{\text{test}}$ distribution since this is in the spirit of keeping $\mathcal{D}_{\text{out}}^{\text{test}}$ unknown like real-world anomalies. Instead, the 0.5 coefficient 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.

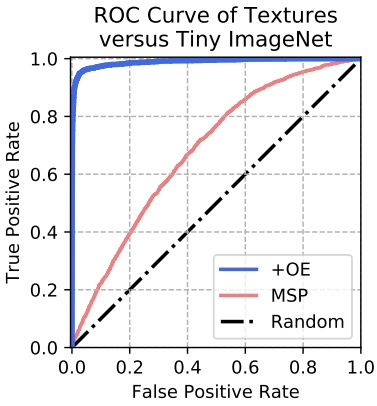

Figure 1: ROC curve with Tiny ImageNet ($\mathcal{D}_{\text{in}}$) and Textures ($\mathcal{D}_{\text{out}}^{\text{test}}$).

For these experiments, we train Wide Residual Networks and then fine-tune network copies with OE for 10 epochs. During each epoch of OE, we use training set examples and equally many outliers. This way OE requires only a few minutes on a single GPU. Networks trained with CIFAR-10/100 are exposed to outliers from 80 Million Tiny Images, and the Tiny ImageNet classifier is exposed to ImageNet-22K outliers. Further architectural and training details are in the Supplementary Materials. Results are shown in Table 1 and Figure 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.

While $-\max_c p_c(x)$ tends to be a discriminative anomaly score for example x, models with OE can do better by using $-H(U; p(x))$ 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 2. This general performance improvement is most pronounced on datasets with many classes. For instance, when $\mathcal{D}_{\text{out}}^{\text{test}}$ = Tiny ImageNet and $\mathcal{D}_{\text{out}}^{\text{test}}$ = Rademacher, swapping the MSP score with the $H(U; p(x))$ score increases the AUROC 57.1% to 89.5%.

	FPR95 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	MSP	$H(U; p)$	MSP	$H(U; p)$	MSP	$H(U; p)$
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 (MSP) and $H(U; p)$ anomaly scoring methods on a network with OE. Results are percentages and averages. For example, CIFAR-10 results are averaged over "Gaussian," "Rademacher," . . ., or "CIFAR-100" measurements.

Confidence Branch. A recently proposed anomaly detection technique [10] involves appending an anomaly 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 and 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 anomaly 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 gathering large quantities of real data from the Internet, but it is possible to generate *synthetic* outliers. Lee et al. [2018] 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 briefly mention that a cautiously trained 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 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 publicly available code. We 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 anomaly score. Now, 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. Last, we take the network trained in tandem with a GAN and fine-tune it with OE. Table 4 shows the large gains from using with a real, massive, diverse database over just using synthetic samples from a cautiously trained GAN.

4.3 Density Estimation

$\mathcal{D}_{\text{in}}$	FPR95 ↓			AUROC ↑			AUPR ↑		
	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 anomaly 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.

Density estimators learn a probability density function over the source 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 [46]. We show the ability of OE to make density estimates useful for anomaly detection since OE can make density estimators behave reasonably at the task of density estimation.

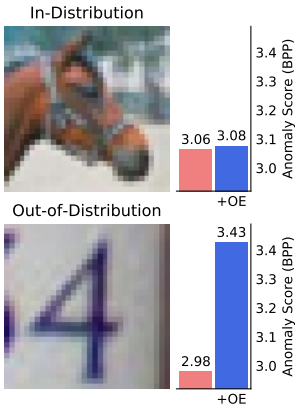

Figure 2: Anomaly scores from PixelCNN++ on images from CIFAR-10 and SVHN.

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++ [37] as a baseline anomaly detector, and we train it on CIFAR-10. The bits-per-pixel (BPP) is used as an anomaly score. This is then fine-tuned for 1 epoch using OE on 80 Million Tiny Images. 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 per pixel. Results are shown in Table 5. For all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets, OE significantly improves results.

$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	FPR95 ↓		AUROC ↑		AUPR ↑	
		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: Anomaly 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.

Language Modelling. We next explore using OE on language models. We use QRNN [28, 27] language models as baseline anomaly detectors. The bits per character (BPC) is used as an anomaly 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 term in the loss.

For $\mathcal{D}_{\text{in}}$, we convert Penn Treebank into a language modeling corpus, split into sequences of length 70 for backpropagation for word-level models, and 150 for 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 [5], which contains text from five different domains: Yahoo! Answers, emails, newsgroups, product reviews, and weblogs.

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 by leaving out characters which 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.

$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	FPR90 ↓		AUROC ↑		AUPR ↑	
		BPC	+OE	BPC	+OE	BPC	+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 6: Anomaly detection results on Penn Treebank examples and English Web Treebank outliers. All results are percentages.

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 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 [13, 32, 19] considers 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 estimate a classifier's confidence. 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 $l \in \mathbb{R}^k$, such as the estimate $\sigma(\max_i l_i) \in [0, 1]$. Another common confidence estimate is $\max_i \left[\exp(l_i)/\sum_{j=1}^k \exp(l_j)\right]$ which has confidence values in $[1/k, 1]$, k the number of classes.

Softmax Temperature Tuning. A confidence estimation technique shown to work well consistently simply involves adjusting the softmax temperature [13]. Specifically, they compute their 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, they first hold out a validation set. Then, using convex optimization software, they find the T which minimizes the softmax to labels 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 possibility is to add a reject 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 Ima-

$\mathcal{D}_{\text{in}}$	RMS Calib. Error ↓			MAV Calib. Error ↓			Soft F1 Score ↑		
	Temp	+Rescale	+OE	Temp	+Rescale	+OE	Temp	+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 Posterior Rescaling + OE.

geNet classifiers with 6000, 5000, 5000, 5000, and 10000 training examples held out, respectively. 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 improvements. On this problem Outlier Exposure noticeably 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%. Next, an alternative anomaly detection formulation is to give classifiers a "reject class" [3]. 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 detect novel anomalies. Clearly $\mathcal{D}_{\text{out}}^{\text{OE}}$ diversity matters. 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.

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 anomaly detectors, including classifiers and density estimators of images and text. We also show that Outlier Exposure improves the calibration of neural network classifiers when anomalous data is present. Outlier Exposure can be applied with low overhead and substantial performance gains to existing systems. Our results suggest that Outlier Exposure is a simple, general-purpose, and complementary approach for enhancing current and future anomaly detection systems.

References

[1] Aharon Azulay and Yair Weiss. "Why do deep convolutional networks generalize so poorly to small image transformations?" In: *arXiv preprint* (2018).

[2] Vassileios Balntas et al. "Learning local feature descriptors with triplets and shallow convolutional neural networks". In: *BMVC* (2016).

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

[4] Abhijit Bendale and Terrance Boult. "Towards Open Set Deep Networks". In: *Computer Vision and Pattern Recognition* (2016).

[5] Ann Bies et al. *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*. 2015, pp. 1431–1439.

[7] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. "A Downsampled Variant of ImageNet as an Alternative to the CIFAR datasets". In: *arXiv preprint* (2017).

[8] M. Cimpoi et al. "Describing Textures in the Wild". In: *Computer Vision and Pattern Recognition* (2014).

[9] Jesse Davis and Mark Goadrich. "The Relationship Between Precision-Recall and ROC Curves". In: *International Conference on Machine Learning*. 2006.

[10] Terrance DeVries and Graham W Taylor. "Learning Confidence for Out-of-Distribution Detection in Neural Networks". In: *arXiv preprint arXiv:1802.04865* (2018).

[11] Andrew F Emmott et al. "Systematic construction of anomaly detection benchmarks from real data". In: *Proceedings of the ACM SIGKDD workshop on outlier detection and description*. ACM. 2013, pp. 16–21.

[12] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *International Conference on Machine Learning*. 2015.

[13] Chuan Guo et al. "On Calibration of Modern Neural Networks". In: *International Conference on Machine Learning* (2017).

[14] Dan Hendrycks and Thomas Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Surface Variations". In: *arXiv preprint* (2018).

[15] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: 2017.

[16] Justin Johnson et al. *Tiny ImageNet Visual Recognition Challenge*. URL: https://tiny-imagenet.herokuapp.com.

[17] Alex Krizhevsky and Geoffrey Hinton. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, 2009.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Neural Information Processing Systems* (2012).

[19] Volodymyr Kuleshov and Percy Liang. "Calibrated Structured Prediction". In: *Neural Information Processing Systems* (2015).

[20] Vijay Kumar B G, Gustavo Carneiro, and Ian Reid. "Learning Local Image Descriptors with Deep Siamese and Triplet Convolutional Networks by Minimizing Global Loss Functions". In: *Computer Vision and Pattern Recognition* (2016).

[21] Kimin Lee et al. "Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples". In: *International Conference on Learning Representations* (2018).

[22] Shiyu Liang, Yixuan Li, and R. Srikant. "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks". In: *International Conference on Learning Representations* (2018).

[23] Si Liu et al. "Open Category Detection with PAC Guarantees". In: *Proceedings of International Conference on Machine Learning*. 2018.

[24] Ilya Loshchilov and Frank Hutter. *SGDR: Stochastic Gradient Descent with Warm Restarts*. 2017.

[25] Dhruv Mahajan et al. "Exploring the Limits of Weakly Supervised Pretraining". In: *arXiv preprint* (2018).

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

[27] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. "An Analysis of Neural Language Modeling at Multiple Scales". In: *arXiv preprint arXiv:1803.08240* (2018).

[28] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. "Regularizing and Optimizing LSTM Language Models". In: *International Conference on Learning Representations* (2018).

[29] Stephen Merity et al. "Pointer sentinel mixture models". In: *International Conference on Learning Representations* (2017).

[30] Yuval Netzer et al. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. 2011.

[31] 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*. 2015, pp. 427–436.

[32] Khanh Nguyen and Brendan O'Connor. "Posterior calibration and exploratory analysis for natural language processing models". In: *Empirical Methods in Natural Language Processing* (2015).

[33] Joan Pastor-Pellicer et al. "F-Measure as the Error Function to Train Neural Networks". In: *International Work-Conference on Artificial Neural Networks (IWANN)*. 2013.

[34] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. "Learning to Generate Reviews and Discovering Sentiment". In: *arXiv preprint arXiv:1704.01444* (2017).

[35] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *International Conference on Machine Learning* (2016).

[36] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[37] Tim Salimans et al. "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications". In: *International Conference on Learning Representations* (2017).

[38] Akshayvarun Subramanya, Suraj Srinivas, and R.Venkatesh Babu. "Confidence Estimation In Deep Neural Networks Via Density Modelling". In: *arXiv preprint* (2017).

[39] Antonio Torralba, Rob Fergus, and William T Freeman. "80 million tiny images: A large data set for nonparametric object and scene recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 30.11 (2008), pp. 1958–1970.

[40] Antonio Torralba, Joshua B Tenenbaum, and Ruslan R Salakhutdinov. "Learning to learn with compound HD models". In: *Neural Information Processing Systems*. 2011, pp. 2061–2069.

[41] 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.

[42] Fisher Yu et al. "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop". In: *CoRR* abs/1506.03365 (2015). arXiv: 1506.03365.

[43] Sergey Zagoruyko and Nikos Komodakis. "Wide Residual Networks". In: *CoRR* abs/1605.07146 (2016).

[44] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[45] Bolei Zhou et al. "Places: A 10 million Image Database for Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[46] Bo Zong et al. "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection". In: *International Conference on Learning Representations* (2018).

A MNIST and SVHN Multiclass Results

Here we evaluate Outlier Exposure on MNIST and SVHN. Although we use the natural image dataset 80 Million Tiny Images as the outlier distribution $\mathcal{D}_{\text{out}}^{\text{OE}}$, anomaly detection performance still improves on these datasets of digits. Results are in Table 8 and Table 9. Descriptions of previously unmentioned $\mathcal{D}_{\text{out}}^{\text{test}}$ distributions are as follows.

Anomalous Data. *Bernoulli* images have each pixel sampled from a Bernoulli distribution if the input range is $[0, 1]$. *Icons-50* is a dataset of icons [14]; icons from the "Number" class are removed. *Fashion-MNIST* is a drop-in replacement for MNIST, where the images and classes correspond to apparel rather than digits. $1 - $ *MNIST* is an inverted dataset obtained by taking $(1 - x)$ for each data point x in the MNIST dataset so that the background and pen ink colors swap. *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 such as "O" and "l" were removed since they can look like numbers.

		FPR99 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	MSP	+OE	MSP	+OE	MSP	+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
	$1 -$ 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 results are percentages.

	FPR99 ↓		AUROC ↑		AUPR ↑	
$\mathcal{D}_{\text{in}}$	MSP	$H(U; p)$	MSP	$H(U; p)$	MSP	$H(U; p)$
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 $H(U; p)$ anomaly scoring techniques. Both scoring techniques are for a model fine-tuned with OE. All results are percentages and averaged across all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets.

B Architectures and Training Details

For CIFAR-10 and CIFAR-100 classification experiments, we use a 40-4 Wide Residual Network [43]. 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 [24]. 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. Outlier Exposure fine-tuning occurs for 10 epochs with each epoch of the length

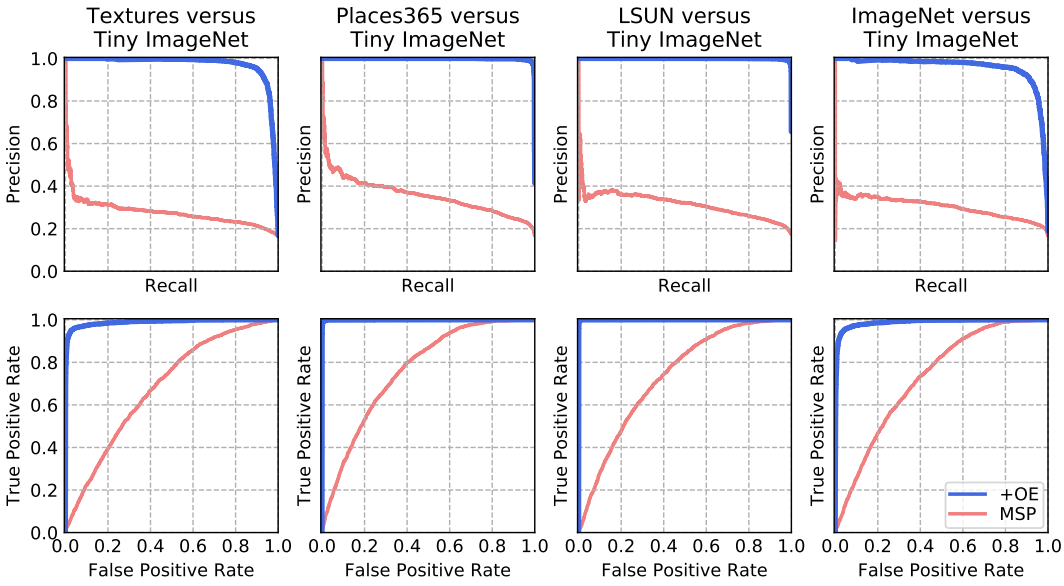

Figure 3: ROC curves with Tiny ImageNet as $\mathcal{D}_{\text{in}}$ and Textures, Places365, LSUN, and ImageNet as $\mathcal{D}_{\text{out}}^{\text{test}}$. Figures show the curves corresponding to the maximum softmax probability (MSP) baseline detector and the MSP detector with Outlier Exposure (OE).

of in-distribution epoch, so that Outlier Exposure completes quickly and does involve reading an entire database. An exception is SVHN; SVHN is fine-tuned with OE for 5 epochs.

C Additional ROC and PR Curves

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

D 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 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 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 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 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 MAV 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 [33, 15] 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

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

E Full Calibration Results

Expanded calibration results are available in Table 10.

$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	RMS Calib. Error ↓			MAV Calib. Error ↓			Soft F1 Score ↑		
		Temp	+Rescale	+OE	Temp	+Rescale	+OE	Temp	+Rescale	+OE
MNIST	Gaussian	10.5	9.0	1.7	4.3	3.8	0.8	65.5	68.7	85.2
	Bernoulli	12.2	11.0	2.6	5.4	5.0	1.2	57.1	61.1	83.0
	CIFAR-10	14.3	12.6	2.1	7.1	6.4	1.0	64.0	68.2	88.6
	Icons-50	14.7	12.7	2.2	7.4	6.5	1.1	63.6	67.2	88.2
	Fashion-MNIST	15.8	15.2	7.4	9.0	8.6	3.5	54.7	58.2	78.7
	$1-$ MNIST	17.2	16.7	6.6	9.7	9.5	2.8	50.9	53.6	81.2
	notMNIST	12.6	12.5	9.5	7.5	7.5	5.3	40.1	43.3	57.4
	Omniglot	11.8	12.1	11.5	6.9	6.9	6.8	45.2	47.6	47.2
	Mean	13.6	12.7	**5.4**	7.1	6.8	**2.8**	55.1	58.5	**76.2**
SVHN	Gaussian	15.4	14.4	2.4	6.1	5.7	0.7	51.6	54.4	89.6
	Bernoulli	15.5	14.5	2.4	6.1	5.7	0.7	51.1	54.6	89.6
	Blobs	14.8	13.4	2.5	5.7	5.2	0.7	54.9	58.7	89.6
	Icons-50	22.9	21.4	1.5	11.4	10.6	0.6	48.6	53.0	93.9
	Textures	14.3	13.2	1.6	6.0	5.4	0.7	52.3	56.5	89.0
	Places365	14.4	13.1	2.2	5.8	5.3	0.7	53.7	57.8	89.5
	LSUN	15.0	14.0	2.2	6.1	5.7	0.7	51.1	54.1	89.5
	CIFAR-10	22.3	20.8	1.9	10.8	10.0	0.7	52.5	56.4	94.2
	Chars74K	16.4	15.4	6.9	7.2	6.8	2.7	39.1	42.5	73.7
	Mean	16.8	15.6	**2.6**	7.2	6.7	**0.9**	50.5	54.2	**88.7**
CIFAR-10	Gaussian	22.7	21.4	4.2	15.3	14.5	2.5	31.6	34.2	78.2
	Rademacher	23.5	21.8	4.6	14.5	13.7	2.6	36.5	39.6	81.2
	Blobs	19.3	17.7	4.0	12.1	11.1	2.4	46.2	49.1	78.0
	Textures	19.0	17.3	5.1	12.5	11.3	3.1	44.0	47.3	74.9
	SVHN	20.3	18.5	7.2	12.4	11.3	4.3	45.4	48.5	71.3
	Places365	18.8	17.3	6.9	12.5	11.5	4.4	43.4	46.2	71.0
	LSUN	19.4	17.8	6.6	12.9	11.6	4.1	42.1	46.1	72.0
	CIFAR-100	19.4	17.5	10.7	13.3	11.9	7.0	39.7	43.8	61.9
	Mean	20.3	18.6	**6.2**	13.2	12.1	**3.8**	41.1	44.4	**73.6**
CIFAR-100	Gaussian	17.9	17.5	6.9	14.0	13.7	5.1	54.5	55.0	71.6
	Rademacher	14.6	14.2	8.6	14.6	11.1	6.1	59.1	59.8	73.8
	Blobs	10.1	9.5	7.4	8.5	7.9	5.6	64.9	65.8	72.8
	Textures	13.5	13.4	11.6	11.4	11.2	9.3	59.2	59.2	62.8
	SVHN	14.2	14.1	12.2	11.7	11.4	9.2	58.6	59.0	62.3
	Places365	13.8	13.6	10.4	11.8	11.4	8.1	58.4	58.9	64.9
	LSUN	14.5	14.5	10.4	12.3	12.3	8.2	57.3	57.3	64.6
	CIFAR-10	14.0	13.6	14.3	11.8	11.4	11.4	58.5	59.0	58.9
	Mean	14.1	13.8	**10.2**	12.0	11.3	**7.9**	58.8	59.3	**66.5**
Tiny ImageNet	Gaussian	11.2	11.0	4.4	8.0	7.9	3.3	64.2	64.4	70.8
	Rademacher	10.1	9.9	7.3	7.7	7.6	5.3	64.7	64.9	68.5
	Blobs	10.2	10.1	3.7	8.3	8.1	2.6	63.2	63.4	74.5
	Textures	8.7	8.4	3.8	7.4	7.1	2.7	63.7	63.9	73.5
	SVHN	4.8	4.6	3.7	3.7	3.7	2.6	70.6	70.9	74.5
	Places365	7.1	7.0	3.5	5.7	5.6	2.5	66.3	66.5	74.5
	LSUN	7.6	7.2	3.7	6.0	5.9	2.6	65.9	66.0	74.5
	ImageNet	7.9	7.8	3.7	6.4	6.3	2.7	65.2	65.5	73.0
	Mean	8.4	8.3	**4.2**	6.6	6.5	**3.0**	65.5	65.7	**73.0**

Table 10: Full calibration results for the softmax temperature tuning baseline, the same baseline after adding 0-1 Posterior Rescaling, and temperature tuning + 0-1 Rescaling + OE. All results are percentages.