Self-Supervised Learning for Generalizable Out-of-Distribution Detection

Sina Mohseni,[1,2] **Mandar Pitale,**[2] **JBS Yadawa,**[2] **Zhangyang Wang**[1]
[1]Texas A&M University, College Station, TX
[2]NVIDIA, Santa Clara, CA
{sina.mohseni, atlaswang}@tamu.edu, {mpitale, jyadawa}@nvidia.com

Abstract

The real-world deployment of Deep Neural Networks (DNNs) in safety-critical applications such as autonomous vehicles needs to address a variety of DNNs' vulnerabilities, one of which being detecting and rejecting out-of-distribution outliers that might result in unpredictable fatal errors. We propose a new technique relying on self-supervision for generalizable out-of-distribution (OOD) feature learning and rejecting those samples at the inference time. Our technique does not need to pre-know the distribution of targeted OOD samples and incur no extra overheads compared to other methods. We perform multiple image classification experiments and observe our technique to perform favorably against state-of-the-art OOD detection methods. Interestingly, we witness that our method also reduces in-distribution classification risk via rejecting samples near the boundaries of the training set distribution.

Introduction

The real-world deployment of Deep Neural Networks (DNNs) in safety-critical applications, such as autonomous vehicles, calls for improving resiliency of DNNs for variety of vulnerabilities in these algorithms. Improving algorithm robustness for real-world scenarios calls for multi-fold efforts in network architecture design (Wang et al. 2017) and post-evaluation (Hendrycks and Dietterich 2019). There has recently been increasing attention to real-world challenge of out-of-distribution (OOD) sample errors. By quantifying model or data uncertainty and rejecting predictions of high uncertainty during inference (Kendall and Gal 2017), one can improve dependability of (already trained) probabilistic models in open-world scenarios. Current research on out-of-distribution detection are taking different directions, including detection based on model confidence (Liang, Li, and Srikant 2017), employing ensemble techniques (Vyas et al. 2018), learning DNN features (Lee et al. 2018) or using reconstruction scores (Pidhorskyi, Almohsen, and Doretto 2018) and recently self-supervised algorithms (Golan and El-Yaniv 2018; Hendrycks et al. 2019).

Our paper proposes a new technique to improve model reliability by adding OOD detector functions (with minimal architectural changes) to the model, to discriminate OOD samples in multiple reject classes without sacrificing the normal (i.e., in-distribution) classification performance. Our high level idea is to simultaneously train in-distribution classifiers and out-of-distribution detectors in one network. Specifically, we use additional nodes as reject functions in the last layer of our neural network. We use a self-supervised approach to train reject functions with free unlabeled OOD samples and the classifier functions with a labeled in-distribution training set.

We demonstrate the effectiveness of the proposed method through extensive comparisons with state-of-the-art techniques, across different datasets. We show that:

- Our method learns to generalize nicely on unseen OOD distributions. In particular, learning such generalizable OOD features is important for the detection robustness when a mixed of unseen distributions are present.
- Different from existing methods (Liang, Li, and Srikant 2017; Lee et al. 2018), our method does not need tuning with a sub-sample of the targeted OOD set, and therefore can use any "free" unlabeled OOD set for training.
- Our method can also benefit in-distribution classification accuracy, via rejecting ambiguous samples near the boundaries of the training set distribution.

Related Work

Earlier work in deep learning presents solutions such as deep ensembles (Lakshminarayanan, Pritzel, and Blundell 2017) and uncertainty estimation (Gal and Ghahramani 2016) to improve the dependability of machine learning solutions in real-world tasks. Despite their effectiveness, they carry significant extra computation and latency costs. (Geifman and El-Yaniv 2017) presents a simpler selective classification approach, and SelectiveNet (Geifman and El-Yaniv 2019) further proposes threshold on model prediction probability and selectively classify inputs below the desired classification risk. They show selective classification can improve model prediction's reliability by allowing the user to set a desired misclassification error-risk in trade-off with the test

Algorithm 1 Two-Step Training for In- and Out-of-Distribution Training Sets

procedure SUPERVISED IN-DISTRIBUTION LEARNING
Input: Batch of D^{in}_{train} samples in c different classes.
Training the in-distribution set by solving: min $(\mathbb{E}_{P_{in(\hat{x},\hat{y})}}[-\log(P_\theta(y=\hat{y}|\hat{x}))])$

procedure SELF-SUPERVISED OUT-OF-DISTRIBUTION LEARNING
Input: Batch of mixed D^{in}_{train} labeled samples and D^{out}_{train} unlabeled samples, set of OOD classes k,
learning coefficient for OOD features λ.
Training the mixed set by solving: min $(\mathbb{E}_{P_{in(\hat{x},\hat{y})}}[-\log(P_\theta(y=\hat{y}|\hat{x}))] + \lambda\,\mathbb{E}_{P_{out(x,target)}}[-\log(P_\theta(y=target|x))])$
Where pseudo-label $target$ for each OOD training sample is calculated at each pass during training:
if $argmax(P_\theta(x)) \in k$ **then**
$target \leftarrow argmax(P_\theta(x))$ ▷ choosing the reject class with maximum class probability.
else
$target \leftarrow random(k)$ ▷ choosing a random reject-label.

coverage. Along the same line, (Guo et al. 2017) presented temperature scaling a post-processing calibration technique to adjust the model probability estimates being off due to over fitting. However, this line of research does not emphasize model robustness against misclassifying OOD outliers. In comparison, this paper presents experiments on how our OOD rejection technique improves classification risk when both in-distribution and OOD samples are present.

To investigate the use of class probabilities as a proper measure for OOD detection, (Hendrycks and Gimpel 2016) presents maximum softmax probability (MSP) as the Baseline for OOD detection in DNN algorithms. Later, (Liang, Li, and Srikant 2017) presents ODIN to calibrate pre-trained models using temperature scaling and small perturbation on in-distribution inputs to improve model robustness against OOD samples. In a more sophisticated approach, (Lee et al. 2017) used a generative adversarial network (Goodfellow et al. 2014) to synthesis samples which are out of but close to the training data distribution boundaries for calibrating the model. They employ a two term loss function to force the predictive distribution of OOD samples toward uniform distribution. Along the same line, (Hendrycks, Mazeika, and Dietterich 2018) investigates the effectiveness of large natural datasets disjoint from the training set to calibrate the model prediction. However, it is an inherent problem with ReLU family activation functions that they produce arbitrary high confidence as inputs get further from the training distribution (Hein, Andriushchenko, and Bitterwolf 2019). Therefore, in contrast to model calibration techniques, we show using additional decision boundaries in the network has a better effect on discriminative feature learning.

Another line of research focuses on unsupervised and self-supervised learning for OOD detection challenge, by estimating a novelty score or training a one-class classifier. For instance, using generative models for novelty detection has been investigated in (Nalisnick et al. 2018). (Pidhorskyi, Almohsen, and Doretto 2018) examines the use of reconstruction error together with probability distribution of the full model in an autoencoder as a novelty measure, and improve OOD detection by incorporating the Mahalanobis distance in the latent space. Recently, (Golan and El-Yaniv 2018) studies self-supervised geometric transformations learners to distinguish normal and outlier samples in a one-vs-all fashion. In a concurrent paper, Hendrycks et al. (Hendrycks et al. 2019) presents experiments on combining different self-supervised geometric translation prediction tasks in one model, using multiple auxiliary heads. Their results show improvements in detecting OOD samples as well as improvements in model robustness against common input corruptions and adversarial examples. Different from their work, this paper proposes using one auxiliary head of self-supervised OOD detection head, to learn generalizable OOD features in addition to learning the normal multi-class classification task.

Self-Supervised OOD Feature Learning

The problem we consider in this paper is to detect OOD outliers (D^{out}) using the same classifier $P_\theta(y|x)$ trained on normal distribution (D^{in}). In order to do so, we add an auxiliary head to the network and train in for the OOD detection task. Therefore, in contrast to softmax calibration methods (Lee et al. 2017; Hendrycks, Mazeika, and Dietterich 2018), we embed OOD discriminators in the model along with in-distribution classifiers. We first use a full-supervised training to learn D^{in}_{train} for the main classification head and then a self-supervised training with OOD training set (D^{out}_{train}) for the auxiliary head. Our method can use any disjoint free unlabeled D^{out}_{train} for learning generalizable OOD features; hence unlike previous methods (Liang, Li, and Srikant 2017; Lee et al. 2018), it requires no validation sub-samples from the target OOD set for tuning.

Despite its conceptual simplicity, later via thorough experiments, we will show our method to compare highly favorably against other state-of-the-arts, in terms of both OOD detection *performance* and *generalizability*.

Network Architecture and Training

Our method imposes the minimal change in the model architecture and can be applied on top of any DNN classifier. Precisely, we add additional nodes (set of reject classes k) in the last layer of the network — which we call OOD detectors — to learn the OOD features in a self-supervised manner. We employ a two-step training that starts with the full-supervised in-distribution feature learning and then continues with self-supervised OOD feature learning. Algorithm

1 describes the two step training procedure. Below we explain the algorithm routine; specific architecture details and training protocol are presented in the next section.

Training starts with a full-supervised in-distribution features learning that can be done in any fashion and duration to reach the optimum/desired classification performance. The training data (D^{in}_{train}) in this step comes with labels that are used for loss minimization. We used cross entropy loss for the supervised training step.

After learning the in-distribution features, in the second step, we mix each mini-batch with both samples from D^{in}_{train} and D^{out}_{train}, which is an auxiliary unlabeled training set, to train the auxiliary head for OOD features. We use a two term loss function for two (in and out features) learning tasks (λ is a coefficient):

$$\mathcal{L}_{total} = \mathcal{L}_{in} + \lambda \mathcal{L}_{out}$$

The model is also able to self-label the unlabeled D^{out}_{train} samples, with new $target$ predictions at each training pass. Similar to the full-supervised step, we also use cross entropy loss for D^{out}_{train} training:

$$\mathcal{L}_{out} = -\log(P_\theta(y = target|x))$$

in which the $target$ pseudo-labels are generated using a simple semi-random method (see Algorithm 1) during the training process. Specifically, the model uses its own prediction of D^{out}_{train} samples at each forward pass to generate labels for D^{out}_{train} samples. If the prediction was a false negative, then it randomly assigns one of the reject class labels to the sample. This is similar to Caron et al. (Caron et al. 2018) where pseudo-labels are generated using a k-means algorithm to train an unsupervised deep clustering network. Throughout the OOD features learning step, we keep some in-distribution samples in each mini-batch so that the model does not forget learned in-distribution features and causing in-distribution generalization error.

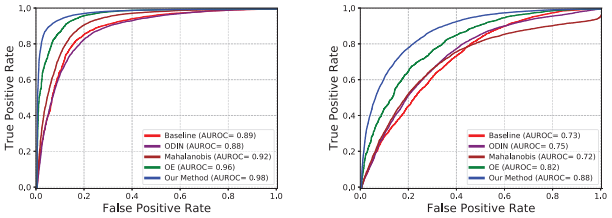
Detection Inference

During inference, we only use one softmax function for all output classes. We take the sum of softmax output of the OOD classes as the OOD-detection signal. Naturally, we take the maximum softmax output of the main classes as the classifier prediction output. We then evaluate the OOD detection performance with only unseen OOD test sets (D^{out}_{test}) and the normal test set (D^{in}_{test}) for each trained model. Therefore, unlike (Hendrycks et al. 2019) and (Pidhorskyi, Almohsen, and Doretto 2018) where the trained model only performs novelty detection; our model unifies both multi-class classification and OOD detection in one model.

Evaluation Experiments

In this section we present a set of experiments on our technique to evaluate the model performance for both OOD detection and normal classification.

Training and Test Sets To provide adequate evaluation results for our technique we trained and evaluated multiple multi-class classifiers on different training sets. Notice that in all experiments we used different OOD train and test sets since our assumption is that we do not have access to outliers in real-world cases. For example, in the MNIST (LeCun et al. 1998) experiment, while the normal D^{in}_{train} is handwritten digits, we used English letters from E-MNIST (Cohen et al. 2017) as the source of D^{out}_{train} set. We then evaluate the OOD detection performance with unseen D^{out}_{test} including Kuzushiji-MNIST (Clanuwat et al. 2018), not-MNIST (Bulatov 2011), and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017) datasets to measure how well can the model generalize on unseen distributions.



(a) CIFAR-10 D^{in} (b) CIFAR-100 D^{in}

Figure 1: Comparison between different OOD detection methods when D^{out}_{test} is mix of five different and disjoint outlier datasets. Detectors without generalized OOD feature learning (i.e., ODIN and Mahanalobis) show significant performance drop when facing mix of outlier distributions.

Other experiments include training multi-class classifiers on CIFAR-10, CIFAR-100 (Krizhevsky, Hinton, and others 2009), and SVHN (Netzer et al. 2011) datasets. In all experiments (except the MNIST) we used 80 Million Tiny Images dataset (Torralba, Fergus, and Freeman 2008) as the source of unlabeled D^{out}_{train}. We discuss our choice of other natural (e.g., ImageNet dataset (Russakovsky et al. 2015)) and synthesized datasets as D^{out}_{train} in the discussion section. We tested each trained model with test sets of five unseen disjoint datasets including Texture (Cimpoi et al. 2014), Places365 (Zhou et al. 2017), and LSUN (Yu et al. 2015) datasets as D^{out}_{test}. For all test experiments, we used the test set or equal mix of test sets of aforementioned outlier datasets as the D^{out}_{test}. For both CIFAR experiments, we removed mutual samples from 80 Million Tiny Images to create a disjoint D^{out}_{train}.

Network Architecture and Training For all experiments on CIFAR-10 and CIFAR-100 datasets we used 40-2 Wide Residual Network architecture (Zagoruyko and Komodakis 2016). We used a smaller 16-2 Wide ResNet for the SVHN and a vanilla convolutional neural network with two convolution layers for the MNIST dataset. We used five reject classes for the CIFAR-10, MNIST, and SVHN experiments and 10 reject classes for the CIFAR-100 experiment. Similar to the conventional practice of clustering algorithms, we perform cross-validation to test different numbers of reject classes to reach the best detection performance. We will discuss the effect of reject class numbers in the later section.

The training starts with full-supervised training of the multi-class classifier the D^{in}_{train} (see Algorithm 1). We trained the model for 100 epochs in CIFAR-10 and CIFAR-

D^{in}_{train}	D^{out}_{test}	FPR at TPR 0.95%			AUROC			AUPR		
		Baseline	OE	Our method	Baseline	OE	Our method	Baseline	OE	Our method
MNIST	not-MNIST	17.11	0.25	**0**	95.98	99.86	**99.99**	95.75	99.86	**99.99**
	F-MNIST	2.96	0.99	**0**	99.30	99.83	**100**	99.19	99.83	**100**
	K-MNIST	10.54	**0.03**	0.35	97.11	97.60	**99.91**	96.46	97.05	**99.91**
SVHN	Texture	4.70	**1.04**	2.28	98.40	**99.75**	99.37	93.07	**99.09**	98.16
	Places365	2.55	**0.02**	0.05	99.27	**99.99**	99.94	99.10	**99.99**	99.93
	LSUN	2.75	0.05	**0.04**	99.18	**99.98**	99.94	97.57	**99.95**	99.85
	CIFAR-10	5.88	3.11	**0.31**	98.04	99.26	**99.83**	94.91	97.88	**99.60**
	CIFAR-100	7.74	4.01	**0.07**	97.48	99.00	**99.93**	93.92	97.19	**99.81**
CIFAR-10	SVHN	28.49	8.41	**3.62**	90.05	98.20	**99.18**	60.27	97.97	**99.13**
	Texture	43.27	14.9	**3.07**	88.42	96.7	**99.19**	78.65	94.39	**98.78**
	Places365	44.78	19.07	**10.86**	88.23	95.41	**97.57**	86.33	95.32	**97.77**
	LSUN	38.31	15.20	**4.27**	89.11	96.43	**98.92**	86.61	96.01	**98.74**
	CIFAR-100	43.12	**26.59**	30.07	87.83	92.93	**93.83**	85.21	92.13	**94.23**
CIFAR-100	SVHN	69.33	52.61	**18.22**	71.33	82.86	**95.82**	67.81	80.21	**95.03**
	Texture	71.83	55.97	**40.3**	73.59	84.23	**89.76**	57.41	75.76	**83.55**
	Places365	70.26	57.77	**39.96**	73.97	82.65	**89.08**	70.46	81.47	**88.00**
	LSUN	73.92	63.56	**41.24**	70.64	79.51	**88.88**	66.35	77.85	**87.59**
	CIFAR-10	65.12	59.96	**57.79**	75.33	77.53	**77.70**	71.29	**72.82**	72.31

Table 1: Out-of-distribution detection results (%) on various D^{in}_{train} and D^{out}_{test} experiments. We compare our method with the Baseline (Hendrycks and Gimpel 2016) and OE (Hendrycks, Mazeika, and Dietterich 2018) techniques. All results are averaged over 10 runs. The D^{out}_{train} is E-MNIST for the MNIST experiment and Tiny Images dataset for all other experiments.

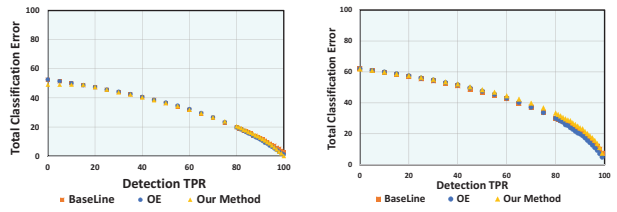
100 experiments, 20 epochs for the SVHN training set, and 10 epochs for the MNIST experiment. We used batch size of 128, learning rate of 0.1 (decayed on a cosine learning rate schedule), and dropout rate of 0.3 for the CIFAR-10, CIFAR-100, and SVHN experiments. For the MNIST experiment, we used batch size of 64, learning rate of 0.01 (decayed on a cosine learning rate schedule), and dropout rate of 0.1. We measured the normal test set error rate for each trained model as follows: 4.72% on CIFAR-10, 23.74% on CIFAR-100, 4.94% on SVHN, and 1.33% on MNIST.

After the model learned in-distribution features, we then continued with the self-supervised OOD feature learning with unlabeled D^{out}_{train} dataset for more epochs. For the self-supervised step, we mixed each mini-batch with both D^{in}_{train} and D^{out}_{train} to maintain features diversity and prevent the model from forgetting normal features when learning new OOD features. In all experiments we used five times larger D^{out}_{train} mini-batches compared to D^{in}_{train} mini-batches. Also, we used a fix $\lambda = 5$ for OOD feature learning coefficient (see Algorithm 1) in all experiments and did not need to tune at each run. We continued the self-supervised training step for 10 epochs in MNIST, 20 epochs in SVHN, and 100 epochs for CIFAR-10 and CIFAR-100 experiments.

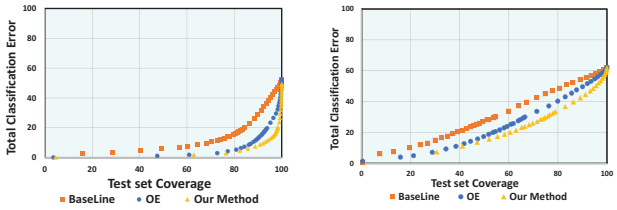
OOD Detection Performance

We used different metrics to measure the OOD detection performance in our experiments. Our threshold independent metrics are Area Under Receiver Operating Characteristic curve (AUROC) (Davis and Goadrich 2006) and Area Under Precision and Recall curve (AUPR). The ROC curve shows the relation between True Positive Rate (TPR) and False Positive Rate (FPR) in detection. The AUROC will be 100% for a perfect detector and 50% for a random detector. We used D^{out}_{test} (test set of outlier datasets) as positive OOD samples and the D^{in}_{test} (test set of normal dataset) as negative samples for detection. Therefore, we calculate FPR as the probability of negative samples being misdetected as positive and TPR as the probability of correctly detecting positive samples. For the main experiments, we calculated the detector's FPR when the detector threshold is set on 95% TPR. We also used Precision-Recall (PR) curve that shows the relation between detector positive predictive value (precision) and TPR (recall) at different thresholds.

Table 1 presents our detailed evaluation and comparison results with two confidence-score based methods including the Baseline (Hendrycks and Gimpel 2016) and Outlier Exposure (OE) technique presented in (Hendrycks, Mazeika, and Dietterich 2018). The choice of comparison with OE was because of the fact that similar our method, they also focus on OOD feature learning together with normal distribution features. To evaluate the robustness of our method, we train multi-class classifiers on four different training sets (CIFAR-10, CIFAR-100, MNIST, SVHN) and test each of them (except MNIST) on five different disjoint D^{out}_{test}. For all test experiments, we used equal number of samples from D^{in}_{test} and D^{out}_{test} sets. All detection results are from one trained model but averaged over 10 runs. We compare our method's detection performance with OE by averaging measured AUROC over the five D^{out}_{test} sets in each experiment. Our method outperforms the predecessor technique on all tests by: 6.89% gain in CIFAR-100 experiment, 1.80% gain in CIFAR-10 experiment, 0.21% gain in SVHN experiment,

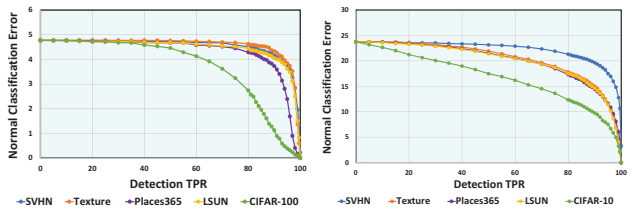


(a) CIFAR-10 D^{in} (b) CIFAR-100 D^{in}

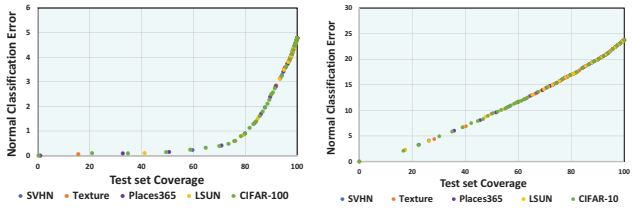


(c) CIFAR-10 D^{in} (d) CIFAR-100 D^{in}

Figure 2: Comparison between the Baseline, OE, and our method's total misclassificaiton error rate **(Top)** for CIFAR-10 and CIFAR-100 experiments at different OOD detection TPR. D^{out}_{test} set is an equal mix of five different test sets. Our method shows the highest test set coverage **(Bottom)** at all classification error rate.



(a) CIFAR-10 D^{in} (b) CIFAR-100 D^{in}



(c) CIFAR-10 D^{in} (d) CIFAR-100 D^{in}

Figure 3: Normal classification error rate **(Top)** and risk-coverage curves **(Bottom)** for CIFAR-10 and CIFAR-100 experiments. Our method consistently improves classification error rate as we reduce the detection threshold for higher detection true positive rates. Colored lines show different D^{out}_{test} sets.

and 1.29% gain in MNIST experiment. Note that none of compared OOD techniques in this section used sub-sample of the targeted D^{out}_{test} set for model tuning.

Mixed-Distribution OOD Detection To evaluate the generalizability of our technique, we simulating a real-world scenario where both samples from normal distribution and outliers from multiple unknown distribution exists. Therefore, we run experiments which D^{out}_{test} is a mix of different disjoint datasets. We create an equal mix of SVHN, Texture, Places365, LSUN, and CIFAR-100 (or CIFAR-10 for the CIFAR-100 experiment) test sets for a more diverse and challenging D^{out}_{test}. We randomly take 2000 samples from the test sets of each dataset to create the new D^{out}_{test} set. We first evaluated the Baseline, OE, and our method with the new mixed-distribution D^{out}_{test}, and observed a slight (less than 0.5%) AUROC drop in OOD detection for these three methods. However, comparison with ODIN (Liang, Li, and Srikant 2017) and Mahanalobis (Lee et al. 2018) detectors in Figure 1 shows a significant detection performance drop of these methods when facing mix of different D^{out}_{test} sets. For the case of ODIN detector, the AUROC drops 14.07% for CIFAR-100 (and 7.41% for CIFAR-10) experiment when facing the mixed distribution D^{out}_{test} set. Similarly, for the case of Mahanalobis detector, the AUROC drops 25.84% for CIFAR-100 (and 7.34% for CIFAR-10) experiment when facing the mixed distribution D^{out}_{test} set. This detection performance drop indicates high reliance of these two methods on tuning on the known outlier distribution rather than learning generalizable OOD features. For both ODIN and Mahanalobis detectors, we used 200 samples from each of five D^{out}_{test} sets to tune Mahanalobis and ODIN detectors in this experiment. We review more detailed valuation and comparison with the two methods in the discussion section.

We next evaluate the total misclassification rate of our model and compare it with the Baseline and OE detectors. We calculate total misclassification rate as the number of misclassified inputs (normal classification error plus false negative samples) divided by total number of classified inputs (total number of negative samples). Figure 2 (a) and (b) show total misclassification error rate at different OOD detection TPR when feeding the model with both D^{in}_{test} and the mixed D^{out}_{test} set. All three techniques show total misclassification rate reduction with higher detection TPR in CIFAR-10 and CIFAR-100 experiments. Figure 2 (c) and (d) presents a comparison of risk-coverage curves between these techniques which indicates our technique has the highest D^{in}_{test} set coverage in this comparison. For the CIFAR-10 experiments, Our method shows 4.57% higher D^{in}_{test} set coverage compared to the OE and 28.68% higher compared to the Baseline method when our detector is set on 95% TPR. Likewise, for the CIFAR-100 experiments, Our method shows 6.45% higher D^{in}_{test} set coverage compared to the OE and 21.08% higher compared to the Baseline method when our detector is set on 95% TPR.

Normal Classification Performance

Despite most OOD detection papers which restrict the evaluations experiments to only the D^{out}_{test}, the presence of an OOD detector affects normal classification performance as well. Specifically, the false negative detection samples always increase the classification error rate, but false positive detection samples could decrease the classification error. Misdetected samples from D^{in}_{test} (false positive samples) near the training distribution boundary (regions with low density) could reduce classification error. We evaluate how does our OOD detector affect normal classification risk

and coverage in the concept of selective classification (Geifman and El-Yaniv 2017). To measure the normal error rate at different desired OOD detection TPR, we first calculate the detection threshold using equal size of normal and outlier samples. We then feed the D^{in}_{test} to the network which selectively classifies samples that are not detected as OOD.

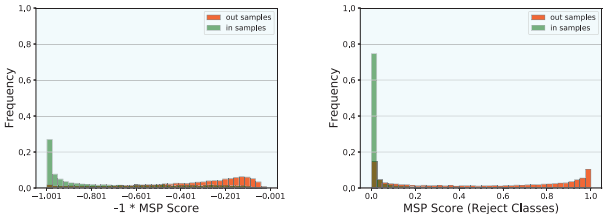
Figure 3 (a) and (b) show normal misclassification error rate at different OOD detection TPRs on CIFAR-10 and CIFAR-100 experiments. The normal misclassification error drops consistently as we reduce the detection threshold for higher TPR. Note that in all tests the normal misclassification rate is 4.72% for the CIFAR-10 dataset and 23.74% for CIFAR-100 without using the OOD detector. Experiments on CIFAR-10 show the normal misclassification error rate is reduced by 1.92% on average when the detector is set on 95% TPR detection. Similar to that, averaging on experiments for CIFAR-100 dataset show the normal misclassification error rate is fallen by 11.81% when the detector is set on 95% TPR detection. However, this surge in the classification performance is in a trade-off with the D^{in}_{test} coverage which is due to higher detection FPR: see Figure 3 (c) and (d) for D^{in}_{test} risk-coverage curve in different experiments.

Discussion and Analysis

In this section we discuss the robustness of our technique by reviewing how different hyperparameters and training variations affect the OOD detection performance. We compare the OOD detection performance and generalization with ODIN (Liang, Li, and Srikant 2017), Mahanalobis detector (Lee et al. 2018), and Deep SVDD (Ruff et al. 2018) one class classifier.

Generalizable OOD Feature Learning In our experiments, we found size and diversity of the D^{out}_{train} set are important factors to learn generalizable OOD features. Since our normal D^{in}_{train} datasets (CIFAR-10, CIFAR-100, and SVHN) are much smaller than D^{out}_{train} (Tiny Images dataset with 80 million samples), we used five times more OOD samples in each training iteration to create large enough mini-batches for the self-supervised OOD training step. Note that the model learns the in-distribution features in the full-supervised training step and the self-supervised training goal is learning D^{out}_{train} features to generalize well on unseen D^{out}_{test}. To even further enhance the OOD feature learning, we perform random sampling without replacement from our large D^{out}_{train} to increase the diversity of OOD training features. This sampling resulted in using about 30% of the Tiny Images dataset throughout the 100 epochs of self-supervised training for CIFAR-10 and CIFAR-100 experiments. Due to the simpler features in SVHN dataset (compared to CIFAR), we observed that using only about 5% of the entire D^{out}_{train} is enough for the SVHN experiment during the 20 epochs of self-supervised learning.

An important factor for OOD detectors is to generalize on any unseen D^{out}_{test} independent from the training and tuning data. We performed rigorous evaluations with different individual and mixed disjoint image datasets to convey the importance of generalization in OOD detection. On the other hand, techniques like ODIN (Liang, Li, and Srikant 2017), Lee et al. (Lee et al. 2017), and Mahanalobis detector (Lee et al. 2018) heavily rely on a small sub-sample of targeted D^{out}_{test} for parameter tuning. For example, the Mahanalobis detector shows an average of 13.4% AUROC drop for CIFAR-100 classifier (9.87% AUROC drop for ODIN) and 1.4% drop for CIFAR-10 classifier (3.6% AUROC drop for ODIN) when using adversarial perturbation samples for parameter tuning instead of taking 1000 samples from the D^{out}_{test} set. Even in the case of mixed D^{out}_{test} set, figure 1 shows significant detection performance drop for these methods. The difference between ROC curves shows how well different methods can generalize on mixed of unseen D^{out}_{test} sets.



(a) OE (Confidence score)

(b) Our Method (Multiple reject classes)

Figure 4: Comparison of OOD detection score histograms between OE ($-1 \times MSP$ score) and our method (MSP^{Out} score) for the CIFAR-10 experiment.

Synthesized OOD Training Set Early in our experiments, we found out the closeness of D^{out}_{train} and D^{in}_{train} is important for learning features which are outside but near the training distribution. In our CIFAR-10 and CIFAR-100 experiments, D^{in}_{train} are disjoint subsets of D^{out}_{train} (Tiny Images dataset), and hence in the self-supervised training step, OOD samples fall somewhat near (yet non-identical to) D^{in}_{train} in the feature space, as analyzed in (Recht et al. 2018).

To test the flexibility in choosing other D^{out}_{train}, we also used down-sampled ImageNet-22k (with ImageNet-1k removed from it) dataset as another choice of large scale natural images and repeated the CIFAR-10 and CIFAR-100 experiments. However, we saw an average detection AUROC drop from 88.24% to 84.99% on CIFAR-100 and from 97.37% to 90.40% on CIFAR-10 experiment. To improve the ImageNet as the D^{out}_{train} set, we simply blended OOD samples with the D^{in}_{train} to create new synthesized OOD training set. The new OOD training set (with $\alpha = 0.1$ image blending) improve averaged detection AUROC to 85.98% on CIFAR-100 and to 91.72% on CIFAR-10 experiment. Our conclusion is that a suitable unlabeled D^{out}_{train} could be provided by a mother dataset (like the Tiny Images for the cases of CIFAR-10 and CIFAR-100), or simply collected during the normal training data collection, and improved with different augmentation and synthesizing (as in (Liang, Li, and Srikant 2017)) techniques.

In the case of SVHN and MNIST training sets, we observed that the network was able to easily distinguish in-distribution features from the OOD features. For the SVHN experiment, we observed no reduction or improvement in

Detection Score	D^{in}_{train}	
	CIFAR-10	CIFAR-100
$max\,(softmax^{out})$	97.77	88.31
$sum\,(softmax^{out})$	97.83	88.20
$weighted\,(softmax^{out})$	97.75	88.32
$entropy^{out}$	97.88	87.93
$entropy^{out} - entropy^{in}$	97.86	87.72
Weighted All Scores	97.85	88.44

Table 2: OOD detection AUROC results (%) when using various detection scores. Using different scores does make make significant improvement in detection performance.

OOD detection performance when using ImageNet as the sources of D^{out}_{train} compared to the choice of Tiny Images dataset. Likewise for the MNIST experiment, the OOD detection performance when using E-MNIST as the sources of D^{out}_{train} (average AUROC = 99.65%) was not much different from using K-MNIST as the sources of D^{out}_{train} (average AUROC = 99.97%).

Multiple Reject Classes Similar to the conventional practice of unsupervised clustering techniques, we test our technique with using different number of reject classes for OOD distribution. We vary the number of reject classes, denoted as *k*, in CIFAR-10 and CIFAR-100 experiments, train for equal number of epochs, and measure the OOD detection performance. We tried $k = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40,$ and 50 number of reject classes for the CIFAR-10 experiment and $k = 1, 2, 5, 10, 20, 30, 50,$ and 100 number of reject classes for the CIFAR-100 experiment.

Comparing the OOD detection AUROC results shows an average of 91.1% with a standard deviation of 0.31% when using the different k-reject classes for the CIFAR-10 data set. Likewise, for the CIFAR-100 dataset, OOD detection AUROC results show an average of 79.2% with a standard deviation of 5.40% when using the different k-reject classes for the CIFAR-100 data set. We chose *k=5* for all 10-class classifiers and *k=10* for the 100-class classifier experiments. Our results indicate that the optimal number of reject classes, which results in neither over- nor under-partition of OOD features, would be dependent on the choice of in and out of distribution training data. However, its impact on OOD detection performance is mild and insensitive. Figure 4 shows a comparison of the histogram of OOD detector scores between OE and our method.

OOD Detection Scores We also considered using different OOD scoring methods rather than taking maximum softmax probability (MSP) of reject classes as the detection signal. During the self-supervised training step, our random pseudo-labeling clusters OOD features into multiple reject classes which is advantageous compared to 1-reject class. For this purpose, we examined weighting softmax probability of reject classes, the entropy of softmax vector for both reject classes ($Entropy^{Out}$) and normal classes ($-1 \times Entropy^{In}$), and combination of weighted softmax probabilities and entropy of softmax vector.

Table 2 shows a list of various detection scores that we examined as OOD detection score. We used a mix of five difference disjoint datasets as the D^{out}_{test} set and the results show AUROC of the OOD detection in CIFAR-10 and CIFAR-100 experiments. With a non-weighted sum of softmax probabilities of reject classes we observed an AUROC increase of 0.06% in CIFAR-10 experiment over maximum softmax probability detection. Using a greedy search we weighted the softmax scores of reject classes and observed 0.01% increase in CIFAR-100 experiment. Adopting entropy of the softmax vector (from reject classes) also resulted in 0.11% increase in CIFAR-10 experiment. Lastly, we examined combining the weighted softmax scores and softmax vector entropy for higher AUROC. Our conclusion is that training multiple reject classes for OOD detection improves the detection performance via allowing better OOD detection scores compared to using only 1-reject class.

Comparison to One Class Classification We do not primarily compare our method to one class classifiers and other families of unsupervised outlier detectors/ uncertainty estimators, due to their often significantly higher inference latency and memory overheads. However, we briefly compared our method with the Deep SVDD (Ruff et al. 2018) one class classifier on CIFAR-10 dataset. We train 10 different classifiers that each takes one of the CIFAR-10 classes as the D^{in}_{train} and the other 9-classes as D^{out}_{test}. Similar to other CIFAR experiments, we use Tiny Images as the D^{out}_{train}. Our experimental results show average AUROC of 77.75% for 10 trained one class classifiers which outperforms Deep SVDD method with averaged AUROC of 64.81% with the same train and test sets.

Conclusion and Future Work

We presented a new method to detect OOD samples with a minimal twist in a regular multi-class DNN classifier. In a two step training, our model jointly learns generalizable outlier features as well as in-distribution features for normal classification. Our evaluation results show the proposed self-supervised learning of OOD features can very well generalize to reject other unseen distribution. Also, our method reduces the classification risk for the test sets while by rejecting ambiguous samples near the boundaries of training distribution. The immediate future directions for our technique is using a clustering method to assign pseudo-labels (instead of random pseudo-labels) to OOD samples. Including temperature scaling as another tuning step is also worthy to explore for better calibrating the model.

References

Bulatov, Y. 2011. Notmnist dataset.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *ECCV*, 132–149.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; ; and Vedaldi, A. 2014. Describing textures in the wild. In *CVPR*.

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

Cohen, G.; Afshar, S.; Tapson, J.; and van Schaik, A. 2017. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*.

Davis, J., and Goadrich, M. 2006. The relationship between precision-recall and roc curves. In *ICML*, 233–240. ACM.

Gal, Y., and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 1050–1059.

Geifman, Y., and El-Yaniv, R. 2017. Selective classification for deep neural networks. In *NIPS*, 4878–4887.

Geifman, Y., and El-Yaniv, R. 2019. Selectivenet: A deep neural network with an integrated reject option. *arXiv preprint arXiv:1901.09192*.

Golan, I., and El-Yaniv, R. 2018. Deep anomaly detection using geometric transformations. In *NIPS*, 9758–9769.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *ICML*.

Hein, M.; Andriushchenko, M.; and Bitterwolf, J. 2019. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *CVPR*, 41–50.

Hendrycks, D., and Dietterich, T. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Hendrycks, D., and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hendrycks, D.; Mazeika, M.; Kadavath, S.; and Song, D. 2019. Using self-supervised learning can improve model robustness and uncertainty. *CoRR* abs/1906.12340.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. G. 2018. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*.

Kendall, A., and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 5574–5584.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 6402–6413.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Lee, K.; Lee, H.; Lee, K.; and Shin, J. 2017. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NIPS*, 7167–7177.

Liang, S.; Li, Y.; and Srikant, R. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Nalisnick, E.; Matsukawa, A.; Teh, Y. W.; Gorur, D.; and Lakshminarayanan, B. 2018. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Pidhorskyi, S.; Almohsen, R.; and Doretto, G. 2018. Generative probabilistic novelty detection with adversarial autoencoders. In *NIPS*, 6822–6833.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2018. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*.

Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S. A.; Binder, A.; Müller, E.; and Kloft, M. 2018. Deep one-class classification. In *ICML*, 4393–4402.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *IJCV* 115(3):211–252.

Torralba, A.; Fergus, R.; and Freeman, W. T. 2008. 80 million tiny images: A large data set for nonparametric object and scene recognition. *TPAMI* 30(11):1958–1970.

Vyas, A.; Jammalamadaka, N.; Zhu, X.; Das, D.; Kaul, B.; and Willke, T. L. 2018. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *ECCV*, 550–564.

Wang, X.; Luo, Y.; Crankshaw, D.; Tumanov, A.; Yu, F.; and Gonzalez, J. E. 2017. Idk cascades: Fast deep learning by learning not to overthink. *arXiv preprint arXiv:1706.00885*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Zagoruyko, S., and Komodakis, N. 2016. Wide residual networks. *BMVC*.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *TPAMI* 40(6):1452–1464.