GAN Based Boundary Aware Classifier for Detecting Out-of-distribution Samples

Sen Pei1, Xin Zhang1, Richard Yi Da Xu2 and Gaofeng Meng1

1NLPR, Institute of Automation, Chinese Academy of Sciences
2Department of Mathematics, Hong Kong Baptist University

peisen2020@ia.ac.cn, {xin.zhang2018, gfmeng}@nlpr.ia.ac.cn, xuyida@hkbu.edu.hk

Abstract

This paper focuses on the problem of detecting out-of-distribution (ood) samples with neural nets. In image recognition tasks, the trained classifier often gives high confidence score for input images which are remote from the in-distribution (id) data, and this has greatly limited its application in real world. For alleviating this problem, we propose a GAN based boundary aware classifier (GBAC) for generating a closed hyperspace which only contains most id data. Our method is based on the fact that the traditional neural net separates the feature space as several unclosed regions which are not suitable for ood detection. With GBAC as an auxiliary module, the ood data distributed outside the closed hyperspace will be assigned with much lower score, allowing more effective ood detection while maintaining the classification performance. Moreover, we present a fast sampling method for generating hard ood representations which lie on the boundary of pre-mentioned closed hyperspace. Experiments taken on several datasets and neural net architectures promise the effectiveness of GBAC.

1 Introduction

Deep convolutional neural networks are one of the basic architectures of deep learning, and they have achieved great success in modern computer vision tasks. However, the over-confidence issue has always been with CNN which harms its generalization performance seriously. In previous research, neural networks have been proved to generalize well when the testing data are sampled from the same distribution as training data (i.e., id data), but in open world recognition, this prerequisite is hard to be guaranteed.

From the current situation, there are two major challenges need to be settled with an increasing need in recognition tasks: adversarial examples and ood examples. In [Goodfellow et al., 2014a], it is found that even adding very small perturbations to the input can fool a well trained classification net, and these modified inputs are the so called adversarial examples. Another problem is detecting ood examples which are drawn far away from the training data. The trained neural net often gives very high confidence to these ood samples which has raised concerns for AI Safety [Amodei et al., 2016] in many applications, and that’s the so called over-confidence issue [Nguyen et al., 2015].

Recently, several studies have proposed different approaches for detecting ood samples, and thus, improving the robustness of classification net. In [Hendrycks and Gimpel, 2016], a softmax based baseline is proposed for identifying ood samples, and the metrics are also defined properly in detail. Further, in ODIN [Liang et al., 2020], temperature scaling and input pre-processing are introduced for improving the confidence score of id samples. Most subsequent techniques can leverage ODIN since it is a post-processing method. In [Yang et al., 2018], convolutional prototype learning (CPL) is proposed for image classification which shows effectiveness in outlier rejection and class-incremental learning. Moreover, in [DeVries and Taylor, 2018], it points out that the outputs of softmax can not represent the confidence of neural net actually, and thus, a new branch is separated for confidence estimation independently. All these previous works have brought many different perspectives and inspirations for solving the open world recognition tasks, however, they pay limited attention to the learning of hard ood features which have great influence on ood detection.

In this paper, we attribute the reason of bad ood detec-
tion performance to the fact that the traditional classification net can not perceive the boundary of id data. And thus, this paper focuses on where and how to find these data that distribute at the boundary of id and ood, i.e., the hard ood data. The key idea of our proposed GBAC is to make the CNN have the ability to perceive the boundary of id samples. In Fig.1, a trained ResNet18 is used for extracting features from MNIST dataset, and the blue points indicate the location of these images in feature space. It can be clearly found that in feature space, each category of id data is concentrated in a certain area with the mapping of CNN, but almost all the feature space is assigned with high confidence score. Motivated by this abnormal phenomenon, we uniformly sample feature points in feature space and only those close to clusters are treated as positive samples for training a conditional boundary aware discriminator. With that, one can verify whether an input image belongs to id data or not. This work mainly has the following contributions:

- A GAN based boundary aware classifier is proposed for improving the rejection ability of neural networks while maintaining the classification performance. This framework never changes the weights of pre-trained model and can be combined with major supervised classification nets.
- An efficient method named RSM (Representation Sampling Method) for sampling hard ood feature representations is given. We argue that the hard ood examples distribute at the boundary between id and ood data, and these ood examples greatly influence the generalization ability of neural networks.
- We test the proposed method on several datasets with different neural net architectures, the results suggest that our proposed GBAC significantly improve the performance of ood detection, allowing more robust classification in open world.

2 Related Works

There have been many techniques proposed for detecting ood examples. To the best of our knowledge, the current approaches can be divided into four branches mainly which are softmax based methods, generative model based methods, classifiers based methods and the ensembles based methods.

**Softmax based methods** In [Hendrycks and Gimpel, 2016], a baseline approach to detect ood inputs named max-softmax is proposed, also in this work, the metrics of evaluating ood detectors is defined properly. The max-softmax method based on the observation that the predicted score assigned to id examples usually higher than ood examples. Following this, inspired by [Goodfellow et al., 2014a], ODIN [Liang et al., 2020] is proposed for improving the detection ability of max-softmax using temperature scaling and input preprocessing. ODIN can increasing the confidence score of id examples while decreasing that of ood examples. In [Abdelzad et al., 2019; Lee et al., 2018b], these studies argue that the feature maps from the penultimate layer of neural networks are not suitable for detecting ood examples, and thus, they use the features from the well-chosen layer and adopt some metrics such as Euclidean distance, Mahalanobis distance and OSVM classifier. In [DeVries and Taylor, 2018], a branch is separated out for confidence regression since the outputs of softmax can not well represent the confidence of neural networks.

**Generative model based methods** This kind of methods usually use the id samples to generate fake ood samples, and then, train a (C + 1) classifiers which can improve the rejection ability of neural nets. [Vernekar et al., 2019] treats the ood samples as two types, one indicates these samples that are close to but outside the id samples, and the other is these samples which lie on the id boundary. This work uses Variational AutoEncoder [Sohn et al., 2015] to generate such data for training. In [Lee et al., 2018a], the authors argue that samples lie on the boundary of id manifold can be treated as ood samples, and they use GAN [Goodfellow et al., 2014b] to generate these data. The proposed joint training method of confident classifier and adversarial generator inspire our work. It can not be ignored that the methods mentioned above are only suitable for small toy datasets, and the joint training method harms the classification performance of neural net to some extent. Further, in [Denouden et al., 2018], the study points out that AutoEncoder [Kingma and Welling, 2014] can reconstruct the id samples with much less error than ood examples, allowing more effective detection with taking reconstruction error into consideration.

**Ensembles based methods** This kind of methods are similar to bagging in machine learning. In [Lakshminarayanan et al., 2017], the authors initialize different parameters for neural networks randomly, and the bagging sampling method is used for generating training data. This method enjoys the benefits of bagging method. Similarly, in [Sastry and Oore, 2019], the features from different layers of neural network are used for identifying ood samples. The defined higher order Gram Matrices in this work yield better ood detection performance. More recently, [Shalev et al., 2019] converts the labels of training data into different word embeddings using GloVe or FastText as a supervision to gain diversity and redundancy, the semantic structure improve the robustness of neural networks.

**Some new tendency** In [Ren et al., 2019], the authors argue that the likelihood score is heavily affected by the population level background statistics, and thus, they propose a likelihood ratio method to deal with background and semantic targets in image data. In [Hendrycks et al., 2019], the study finds that self-supervision can benefit robustness of recognition tasks in a variety of ways. In [Zisselman and Tamar, 2020], a residual flow method is proposed for learning the distribution of feature space of a pre-trained deep neural network which can help to detect ood examples. The latest work in [Winkens and Bunel, 2020] treats ood samples as near-ood and far-ood samples, it argues that contrastive learning can capture much richer features which improves performance in detecting near-ood samples.

3 Problem Statement

This work considers the problem of seperating id and ood samples. Suppose $P_{in}$ and $P_{out}$ are distributions of id and
ood data, $X = \{x_1, x_2, ..., x_N\}$ are images randomly sampled from these two distributions. This task aims to give lower confidence of image $x_i$ sampled from $P_{out}$ while higher to that of $P_{in}$. Fig.2 (a) shows the challenges faced in traditional classification net, and what we expected is to reduce the confidence of ood data while maintaining that of id data.

In traditional neural networks, the confidence scores output by softmax are usually very high even for ood samples. We argue that the reason of this abnormal phenomenon is because the neural networks can not perceive the boundary of id data. Traditionally, the feature in different layer of neural networks is assumed to follow a multivariate gaussian distribution, and thus, as in works [Abdelzad et al., 2019; Lee et al., 2018b; Vernekar et al., 2019], it claims that the area with lower probability density of gaussian distribution can be treated as the edge of id data. But these methods exist three main nonnegligible drawbacks as follows:

**Reasonability** It is not guaranteed that the features from each layers follow a multivariate gaussian distribution, although it is seemingly straightforward sometimes.

**Curse of dimensionality** The neural network usually has extremely deep or wide architectures, and the number of channels is huge in feature maps. This will result in very high number of each layers follow a multivariate gaussian distribution, which still bring a markable improvement in ood detection.

**Inefficient sampling method** With a given multivariate gaussian distribution, it is in low efficiency if each sampling needs to compute the probability density.

For evading these drawbacks, we make no assumption on the distribution of features from the penultimate layer. We use a generator to capture the feature distribution while a discriminator to identify whether a data point is id or ood.

4 GAN Based Boundary Aware Classifier

In this section, we will talk the details of our proposed GBAC and give the method for sampling ood representations from a trained generator with high efficiency.

4.1 Architecture

The proposed framework has three basic components: the Representation Extraction Module (REM), the Representation Generation Module (RGM), and the Representation Sampling Module (RSM).

**REM** The pre-trained neural network is used for feature extraction. In [Abdelzad et al., 2019], the authors argue that the features from penultimate layer of trained neural nets are not suitable for detecting ood examples, but nowadays, the neural net has extremely deep architectures which makes searching the optimal layer a very burdensome task. In this work, we just use the features extracted from the penultimate layer which still bring a markable improvement in ood detection.

In the following parts, $\mathcal{H}$ and $h$ are used to indicate the pre-trained classification net with and without the top classification layer, and $\theta$ is the pre-trained weights. With these notations, the feature $f$ of an input image $x$ can be described as:

$$f = h(x; \theta)$$

**RGM** This module consists a conditional generator $G$ and a conditional discriminator $D$. The generator is used for generating feature representations of id data while the discriminator is used for giving quality scores to them. The latent variable $z$ is sampled from a normal distribution $P_z$. The features of training images from REM follow a distribution $P_f$. For learning the boundary of id data via discriminator, we propose **shuffle loss** and **uniform loss**. In each batch of training data, we get feature-label pairs like $(f, c)$. In a conditional GAN, these $(f, c)$ pairs are treated as positive samples. With a shuffle function $T(\cdot)$, the positive pair $(f, c)$ is transformed to a negative pair $(f, \hat{c})$ where $\hat{c} = T(c)$ is a mismatched label with feature $f$. The discriminator is expected to identify these samples as ood data, and the classification loss is the so called **shuffle loss** as below:

$$L_s = E_P_f (\log D(f; T(c)) - \log D(f; c))$$  \hspace{1cm} (2)

In feature space, each category of id data is concentrated in some specific regions densely, therefore the space beyond each cluster area should be treated as ood region. Given a batch features $\{f_1, f_2, f_3, ..., f_k\}$, the length of each feature $f_i$ is $m$. For uniformly sampling, we first calculate the minimal and maximal bound in $m$-dimensional space that contains all features within this batch. Thus, for all $j$, we have:

$$R_{\text{min}}^{(j)} = \min_{1 \leq i \leq k} f_{i,j}$$  \hspace{1cm} (3)

$$R_{\text{max}}^{(j)} = \max_{1 \leq i \leq k} f_{i,j}$$  \hspace{1cm} (4)

therefore, the lower and upper bound of feature vectors are got as follows:

$$a = (R_{\text{min}}^{(1)}, R_{\text{min}}^{(2)}, ..., R_{\text{min}}^{(m)})^T$$

$$b = (R_{\text{max}}^{(1)}, R_{\text{max}}^{(2)}, ..., R_{\text{max}}^{(m)})^T$$

We use $\mathbb{U}(a, b)$ to indicate a uniform distribution in feature space based on this batch of training data. Randomly sampled features $\tilde{f}$ from $\mathbb{U}(a, b)$ are treated as negative samples with randomly generated labels $\hat{c}$. Conflict happens when $\tilde{f}$ is sampled near one cluster and its randomly generated label $\hat{c}$ is consistent with that. For tackling these conflicts, we
strengthen the memory of discriminator about positive pairs. The uniform loss is defined as follows:

\[ L_u = \mathbb{E}_{P_z} \log D(\hat{f} ; \hat{c}) - \mathbb{E}_{P_z} \log D(f ; c) \]  

(7)

With notations above, the loss function \( L_d \) for discriminator can be formulated as below:

\[ L_t = -\mathbb{E}_{P_z} \log D(f ; c) - \mathbb{E}_{P_z} \log(1 - D(G(z) ; c)) \]  

(8)

\[ L_d = L_t + L_{id} + L_u \]  

(9)

where \( L_d \) is the loss of discriminator in traditional cGAN. A well trained discriminator is a binary classifier for separating \( id \) and \( ood \) data. The loss function for training generator is almost unchanged, we add a regularization term to accelerate the convergence. For positive feature-label pairs \( \langle f, c \rangle \), the loss function of generator is formulated as follows:

\[ L_g = \mathbb{E}_{P_z} \log(1 - D(G(z) ; c)) + \lambda \|f - G(z ; c)\|_1 \]  

(10)

where \( \| \cdot \|_1 \) indicates the L1 norm, and \( \lambda \) is a balance hyper-parameter. We set \( \lambda \) to 1 in our experiments.

### 4.2 Training Phase

There are two kinds of calibration method which are patched style and inplace style.

**Patched Style** This style of framework only need to train the cGAN. The discriminator is a binary classifier for separating \( id \) and \( ood \) data as mentioned before, and the confidence score outputed by a well trained discriminator can be treated as \( P(f \in \mathcal{M}_f | f) \). Here, \( \mathcal{M}_f \) represents the set of \( id \) features. For a given feature representation \( f \), based on the total probability theorem, we have:

\[ P(w | f) = P(w | f \in \mathcal{M}_f) \cdot P(f \in \mathcal{M}_f | f) + P(w | f \not\in \mathcal{M}_f) \cdot P(f \not\in \mathcal{M}_f | f) \]  

(13)

where \( w \) represents the category label of \( id \) data, and \( P(f \in \mathcal{M}_f | f) \) is the output of discriminator. In traditional neural networks, we have no access to \( ood \) data, and thus the softmax output is actually the conditional probability assuming the input are \( id \) data, which means \( P(w | f \in \mathcal{M}_f) \) is the output of pre-trained classification net.

Empirically, the \( ood \) data usually have quite different semantic meanings compared with \( id \) data, and thus it is reasonable to approximate \( P(w | f \not\in \mathcal{M}_f) \) to 0. Then, we have:

\[ P(w | f) \approx P(w | f \in \mathcal{M}_f) \cdot P(f \in \mathcal{M}_f | f) \]  

(14)

It tells that the approximation of posteriori can be formulated as the product of outputs from pre-trained classification net and discriminator. With this style of framework, there is no need to finetune the top classification layer, and the only drawback is to save the discriminator additionally.

**Inplace Style** This style of framework has no need to save the additional discriminator, but the pre-trained classification net has to be finetuned. We use \( P_0(y = j | f_i) \) to represent the propability of category label \( j \) outputed by softmax given feature \( f_i \). The positive features are generated via REM, and the negative \( hard \ ood \) data are generated by RSM. The softmax outputs for negative \( hard \ ood \) data are expected to follow a uniform distribution which means a much lower confidence score. Suppose we have \( C \) classes, therefore the cross-entropy loss \( (L_1) \) and KL-divergence loss \( (L_2) \) can be formulated as follows:

\[ L_1 = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} \mathbb{I}(y_i = j) \log P_0(y = j | f_i) \]  

(15)

\[ L_2 = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} Q(y = j | f_i) \log Q(y = j | f_i) / P_0(y = j | f_i) \]  

(16)

where \( \mathbb{I}(\cdot) \) is the indicator function, \( y_i \) is the target label of feature \( f_i \), \( Q(\cdot) \) is a classes-wise uniform distribution which means \( Q(\cdot) = \frac{1}{C} \) is always hold. Therefore, the Eq. (16) can be simplified to:

\[ L_2 = -\frac{1}{NC} \sum_{i=1}^{N} \sum_{j=1}^{C} \log C \cdot P_0(y = j | f_i) \]  

(17)

The objective of calibration framework is a weighted sum of cross-entropy loss and KL-divergence loss:

\[ L = L_1 + \lambda L_2 \]  

(18)

### 4.3 Inference Phase

The inference process is corresponding to the training process, we give the inference pipeline below.

**Patched Style** Recall that \( \mathcal{H}, h \) and \( D \) are classification net, feature extractor and discriminator correspondingly. For a given image \( x \), we have:

\[ h(x; \theta_h) = f \]  

(19)

\[ \mathcal{H}(x; \theta_h) = P(w | f \in \mathcal{M}_f) \]  

(20)

\[ D(f; \theta_D, \hat{c}) = P(f \in \mathcal{M}_f | f) \]  

(21)

where \( \hat{c} \) is the prediction of pre-trained classification net \( \mathcal{H} \). With Eq.(14), we can get the posteriori of a given image \( x \), and this confidence score can be used for evaluating the performance of \( ood \) detection.

\[ P(w | f) \approx \mathcal{H}(x; \theta_h) \cdot D(f; \theta_D, \hat{c}) \]  

(22)

\[ \mathcal{H}(x; \theta_h) \cdot D(h(x; \theta_h); \theta_D, \hat{c}) \]  

(23)
Inplace Style In this style of framework, since the classification net $\mathcal{H}$ has been calibrated, the output $\mathcal{H}(x; \theta_{\mathcal{H}})$ of a given image is the approximation of posteriori, and it can be used for detecting ood examples directly.

5 Experiments

In this section, we first introduce the datasets and evaluation metrics, and then, we give the details of training and parameter setting in experimental setup. The proposed GBAC is compared with some current techniques such as max-softmax (baseline) [Hendrycks and Gimpel, 2016], ODIN [Liang et al., 2020], GCPL [Yang et al., 2018], Mahalanobis distance based approach (MD) [Lee et al., 2018b] and so on.

5.1 Datasets

Synthetic toy noise datasets such as gaussian random noise or uniform noise are not adopted. During evaluation, we keep id and ood examples the same size.

MNIST and Fashion-MNIST These are two basic datasets in deep learning. MNIST is a handwritten digit dataset released in [LeCun, 1998] and Fashion-MNIST contains some clothing images which is released in [Xiao et al., 2017]. Omniglot is another handwritten dataset which contains 50 different alphabets released in [Lake et al., 2015]. This dataset is often used in few-shot learning or meta-learning tasks. In this work, we treat Omniglot as ood data.

CIFAR-10 and CIFAR-100 These two tiny image datasets are released in [Krizhevsky et al., 2009] as much hard image recognition benchmark.

TinyImageNet is a subset of ImageNet images [Deng et al., 2009] which contains 10k images belong to 200 categories.

SVHN is a dataset which contains street view houses numbers from real world released in [Netzer et al., 2011].

LSUN is a large-scale scene understanding dataset including bedroom, classroom, kitchen and many other scene images. This dataset is released in [Yu et al., 2015].

5.2 Evaluation metrics

We adopt the following commonly used metrics to measure the performance of detecting ood examples in this work.

FPR at 95% TPR is the probability of an ood example being misclassified as id examples when the True Positive Rate is as high as 95%. True positive Rate and False Positive Rate are the same as defined in ROC curve. Lower is better. In some tables, it is abbreviated to FPR 95%.

Detection Error measures the misclassification probability when True Positive Rate is as high as 95%. It is defined as $0.5(1 – TPR) + 0.5FPR$. Lower is better.

AUROC represents the area under ROC curve. Greater AUROC indicates that the neural network is more confident to assign higher score to id data than ood data. An ideal classifier has an AUROC score of 100%. Greater is better.

AUPR represents the area under Precision-Recall curve. AUPR$_{in}$ indicates the ability of detecting id examples while AUPR$_{ood}$ indicates that of ood examples. Both are expected to be higher.

5.3 Experimental setup

We give the training details of each architectures here. More detailed settings can be easily found in our released code.

Softmax as baseline ResNet and DenseNet are used as backbones, and they are trained with an Adam optimizer using cross-entropy loss in total 300 epochs. Weight decay is set to 1e-4. Batch size of all datasets is set to 64. The learning rate starts from 1e-3 and halves every 30 epochs. Data augmentation tricks such as rotation, flip and crop are used. Images from MNIST, Fashion-MNIST and Omniglot are resized to $28 \times 28$ with only one channel. Other datasets are resized to $32 \times 32$ with RGB channels. For MNIST, Fashion-MNIST and Omniglot, ResNet18 is used as feature extractor. For any other datasets, ResNet34 and DenseNet-BC with 100 layers are used for feature extraction.

GCPL We use distance based cross-entropy loss and prototype loss as mentioned in [Yang et al., 2018] for generalized convolutional prototype learning. The hyper-parameter $\lambda$ that controls the weight of prototype loss is set to 0.01.

ODIN Official code in ODIN is used. For any other methods, we do not adopt this technique for fair comparison.

MD is a mahalanobis distance based method without re-training the neural networks. The official code can be found at deep_mahalanobis_detector. It’s worth pointing out that this method uses data explicitly labeled as ood for training, therefore it’s unfair for comparison with other techniques. We just list the metric results from its original paper.

5.4 Detection results

We detail the experimental results on several datasets with ResNet and DenseNet-BC.

Results on MNIST, Fashion-MNIST and Omniglot. In this experiment, we set two groups for observing the effects of GBAC. The first group takes MNIST as id data, and the mixture of Fashion-MNIST and Omniglot is treated as ood data. It’s worth noting that in all experiments, we use no ood data for training or tuning the model, and no access to ood data is our basic setting. The second group takes Fashion-MNIST as id data while MNIST and Omniglot as ood data. For simplicity, Cls Acc and Det Err are used to represent Classification Accuracy and Detection Error correspondingly. The results listed in Table.1 promise that GBAC dramatically improves the softmax baseline and outperforms the other techniques with a markable gap. For ODIN, we use a temperature 10 and magnitude 5e-4, no hyper-parameter searching is performed.

| ID OOD | MNIST | F-MNIST & Omniglot | F-MNIST & Omniglot |
|--------|-------|---------------------|---------------------|
| Methods | baseline / ODIN / GCPL / GCPL (baseline) | 99.43 / 99.43 / 99.23 / 99.43 | 91.51 / 91.51 / 90.93 / 91.51 |
| † Cls Acc | 94.14 / 5.01 / 4.77 / 3.06 | 94.14 / 5.01 / 4.77 / 3.06 | 94.14 / 5.01 / 4.77 / 3.06 |
| † Det Err | 3.29 / 5.03 / 4.54 / 1.11 | 3.29 / 5.03 / 4.54 / 1.11 | 3.29 / 5.03 / 4.54 / 1.11 |
| † AUROC | 97.66 / 97.94 / 97.96 / 99.32 | 97.66 / 97.94 / 97.96 / 99.32 | 97.66 / 97.94 / 97.96 / 99.32 |
| † AUPR in | 97.22 / 97.42 / 98.14 / 99.46 | 97.22 / 97.42 / 98.14 / 99.46 | 97.22 / 97.42 / 98.14 / 99.46 |
| † AUPR out | 97.24 / 97.64 / 97.35 / 99.09 | 97.24 / 97.64 / 97.35 / 99.09 | 97.24 / 97.64 / 97.35 / 99.09 |

Table 1: Detecting ood samples on MNIST, Fashion-MNIST and Omniglot with ResNet18.
DenseNet-BC on CIFAR-10 dataset. The points in deep pink

Table 2: Detecting ood samples on CIFAR-10, CIFAR-100 and SVHN with ResNet34 and DenseNet-BC. The softmax baseline, ODIN and proposed GBAC use no ood data for training or tuning, therefore it doesn’t make sense to compared these methods with MD. We just list the results of MD served as an upper bound.

| ID | OOD | ↓ FPR (at 95% TPR) | ↑ AUROC | ↑ AUPR in | ↑ AUPR out |
|----|-----|----------------------|---------|-----------|-----------|
| CIFAR-10 DenseNet-BC | SVHN | 59.8/1.5/63.6/1.6/9.2 | 89.9/99.7/87.2/99.7/98.1 | 91.9/99.7/89.1/99.5/ - | 87.0/99.8/83.9/99.7/ - |
| | LSUN | 33.4/11.6/5.6/5.5/2.8 | 95.4/98.2/98.9/99.0/99.3 | 96.4/98.3/98.9/99.0/ - | 94.0/98.1/98.7/89.3/ - |
| | TinyImageNet | 41.1/19.0/10.5/10.2/5.0 | 94.1/97.0/98.9/98.1/98.8 | 95.3/97.2/98.1/98.2/ - | 92.2/96.8/97.8/89.0/ - |
| CIFAR-10 ResNet-34 | SVHN | 67.5/16.1/64.4/14.8/3.6 | 89.9/97.1/83.9/96.3/99.1 | 92.2/96.7/85.8/95.5/ - | 84.9/97.4/81.8/97.1/ - |
| | LSUN | 54.6/40.5/26.2/19.1/1.1 | 91.0/92.7/94.1/93.6/99.7 | 92.3/93.0/93.9/93.7/ - | 88.5/91.8/94.1/93.2/ - |
| | TinyImageNet | 55.3/43.6/28.0/31.6/2.9 | 91.0/92.3/93.9/93.5/99.5 | 92.4/93.0/94.0/93.9/ - | 88.3/91.3/93.8/92.9/ - |
| CIFAR-100 DenseNet-BC | SVHN | 73.3/51.9/60.9/49.9/17.9 | 82.7/90.1/88.2/90.2/97.2 | 85.9/91.5/90.2/91.5/ - | 78.5/88.6/85.2/88.8/ - |
| | LSUN | 83.3/54.0/58.4/42.9/8.6 | 70.8/86.0/85.7/90.2/98.0 | 72.4/84.8/85.0/89.3/ - | 65.4/84.0/82.0/88.4/ - |
| | TinyImageNet | 82.4/51.0/56.9/38.2/13.4 | 71.7/87.3/85.3/91.8/97.4 | 73.0/86.7/84.7/91.5/ - | 67.4/86.1/83.0/90.6/ - |
| CIFAR-100 ResNet-34 | SVHN | 79.7/47.1/76.5/45.6/8.1 | 79.5/87.4/74.8/85.5/99.4 | 81.5/85.9/73.8/87.1/ - | 74.5/88.2/74.2/88.7/ - |
| | LSUN | 81.2/51.3/54.6/49.7/9.1 | 75.8/87.9/85.0/90.7/98.2 | 76.0/87.3/82.4/91.5/ - | 70.1/86.5/84.1/88.8/ - |
| | TinyImageNet | 79.6/51.5/50.6/49.7/9.1 | 77.2/88.5/87.6/90.6/98.2 | 79.2/88.9/86.9/91.6/ - | 72.3/87.3/87.0/88.9/ - |
| SVHN DenseNet-BC | LSUN | 22.9/18.9/22.1/18.7/2.1 | 94.1/96.0/92.6/96.2/99.9 | 96.7/98.3/95.3/98.5/ - | 88.0/89.1/89.2/89.3/ - |
| | CIFAR-10 | 30.7/25.0/24.7/20.2/3.2 | 91.9/99.4/93.1/98.5/99.8 | 95.4/97.3/92.5/97.3/ - | 83.5/86.7/81.7/86.9/ - |
| | TinyImageNet | 21.2/15.4/19.9/15.2/0.1 | 94.8/96.5/93.6/96.6/99.9 | 97.0/98.4/95.5/98.7/ - | 88.9/90.4/90.1/90.6/ - |
| SVHN ResNet-34 | LSUN | 25.7/18.8/22.2/18.7/0.1 | 91.6/95.4/89.4/95.3/99.9 | 93.8/97.9/91.9/97.8/ - | 84.6/88.8/85.9/88.8/ - |
| | CIFAR-10 | 21.7/18.0/20.0/18.1/1.6 | 92.9/99.3/90.5/99.2/99.3 | 94.8/98.7/91.9/97.6/ - | 86.4/89.0/87.1/89.0/ - |
| | TinyImageNet | 21.0/16.9/18.0/16.7/0.1 | 93.5/96.0/92.0/95.9/99.9 | 95.4/98.2/93.5/98.2/ - | 86.9/89.2/88.6/89.4/ - |

* Uses data explicitly labeled as ood for training.

Results on CIFAR-10, CIFAR-100 and SVHN. We set sufficient experiments in this part for testing the generalization ability of proposed GBAC. The pre-trained ResNet-34 and DenseNet-BC on CIFAR-10, CIFAR-100 and SVHN come from ODIN. Since Detection Error is consistent with FPR at 95% TPR, therefore we just give the results of the latter. From Table 2 we can see that GBAC improves the metric results significantly no matter combined with softmax baseline or ODIN which promises its generalization ability. For ODIN, optimal magnitudes and temperatures are searched in each group.

**t-SNE visualization for hard ood features.** We use Eq.11 to sample hard ood features of each category. The dimensions of feature space in penultimate layer are 512 and 342 for ResNet34 and DenseNet-BC correspondingly, thus we use t-SNE to reduce the dimension of features allowing intuitive visual verification.

![t-SNE visualization](image)

Figure 3: Synthetic data generated by the generator.

In Fig.3, the green points are real features extracted from DenseNet-BC on CIFAR-10 dataset. The points in deep pink are synthetic id features directly generated with a trained generator while the points in yellow are hard ood samples got via RSM. It can be seen clearly that with RSM, the synthetic id features which are confused with real data are pushed away from the clusters.

6 Conclusion and Discussion

In this paper, we propose GBAC which leverages the benefits of generative model and ensemble methods. GBAC is inspired by the fact that traditional classification nets only focus on the separation between classes and thus divide the feature space as several open regions without explicit boundary. This manner is not suitable for ood detection since the classification net can not aware the boundary of each category in training data. With GBAC, we train a boundary aware discriminator as an auxiliary module which dramatically boosts the performance of ood detection, and the discriminator of GBAC can be well combined with different classification nets. Besides, with our proposed framework we can get a generator simultaneously, the generator can be used for generating id features and hard ood features. The sufficient experiment results promise the generalization ability of method, and the visualization results provide an intuitive way to realize the method and its effects. Recently, some new works introduce self-supervised learning methods into ood detection which can help generate more rich features. This kind of techniques believe that with some redundancy methods, the neural nets can learn some task-agnostic features which may promise some improvement on many vision tasks, e.g., ood detection, few-shot learning, highlights extraction, metric learning, etc.
References

[Abdelzad et al., 2019] Vahdat Abdelzad, Krzysztof Czarnecki, Rick Salay, Taylor Denouden, Sachin Vernekar, and Buu Phan. Detecting out-of-distribution inputs in deep neural networks using an early-layer output. CoRR, abs/1910.10307, 2019.

[Amodei et al., 2016] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. CoRR, abs/1606.06565, 2016.

[Deng et al., 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[Denouden et al., 2018] Taylor Denouden, Rick Salay, Krzysztof Czarnecki, Vahdat Abdelzad, Buu Phan, and Sachin Vernekar. Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance. CoRR, abs/1812.02765, 2018.

[DeVries and Taylor, 2018] Terrance DeVries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks, 2018.

[Goodfellow et al., 2014a] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. Computer Science, 2014.

[Goodfellow et al., 2014b] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[Hendrycks and Gimpel, 2016] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. CoRR, abs/1610.02136, 2016.

[Hendrycks et al., 2019] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. Advances in Neural Information Processing Systems (NeurIPS), 2019.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[Krizhevsky et al., 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Lake et al., 2015] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. Science, 350(6266):1332–1338, 2015.

[Lakshminarayanan et al., 2017] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[LeCun, 1998] Yann LeCun. The mnist database of handwritten digits. $http://yann.lecun.com/exdb/mnist/$. 1998.

[Le et al., 2018a] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples, 2018.

[Lee et al., 2018b] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks, 2018.

[Liang et al., 2020] Shiyu Liang, Yixuan Li, and R. Srikanth. Enhancing the reliability of out-of-distribution image detection in neural networks, 2020.

[Netzer et al., 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. NIPS, 01 2011.

[Nguyen et al., 2017] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, 2015.

[Ren et al., 2019] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection, 2019.

[Sastry and Oore, 2019] Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with in-distribution examples and gram matrices. CoRR, abs/1912.12510, 2019.

[Shalev et al., 2019] Gabi Shalev, Yossi Adi, and Joseph Keshet. Out-of-distribution detection using multiple semantic label representations, 2019.

[Sohn et al., 2015] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc., 2015.

[Vernekar et al., 2019] Sachin Vernekar, Ashish Gaurav, Vahdat Abdelzad, Taylor Denouden, Rick Salay, and Krzysztof Czarnecki. Out-of-distribution detection in classifiers via generation. CoRR, abs/1910.04241, 2019.

[Winkens and Bunel, 2020] Jim Winkens and Rudy Bunel. Contrastive training for improved out-of-distribution detection. CoRR, abs/2007.05566, 2020.

[Xiao et al., 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[Yang et al., 2018] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. CoRR, abs/1805.03438, 2018.

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

[Zisselman and Tamar, 2020] Ev Zisselman and Avi Tamar. Deep residual flow for out of distribution detection, 2020.