Towards In-Distribution Compatible Out-of-Distribution Detection

Boxi Wu[1]*, Jie Jiang[2]*, Haidong Ren[4], Zifan Du[3], Wenxiao Wang[3],
Zhifeng Li[2], Deng Cai[1], Xiaofei He[1], Binbin Lin[3]†, Wei Liu[2]
[1]State Key Lab of CAD&CG, Zhejiang University.
[2]Tencent Data Platform.
[3] School of Software Technology, Zhejiang University.
[4]Ningbo Zhoushan Port Group Co.,Ltd., Ningbo, China.

Abstract

Deep neural network, despite its remarkable capability of discriminating targeted in-distribution samples, shows poor performance on detecting anomalous out-of-distribution data. To address this defect, state-of-the-art solutions choose to train deep networks on an auxiliary dataset of outliers. Various training criteria for these auxiliary outliers are proposed based on heuristic intuitions. However, we find that these intuitively designed outlier training criteria can hurt in-distribution learning and eventually lead to inferior performance. To this end, we identify three causes of the in-distribution incompatibility: *contradictory gradient*, *false likelihood*, and *distribution shift*. Based on our new understandings, we propose a new out-of-distribution detection method by adapting both the top-design of deep models and the loss function. Our method achieves in-distribution compatibility by pursuing less interference with the probabilistic characteristic of in-distribution features. On several benchmarks, our method not only achieves the state-of-the-art out-of-distribution detection performance but also improves the in-distribution accuracy.

Introduction

Deep neural networks have achieved extraordinary performance across a wide range of artificial intelligence and pattern recognition tasks. Many of these tasks are formulated in a constrained scenario. That is, all the considered training and testing samples are assumed to belong to a few predefined limited categories. This is the case for many standard computer vision tasks such as classification (Krizhevsky 2012), detection, and segmentation. Naturally, people start to wonder how deep networks will react to out-of-distribution (OOD) data, data that do not belong to any of the predefined categories.

Hendrycks and Gimpel (2017) first studied this question and found that deep networks tend to assign high confidence scores to OOD samples. This problem hugely hampers safely deploying deep models in the open world. The behavior of artificial intelligence applications such as autonomous driving and medical image processing (Ren et al. 2019) can be unpredictable when facing OOD data. Various approaches have been proposed to solve this problem. One of the most effective approaches is Outlier Exposure (OE) (Hendrycks, Mazeika, and Dietterich 2019), which chooses to train deep networks with an auxiliary dataset of outliers. (Hendrycks, Mazeika, and Dietterich 2019) pointed out that the more realistic the auxiliary dataset is, the better performance for OOD detection.

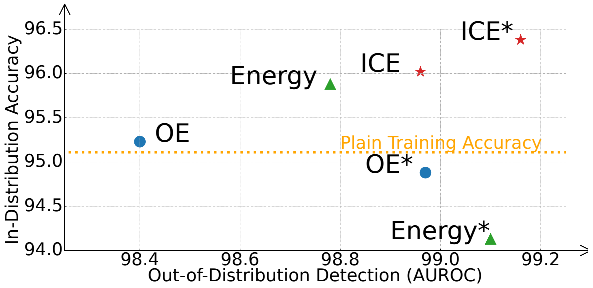


Figure 1: The in-distribution accuracies and out-of-distribution detection AUROC scores of different models. Models annotated with * indicate the model is trained from scratch; otherwise, the model is fine-tuned.

The original work of OE adopts the KL-divergence to the uniform distribution as the training criterion of outliers. Subsequent works followed the framework of Outlier Exposure and proposed various different training criteria of outliers for better effectiveness (Liu et al. 2020). Most of these are designed based on intuition. However, when experimenting with these methods, we find that training with OE can lead to inferior in-distribution accuracy. To better understand this issue, we dive into the detailed design of these OOD training criteria and identify three major factors that cause the in-distribution incompatibility:

1. **Contradictory Gradient.** The training objective of existing OE methods occasionally generates gradients in the opposite direction of those generated by the in-distribution objective, which further hampers the in-distribution discriminant learning.
2. **False Likelihood.** Existing OE algorithms adopt the logit value generated by neural networks as a surrogate signal for likelihood estimation. Yet, under Gaussian discrimi-

*These authors contributed equally.

†Binbin Lin is the corresponding author.

nant analysis (Lee et al. 2018), the logit value is a defective estimation of likelihood, which can result in assigning high in-distribution scores on outliers.

3. **Distribution Shift.** OE objective essentially shifts OOD features to the center of features space. During this process, in- and out-of-distribution features may mix with each other, which prevents separating different classes.

These three factors are intrinsically related and together hamper in-distribution learning. In Section 3, we will demonstrate these three factors for both linear models as well as deep networks. Existing OE methods partially addressed one or two factors, intentionally or unintentionally. Yet, an in-distribution-compatible OE algorithm that can universally solve all these problems is still absent. Therefore, in Section 4, we intend to design a new OE method, **I**n-distribution **C**ompatible outlier **E**xposure (ICE), which can achieve high performance on in-distribution classification and out-of-distribution detection simultaneously, as shown in Fig. 1. Our loss function is designed based on the principle of **not contradicting the in-distribution gradients**. Meanwhile, we focus on the probabilistic characteristic of in-distribution samples. Commonly-used deep models generate high-dimension in-distribution features that approximately form a class-conditional Gaussian distribution. Then a linear layer will transform the features to the scalar logit value. **To avoid the false likelihood problem**, we replace the linear layer with a Gaussian mixture model so that we can estimate the actual class-conditional likelihood. **To prevent distribution shift**, our loss function is designed to push the outliers away from the closest in-distribution cluster so that the in-distribution cluster can be minimally affected.

We test ICE on benchmark datasets in Section 5. In the most challenging case where CIFAR10 (Krizhevsky 2012) serves as the in-distribution set and CIFAR100 as the out-of-distribution set, ICE improves the FPR95 score to 22.36% with an improvement of 3.79% over previous *sota* results. Meanwhile, ICE achieves high in-distribution accuracy from 95.11% to 96.38% over the plain in-distribution training. The visualization of features learned by ICE further verifies the soundness of our design.

Related Works

The pioneering work of Hendrycks and Gimpel (2017) pointed out the importance of detecting outliers with deep networks. Hein, Andriushchenko, and Bitterwolf (2019) analyzed that the ReLU activation can hamper detecting outliers. To alleviate this issue, Hendrycks and Gimpel (2017) proposed to use the maximum soft-max score as an indicator of outliers. Similarly, Liang, Li, and Srikant (2018) proposed a refined outlier score by adopting temperature scaling and adding small perturbations to input. Hsu et al. (2020) further improved the framework of Liang, Li, and Srikant (2018) by decomposing confidence scoring. Other different forms of confidence scores were kept being proposed (Ren et al. 2019; Hendrycks et al. 2022). Lee et al. (2018) considered using a confidence score of Mahalanobis distance to detect abnormal samples. Xie et al. (2021) address detection performance by using auxiliary information.

The works mentioned above mainly improve detection performance without modifying the trained models. The methodology of Outlier Exposure (Hendrycks, Mazeika, and Dietterich 2019) considers using training techniques with an auxiliary dataset of outliers and achieves superior results. Many OOD detection techniques are constrained to small-scale datasets. Recently, several works (Sun, Guo, and Li 2021; Huang and Li 2021; Yang et al. 2021) investigated the detection task on large-scale datasets such as ImageNet. Lin, Roy, and Li (2021) accelerated the detection speed with multi-level features. Wang et al. (2022) improve the performance by crafting virtual logit from heterologous confidence scores. Wei et al. (2022) proposed an alternative loss function to replace the commonly-used softmax cross-entropy loss, which can be combined with outlier exposure methods for better performance.

In-Distribution Incompatibility

The state-of-the-art Outlier Exposure accomplishes the task of OOD detection via training outliers with an extra loss function. Different approaches propose their own outlier loss function. In this section, we use two of the most representative outlier loss functions, the standard KL-Divergence to uniform distribution (Hendrycks, Mazeika, and Dietterich 2019) and the Energy score (Liu et al. 2020), to illustrate the three causes of in-distribution incompatibility.

Preliminaries

We choose the fundamental problem of multi-class classification as the subject to illustrate our methodology. We first introduce the notations and rudiments of the studied topic as below.

Notations. For the multi-classification task with K classes, a neural work will first transform input $\mathbf{x}$ with a non-linear mapping to feature $\mathbf{z} = Z(\mathbf{x}) \in \mathbb{R}^d$. Z represents deep network backbone. Then, a linear model f is supposed to output a real-value score for each class:

$$f(\mathbf{x}) = \boldsymbol{W}\mathbf{z} + \boldsymbol{b}. \tag{0.1}$$

$\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K]$ and $\boldsymbol{b} = [b_1, \cdots, b_K]$ are trainable parameters. $f_i(\mathbf{z}) = \boldsymbol{w}_i^\top \mathbf{z} + b_i$ is the predicted score for class $i \in [K]$, which is known as the logit value. $[K] := \{1, \cdots, K\}$. $\boldsymbol{w}_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$ compute a scalar score for class i from feature $\mathbf{z}$. Finally, the stochastic gradient-descending algorithm will optimize the Softmax Cross-Entropy loss:

$$\mathcal{L}_{\text{sce}}(\mathbf{x}, \mathbf{1}_y) = -\mathbf{1}_y^\top \log\big[\text{softmax}[f(\mathbf{z})]\big]. \tag{0.2}$$

y is the ground-truth label for $\mathbf{x}$, and $\mathbf{1}_y \in \mathbb{R}^K$ is its one-hot encoding. We define the softmax function $\text{softmax}(f) : \mathbb{R}^K \to \mathbb{R}^K$ as $\text{softmax}(f)_i = \exp(f_i)/\sum_{k=1}^K \exp(f_k)$. the logarithm is defined as element-wise.

Outlier Exposure. Denoting the in-distribution set as $\mathcal{D}_{in}$, any possible input that does not belong to $\mathcal{D}_{in}$ is considered to belong to the OOD set $\mathcal{D}_{out}$. The intriguing part of detecting OOD data is that we cannot cover the entire $\mathcal{D}_{out}$ set during the training stage. Nevertheless, Hendrycks, Mazeika, and Dietterich (2019) showed that a good choice of a subset of $\mathcal{D}_{out}$ is crucial for learning models that can effectively detect unseen testing OOD data $\mathcal{D}_{out}^{test}$. Particularly, denoting the auxiliary set of outliers as $\mathcal{D}_{out}^{oe}$, the network is trained with:

$$\underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{in}}\mathcal{L}_{\text{sce}}(\mathbf{x},\mathbf{1}_y)}_{\text{in-distribution risk}} + \lambda \cdot \underbrace{\mathbb{E}_{\widetilde{\mathbf{x}}\sim\mathcal{D}_{out}^{oe}}\mathcal{L}_{\text{sce}}(\widetilde{\mathbf{x}},\mathbf{u})}_{\text{outlier exposure}}. \tag{0.3}$$

$\mathbf{u} \in \mathbb{R}^K$ is the uniform distribution over K classes. Weight parameter λ balances the two training objectives. For better detection performance, Liu et al. (2020) proposed to use the energy score as an alternative loss function for outliers:

$$\mathcal{L}_{\text{energy}}(\widetilde{\mathbf{x}}) = \log\big[\sum\nolimits_{k\in[K]}\exp(\boldsymbol{w}_k^\top\widetilde{\mathbf{z}} + b_k)\big]. \tag{0.4}$$

$\widetilde{\mathbf{z}} = Z(\widetilde{\mathbf{x}})$. Minimizing the above metric results in low confidence in all the in-distribution categories. To balance in-distribution energy, the negative energy of in-distribution features $-\mathcal{L}_{\text{energy}}(\mathbf{x})$ are simultaneously optimized:

$$\begin{aligned}&\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{in}}\mathcal{L}_{\text{sce}}(\mathbf{x},\mathbf{1}_y) + \\ &\lambda\cdot\big[\mathbb{E}_{(\mathbf{x})\sim\mathcal{D}_{in}} - \mathcal{L}_{\text{energy}}(\mathbf{x}) + \mathbb{E}_{\widetilde{\mathbf{x}}\sim\mathcal{D}_{out}^{oe}}\mathcal{L}_{\text{energy}}(\widetilde{\mathbf{x}})\big].\end{aligned} \tag{0.5}$$

In the following parts, we will discuss the differences between these two carefully-designed criteria and their common deficiencies.

Contradictory Gradient

The gradient generated by training criteria can reveal how it influences trainable variables. For instance, the softmax cross-entropy in Eqn. (0.2) generates gradients as:

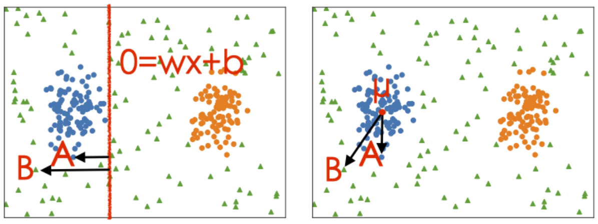
$$\frac{\partial\mathcal{L}_{\text{sce}}(\mathbf{x},\mathbf{1}_y)}{\partial f_i(\mathbf{z})} = \begin{cases}\frac{\exp[f_i(\mathbf{z})]}{\sum_{k\in[K]}\exp[f_k(\mathbf{z})]} - 1 < 0, & \text{if } y = i;\\ \frac{\exp[f_i(\mathbf{z})]}{\sum_{k\in[K]}\exp[f_k(\mathbf{z})]} > 0, & \text{if } y \neq i.\end{cases}$$

With the gradient-descending optimization, $\mathcal{L}_{\text{sce}}$ will increase the logit score $f_i(\mathbf{x})$ when i being the ground-truth class, otherwise decrease $f_i(\mathbf{x})$. Such design aligns with the belief that $f_i(\mathbf{x})$ is a good hint of estimating the confidence that $\mathbf{x}$ belongs to category i. However, the OE objective in Eqn. (0.3) contradicts the principle that only the ground-truth logit $f_y(\mathbf{x})$ is enlarged during training. Specifically, the gradient generated by $\mathcal{L}_{\text{sce}}(\widetilde{\mathbf{x}},\mathbf{u})$ in Eqn. (0.3) is:

$$\frac{\partial\mathcal{L}_{\text{sce}}(\widetilde{\mathbf{x}},\mathbf{u})}{\partial f_i(\widetilde{\mathbf{z}})} = K\cdot\Big[\frac{\exp[f_i(\widetilde{\mathbf{z}})]}{\sum_{k\in[K]}\exp[f_k(\widetilde{\mathbf{z}})]} - \frac{1}{K}\Big]. \tag{0.6}$$

When the softmax probability $\mathrm{softmax}(f)_i$ exceeds $1/K$, $\mathcal{L}_{\text{sce}}(\widetilde{\mathbf{x}},\mathbf{u})$ will generate positive gradient and thus surpass the value of $f_i(\widetilde{\mathbf{z}})$. This is reasonable as it reduces the confidence of predicting class i on outlier $\widetilde{\mathbf{x}}$. The problem is, when $\mathrm{softmax}(f)_i < 1/K$, $\mathcal{L}_{\text{sce}}(\widetilde{\mathbf{x}},\mathbf{u})$ will generate negative gradient and increases the confidence of predicting class i. Although such a case only happens when the posterior is lower than $1/K$, it still violates the principle mentioned above and increases the probability of mistaking $\widetilde{\mathbf{x}}$ with an in-distribution class. In contrast, the energy objective $\mathcal{L}_{\text{energy}}(\widetilde{\mathbf{x}})$ in (0.4) obeys the principle by punishing $f_i(\widetilde{\mathbf{z}})$ for all classes:

$$\frac{\partial\mathcal{L}_{\text{energy}}(\widetilde{\mathbf{x}})}{\partial f_i(\widetilde{\mathbf{z}})} = \frac{\exp[f_i(\widetilde{\mathbf{z}})]}{\sum_{k\in[K]}\exp[f_k(\widetilde{\mathbf{z}})]} > 0. \tag{0.7}$$



(a) Logit Value $f_i(\mathbf{z})$. (b) Class Likelihood $p(i|\mathbf{z})$.

Figure 2: In a 2-dimension binary classification problem, we compare the difference between logit value $f_i(\mathbf{z})$ and class likelihood $p(i|\mathbf{z})$. For in-distribution sample A and out-of-distribution sample B, $f_i(\mathbf{z})$ assigns higher confidence on B, while $p(i|\mathbf{z})$ assigns higher confidence on A.

Despite that the $\mathcal{L}_{\text{energy}}$ is compatible with in-distribution learning, the energy score falls into the issue of contradictory gradient due to the negative in-distribution energy $-\mathcal{L}_{\text{energy}}(\mathbf{x})$ in (0.4):

$$\frac{\partial - \mathcal{L}_{\text{energy}}(\mathbf{x})}{\partial f_i(\mathbf{z})} = -\frac{\exp[f_i(\mathbf{z})]}{\sum_{k\in[K]}\exp[f_k(\mathbf{z})]} < 0. \tag{0.8}$$

$-\mathcal{L}_{\text{energy}}(\mathbf{x})$ increase $f_i(\mathbf{z})$ for all classes instead of only the ground-truth, which contradicts the gradient of $\partial\mathcal{L}_{\text{sce}}(\mathbf{x},\mathbf{1}_y)/\partial f_i(\mathbf{z})$ when $i \neq y$. The contradictory gradient confuses the learning process by sending signals in opposite directions, eventually leading to inferior performance.

False Likelihood

One effective way to detect outlier $\widetilde{\mathbf{x}}$ is by estimating the likelihood of each class. A potential outlier is supposed to have a low likelihood value for all classes. Many outlier loss functions are designed based on this belief. Yet, many of these loss functions use the logit value $f_i(\widetilde{\mathbf{z}})$ as a substitute for class likelihood $p(\mathbf{x}|y = i)$. These two scores are not equivalent. Occasionally, an outlier with a low likelihood value $p(\mathbf{x}|y = i)$ can be assigned with a high logit score.

To understand this problem, we need to first inspect the intrinsic reasons why in-distribution learning uses the linear model in Eqn. (0.1) to generate a confidence score f_i. Under the assumption that the learned features $\mathbf{z}$ distribute as a class-conditional Gaussian: $p(\mathbf{z}|y) \sim \mathcal{N}(\boldsymbol{\mu}_y, \mathbf{\Sigma})$, $\mathbf{\Sigma}$ is a tied covariance, and $\boldsymbol{\mu}_i$ is the mean of class i, the methodology of logistic regression deduces from the Bayes' Theorem that the posterior probability $p(i|\mathbf{z})$ is equal to:

$$\begin{aligned}p(i|\mathbf{z}) &= \frac{\widehat{\boldsymbol{w}}_i^\top\mathbf{z} + \widehat{b}_i}{\sum_{k\in[K]}(\widehat{\boldsymbol{w}}_k^\top\mathbf{z} + \widehat{b}_k)}, \text{ where}\\ \widehat{\boldsymbol{w}}_i &= \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i,\ \widehat{b}_i = -\frac{1}{2}\boldsymbol{\mu}_i^\top\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i.\end{aligned} \tag{0.9}$$

Thus, under the assumption that parameters $\boldsymbol{w}_i$ and b_i can fit the $\widehat{\boldsymbol{w}}_i$ and $\widehat{b}_i$ from training, f_i is a desired in-distribution discriminator. From geometric perspective, the linear model f_i measures the distance to the hyper-plane of $\mathbf{0} = \boldsymbol{w}_i^\top\mathbf{z}+b_i$, while the likelihood $p(\mathbf{z}|i)$ is proportional to the negative

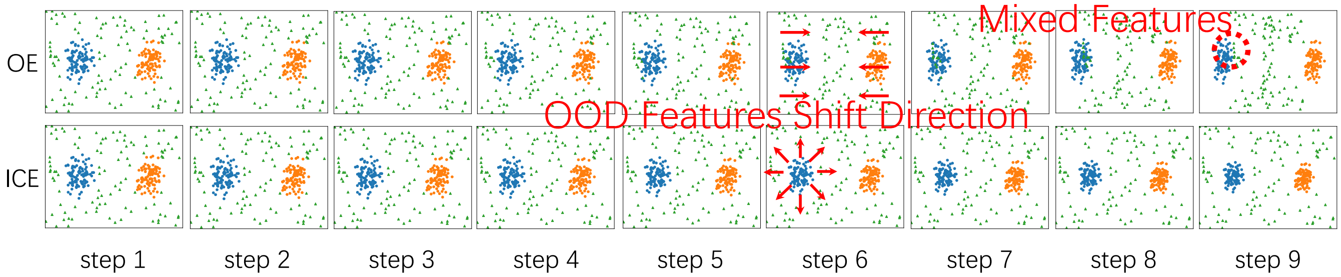


Figure 3: The distribution shift effect is caused by different methods. We plot feature distribution for each step of SGD optimization. Standard Outlier Exposure shifts outliers to the center of space, during which the in- and out-of-distribution samples are mixed. On the other hand, ICE pushes outliers away from the class center and does not suffer from distribution shift.

Mahalanobis distance to the class center $\boldsymbol{\mu}_i$. These two measurements are not equivalent. To demonstrate the difference, we visualize the case of a 2-dimensional binary classification problem. In Fig. 2(a), we can observe that an in-distribution sample A has a lower value of f_y compared with an out-of-distribution sample B, while the likelihood of A is higher than B. Such inconsistency between $f_i(\mathbf{z})$ and $p(i|\mathbf{z})$ is overlooked by previous works, including OE and Energy.

Distribution Shift

The high-dimension feature generated by the penultimate layer of the deep network, *i.e.*$\mathbf{z} = Z(\mathbf{x})$, is trainable. Thus, unlike the static $\mathbf{x}$, the distribution of $\mathbf{z}$ will shift with respect to the imposed loss functions. When only in-distribution features exist, the softmax cross-entropy loss $\mathcal{L}_{\text{sce}}$ will push features away from the discriminant plane. However, when out-of-distribution features are considered, the shifted distribution between in-distribution features and out-of-distribution features may mix, which interferes with the discrimination of deep networks.

We still use the 2-dimension visualization for illustration. In Fig. 2, we set the training features, either in-distribution or out-of-distribution, to be trainable to simulate features generated by deep networks and plot the distribution shift on each optimization step of the gradient-descending algorithm. Parameters $\boldsymbol{w}_i$ and b_i are initialized with $\widehat{\boldsymbol{w}}_i$ and $\widehat{b}_i$. The standard OE objective $\mathcal{L}_{\text{sce}}(\widetilde{\mathbf{x}}, \mathbf{u})$ will push the gradient to:

$$\begin{aligned}\partial \mathcal{L}_{\text{sce}}(\widetilde{\mathbf{x}}, \mathbf{u}) / \partial \widetilde{\mathbf{z}} = & \left[2 \cdot \operatorname{softmax}[f(\widetilde{\mathbf{z}})]_1 - 1\right] \boldsymbol{w}_1 \\ & + \left[2 \cdot \operatorname{softmax}[f(\widetilde{\mathbf{z}})]_2 - 1\right] \boldsymbol{w}_2 .\end{aligned} \tag{0.10}$$

If $\operatorname{softmax}[f(\widetilde{\mathbf{z}})]_1 > 1/2$, the above gradient tend to decrease the value of $f(\widetilde{\mathbf{z}})_1$ and increases $f(\widetilde{\mathbf{z}})_2$, and vice versa if $\operatorname{softmax}[f(\widetilde{\mathbf{z}})]_2 > 1/2$. As a result, out-of-distribution features are dragged to the space between the two in-distribution classes. For OOD features that are initially far away from either of the two in-distribution classes, the OE objective may shift them to the position of in-distribution features during the optimization iterations. These misplaced features then confuse in-distribution learning, as the discriminator is required to make opposite predictions on similar features. In the next section, we will show that our method avoids this issue by encouraging in-distribution features to gather in the center of its distribution and pushing outliers away from the center, as also shown in Fig. 3.

Methodology

In this section, we introduce our **I**n-distribution **C**ompatible outlier **E**xposure (ICE) to address the three problems discussed before. To this end, we adapt both the top design of deep models and the loss function. Besides the refined in-distribution compatibility, our ICE also provides direct probabilistic confidence estimation and saves the usage of hyper-parameters.

Modified Top-Design

We first replace the traditional linear layer of deep networks with the class-conditional Gaussian model. Namely, instead of using Eqn. (0.1) to generate a confidence score, we opt for the following estimation based on Mahalanobis distance:

$$h_i(\mathbf{z}) = -(\mathbf{z} - \mathbf{m}_i)^\top (\mathbf{L}\mathbf{L}^\top)^{-1} (\mathbf{z} - \mathbf{m}_i). \tag{0.11}$$

$\boldsymbol{\mu}_i$ is a trainable parameter that is supposed to simulate the class center $\boldsymbol{\mu}_i$. $\mathbf{L}$ is a real lower triangular matrix with positive diagonal entries, whose elements are also trainable. Based on the Cholesky decomposition, $\mathbf{L}\mathbf{L}^\top$ is guaranteed to be a symmetric positive-definite real matrix and is supposed to learn the covariance matrix $\boldsymbol{\Sigma}$. Previous works in different manners have investigated the philosophy of designing Mahalanobis-distance-based scores. Lee et al. (2018) statistically computed the mean and variance matrix from features generated by a trained network. Such a design cannot leverage the benefit brought by training on auxiliary outliers. Pang et al. (2020) also proposed Max-Mahalanobis Center loss for better performance on adversarial robustness, although it adopted pre-defined constant values of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, which cannot be trained like our method.

For in-distribution classification, the softmax cross-entropy loss will be computed upon our newly proposed distance based score $h_i(\mathbf{z})$ instead of $f_i(\mathbf{z})$ in Eqn. (0.2):

$$\mathcal{L}_{\text{sce}}(\mathbf{x}, \mathbf{1}_y) = -\mathbf{1}_y^\top \log \left[\operatorname{softmax}[h(\mathbf{z})]\right]. \tag{0.12}$$

Compared with the simplified estimation in Eqn. (0.10), our design provides a complete version of posterior probability estimation and prevents false likelihood. Our loss functions discussed below will utilize the estimated likelihood.

A Probabilistic Loss Function

In the last part, we modify the top layer of deep networks to estimate the likelihood under the assumption that the features of the penultimate layer are approximately class-conditional Gaussian distribution. The maximum likelihood value across all classes, *i.e.*$\exp(\max_i h_i)$, can then be a good indicator to separate in-distribution and out-of-distribution features (Hendrycks et al. 2022). During training, we draw lessons from the classic Maximum Likelihood Estimation (MLE) of binary classification as the training loss:

$$\mathcal{L}_{\text{ice-ood}}(\widetilde{\mathbf{z}}) = -\log\left[1 - \exp[\max_i h_i(\widetilde{\mathbf{z}})]\right], \quad (0.13)$$

$$\mathcal{L}_{\text{ice-id}}(\mathbf{z}, y) = -\log\left[\exp[h_y(\mathbf{z})]\right] = -h_y(\mathbf{z}). \quad (0.14)$$

Eqn. (0.14) drags in-distribution feature $\mathbf{z}$ to class center and Eqn. (0.13) pushes outliers away. Their joint effect avoids distribution shift and results in the changing trend of features in Fig. 2. Note that, in Eqn. (0.14), instead of using the maximum likelihood $\exp(\max_i h_i)$, we opt for the likelihood for the ground-truth $\exp[h_y(\mathbf{z})]$ to avoid potential contradictory gradient. Finally, a weight parameter λ is utilized to balance in-distribution learning and out-of-distribution learning:

$$\textbf{ICE:}\ \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{in}}\mathcal{L}_{\text{ce}}(\mathbf{x}, \mathbf{1}_y) + \quad (0.15)$$
$$\lambda \cdot \left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{in}}\mathcal{L}_{\text{ice-id}}(\mathbf{x}, y) + \mathbb{E}_{\widetilde{\mathbf{x}}\sim\mathcal{D}_{out}^{oe}}\mathcal{L}_{\text{ice-ood}}(\widetilde{\mathbf{x}})\right].$$

A Versatile Solution in Practice

Besides solving the three in-distribution incompatibility issues, ICE also possesses other benefit: 1) The ICE detector $\exp[\max_i h_i(\widetilde{\mathbf{z}})] \in (0, 1]$ provides a direct estimation of detection confidence. Unlike previous methods, the learned maximum likelihood is explicitly trained by Eqn. (0.14) and Eqn. (0.13). In empirical evaluations, we will show that this learned confidence is well-calibrated. 2) The parameters of ICE are mostly learned through training, allowing us to easily add ICE into existing frameworks. Methods such as Mahalanobis (Lee et al. 2018) and ODIN (Liang, Li, and Srikant 2018) need a second round of learning procedure after standard training. 3) Methods like Energy use multiple hyper-parameters to threshold the learned energy score. These hyper-parameters need to be exhaustively cross-validated and cannot be shared across different sets. In contrast, the only hyper-parameter of ICE is the λ in Eqn. (0.15).

Experiment

In this section, we verify the effectiveness of ICE on a wide range of OOD evaluation benchmarks.

Experimental Setup

Dataset. For the in-distribution set, we choose the standard CIFAR10 and CIFAR100 (Krizhevsky 2012) as our major verification target. For out-of-distribution set, we adopt several commonly-used benchmarks, including Textures (Cimpoi et al. 2014), SVHN (Netzer et al. 2011), Places365 (Zhou et al. 2018), LSUN (Yu et al. 2015), and iSUN (Xu et al. 2015). We also use CIFAR100 as an OOD source to evaluate models learned on CIFAR10 and vice versa. We choose the 80 Million Tiny Images (Torralba, Fergus, and Freeman 2008) as the $\mathcal{D}_{out}^{oe}$.

Training. Following Hendrycks, Mazeika, and Dietterich (2019), we test two training protocols. Fine-tune: We initialize the model with a pre-trained checkpoint on the in-distribution set and then fine-tune the model with ten epochs. We adopt a cosine decay learning rate schedule with the initial value of 0.01. From-scratch: We train deep networks with both $\mathcal{D}_{in}^{train}$ and $\mathcal{D}_{out}^{oe}$ for 100 epochs. The initial learning rate is set to 0.1 with the commonly-used stair-wise decay learning rate schedule. For both protocols, the batch size for $\mathcal{D}_{in}^{train}$ is set to 128, and $\mathcal{D}_{out}^{oe}$ to 256.

Evaluation Metric. We adopt three mostly commonly-used OOD detection metrics for evaluation: the area under the precision-recall curve (AUPR), the area under the receiver operating characteristic curve (AUROC), and the false positive rate at 95% true positive rate (FPR95).

Outlier Detection Performance

We present our main results in Table. 1. For each experimental setup, we compare our ICE with OE (Hendrycks, Mazeika, and Dietterich 2019) and Energy (Liu et al. 2020). Our ICE outperforms the other two counterparts on most benchmarks. For CIFAR10 as $\mathcal{D}_{in}$, we first present results on the model of WideResNet-34-10. Both fine-tune and from-scratch schedules achieve promising results for the three algorithms, while the from-scratch schedule slightly but consistently outperforms fine-tune schedule. This aligns with the observations by Hendrycks, Mazeika, and Dietterich (2019). To verify the extensibility of our method, we also present results on the DenseNet-40-12 model with the from-scratch schedule. Again, ICE exhibits generally better detection capability. Note that, for most choices of $\mathcal{D}_{out}^{test}$, all the three methods have achieved very high scores, leaving only very little room for improvement. In contrast, for the hardest case where CIFAR100 serves as $\mathcal{D}_{out}^{test}$, the improvement brought by ICE becomes more significant. For the WideResNet model learned by fine-tune schedule, ICE achieves 95.08% AUPR, 94.90% AUROC, and 23.23% FPR95, with the improvement of $2.37\%/1.86\%/7.19\%$ over OE. For CIFAR100 as $\mathcal{D}_{in}$, we test WideResNet with the from-scratch schedule. Again, ICE achieves state-of-the-art results. In the case of SVHN being $\mathcal{D}_{out}^{test}$, ICE achieves 98.97% AUPR, 94.71% AUROC, and 22.41% FPR95.

In-Distribution Accuracy

ICE is designed for better in-distribution compatibility. Therefore, in this part, we compare its in-distribution accuracy with other baselines, including those without training on outliers. In Table. 2, we show the average detection scores across iSUN, Places365, Textures, LSUN, and SVHN. OE and Energy with the from-scratch schedule achieve superior detection performance and greatly improve detection performance over non-training detection algorithms such as ODIN and Mahalanobis but degrade the in-distribution accuracy. OE and Energy with fine-tune schedules achieve relatively better in-distribution accuracy than non-training detection algorithms. The small learning rate of fine-tune schedule

				AUPR ↑			AUROC ↑			FPR95 ↓		
$\mathcal{D}_{in}$	Model	Training	$\mathcal{D}_{out}^{test}$	OE	Energy	ICE(ours)	OE	Energy	ICE(ours)	OE	Energy	ICE(ours)
CIFAR10	WideResNet	Fine-tune	iSUN	98.79	98.92	**99.31**	99.15	99.29	**99.45**	3.85	2.65	**2.27**
			Places365	99.98	99.99	**99.99**	96.57	**97.44**	97.16	15.96	**10.06**	11.73
			Textures	96.42	97.47	**98.53**	97.96	98.58	**99.15**	10.27	5.31	**4.37**
			LSUN	99.59	98.96	**99.64**	99.61	99.29	**99.67**	1.81	2.53	**1.46**
			SVHN	99.73	99.85	**99.89**	98.75	99.30	**99.40**	3.84	2.19	**1.94**
			CIFAR100	92.71	93.86	**95.08**	93.04	93.81	**94.90**	30.42	28.24	**23.23**
		From-scratch	iSUN	99.03	99.11	**99.32**	99.36	99.40	**99.45**	2.29	2.31	**2.30**
			Places365	99.99	99.99	**99.99**	97.48	97.58	**97.68**	10.06	**8.22**	9.97
			Textures	98.39	98.44	**99.07**	99.08	99.25	**99.51**	3.55	2.49	**1.96**
			LSUN	99.32	99.15	**99.47**	99.45	99.50	**99.62**	2.11	1.25	**1.20**
			SVHN	99.91	**99.95**	99.91	99.48	**99.76**	99.58	2.15	**0.79**	1.26
			CIFAR100	94.95	94.65	**95.38**	94.79	94.44	**95.13**	27.10	26.15	**22.36**
	DenseNet	From-scratch	iSUN	97.05	98.61	**98.93**	98.14	99.14	**99.15**	7.09	3.54	**3.52**
			Places365	99.98	99.98	**99.98**	96.19	96.14	**96.21**	**15.24**	18.15	20.25
			Textures	96.40	**97.89**	97.62	98.14	**98.91**	98.46	8.36	**4.02**	9.05
			LSUN	98.32	98.10	**98.97**	98.70	98.88	**99.04**	5.11	3.59	**3.79**
			SVHN	99.44	99.83	**99.89**	97.66	99.24	**99.28**	7.10	3.01	**2.98**
			CIFAR100	91.18	90.91	**91.90**	91.40	90.37	**91.51**	41.13	44.80	**40.44**
CIFAR100	WideResNet	From-scratch	iSUN	83.61	**89.39**	85.34	88.40	**91.70**	88.45	36.55	**29.88**	39.05
			Places365	99.91	99.92	**99.92**	87.25	87.62	**87.55**	**44.89**	45.42	49.00
			Textures	78.32	79.92	**81.15**	88.23	88.41	**89.68**	39.84	43.57	**38.95**
			LSUN	**94.11**	89.51	92.87	94.02	92.53	**94.15**	**15.35**	26.84	25.37
			SVHN	98.92	98.71	**98.97**	94.49	93.41	**94.71**	22.57	23.65	**22.41**

Table 1: OOD detection Performance on various benchmarks, including different choices of models, training schedules, in-distribution sets ($\mathcal{D}_{in}$), and out-of-distribution sets for testing ($\mathcal{D}_{out}^{test}$). We compare our ICE with OE and Energy. Three detection metrics (AUPR/AUROC/FPR95) are evaluated, where ICE consistently performs better. ↓ indicates that, as the metric decreases, the performance improves. ↑ indicates that, as the metric increases, the performance becomes better.

prevents the degradation of in-distribution accuracy. However, the fine-tune schedule cannot reach the same detection performance level as the from-scratch schedule.

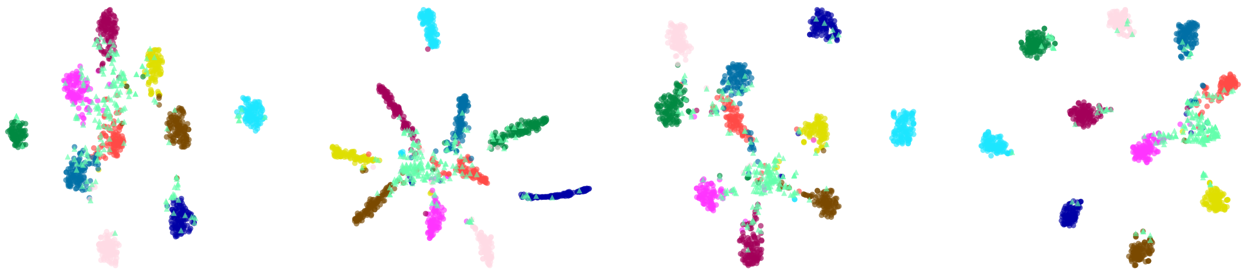
In contrast, ICE accomplishes evident higher accuracy than other baselines, either under fine-tune or from-scratch schedules. The objective of ICE effectively utilizes the auxiliary examples from $\mathcal{D}_{out}^{oe}$ to provide neural network with new information and thus help its generalization. The refined in-distribution compatibility concentrates in-distribution features around the class center, endows in-distribution classification with high confidence, and thus further helps detect outliers with low confidence. Unlike OE and Energy, ICE achieves better in-distribution accuracy with a from-scratch schedule than fine-tune schedule, indicating its intrinsic built-in in-distribution compatibility.

$\mathcal{D}_{in}$	Method	AUPR ↑	AUROC ↑	FPR95 ↓	In-dist Accuracy ↑
CIFAR -10	MSP	97.88	90.82	56.03	95.11
	ODIN	97.39	90.39	37.53	95.11
	Mahalanobis	98.47	93.27	35.97	95.11
	OE†	98.90	98.40	7.14	95.23
	Energy†	99.03	98.78	4.54	95.88
	ICE(ours)†	99.47	98.96	4.35	96.02
	OE‡	99.32	98.97	4.03	94.88
	Energy‡	99.32	99.10	4.12	94.13
	ICE(ours)‡	**99.55**	**99.16**	**3.33**	**96.38**
CIFAR -100	MSP	93.90	75.56	80.01	76.01
	ODIN	93.94	76.55	75.17	76.01
	Mahalanobis	95.22	81.74	60.60	76.01
	OE‡	90.97	90.47	**31.84**	75.82
	Energy‡	91.49	90.73	33.87	75.75
	ICE(ours)‡	**91.65**	**90.91**	34.95	**77.74**

Table 2: The OOD detection performance and in-distribution accuracy for different methods. † represents models trained with fine-tune schedule. ‡ represents models trained with the from-scratch schedule.

Feature Distribution

To get a straightforward insight into how different outlier training methods influence the distribution, we visualize the features generated by the penultimate layer of deep networks, *i.e.*$\mathbf{z}$ and $\widetilde{\mathbf{z}}$, with the t-SNE algorithm. In our visualization, the t-SNE mapping is first learned on all in-distribution samples. Then, we use the learned mapping to transform outliers. As shown in Fig. 4, the Plain Training method without any learning on outliers exhibit poor OOD feature distribution. The OOD features are heavily mixed with in-distribution ones. Standard Outlier Exposure squeezes OOD features in the space center and pushes in-distribution ones into diverse directions. This aligns our visualization on a 2-dimension case in Fig. 3. Due to the effect of OE, the original clustered in-distribution features are transformed into slender ones. Energy maintains the cluster shape of in-distribution features. However, like the OE algorithm, the outlier features are close to in-distribution ones, which increases the difficulty of discriminating them. ICE, on the other hand, resumes the most characteristics of in-distribution features in Plain Training. The in-distribution clusters are tightly bounded and scatter from each other. Meanwhile, outliers are mostly constrained in a local space.



(a) Plain Training.

(b) Outlier Exposure.

(c) Energy Score.

(d) ICE (ours).

Figure 4: Feature visualization with t-SNE algorithm. We present three OE methods and Plain Training. ICE learns compact and discriminating features. ▲ represents OOD examples (CIFAR100), and the rest examples are in-distribution (CIFAR10).

Loss Function	Model	AUPR ↑	AUROC ↑	FPR95 ↓	In-dist Accuracy ↑
Energy	WRN	99.32	99.10	4.12	94.13
Energy	WRN(G)	99.37	99.14	4.01	93.95
BCE	WRN	99.40	99.02	4.48	95.81
ICE^{-}	WRN(G)	83.62	74.03	35.58	90.44
ICE	WRN(G)	**99.55**	**99.16**	**3.33**	**96.38**

Table 3: Ablation studies on various components of ICE.

Ablation Study

In this part, we conduct ablation studies on each component of ICE. First of all, ICE modifies the top design of WideResNet. We dub this modified model as WideResNet(G). To evaluate the effect of this design, we test Energy algorithm on WideResNet(G), which addresses the false likelihood problem but still has the issue of contradictory gradient. As shown in Table 3, Energy method with WideResNet(G) perform slightly better than the baseline with WideResNet. Then, the second counterpart we want to compare is applying the Sigmoid Binary Cross-Entropy (BCE) loss onto WideResNet. Such design avoids the contradictory gradient issue yet still has false likelihood and distribution shift problems. The improvement is still observable but limited. We also test eliminating the $\mathcal{L}_{\text{energy-id}}$ in Eqn. (0.15) (ICE^{-}), which prevents contradictory gradient and false likelihood but not distribution shift. The detection performance is heavily degraded, indicating the importance of balancing $\mathcal{L}_{\text{energy-ood}}$ with $\mathcal{L}_{\text{energy-ood}}$ in Eqn. (0.15). The above ablation studies show that the individual improvement of each component is not as significant as the total improvement. The superior performance of ICE is achieved by the collective effect of all components.

Confidence Estimation

As discussed before, unlike OE and Energy, our method provides the capability of estimating the $(0, 1]$ probability of whether the upcoming input is in-distribution or not. Here we present a direct visualization of our in-distribution confidence indicator of $\exp[\max_i h_i(\widetilde{\mathbf{z}})]$. In Fig. 5, we plot the distribution of $\exp[\max_i h_i(\mathbf{z})]$ (CIFAR10) and $\exp[\max_i h_i(\widetilde{\mathbf{z}})]$ (CIFAR100) against the training epochs. As the training progresses, the two distributions gradually separate from each other and eventually concentrate on their ground-truth value. Moreover, the two distributions both formulate a typical pattern of long-tail distribution, indicating that the majority of the samples are trained to their desirable category.

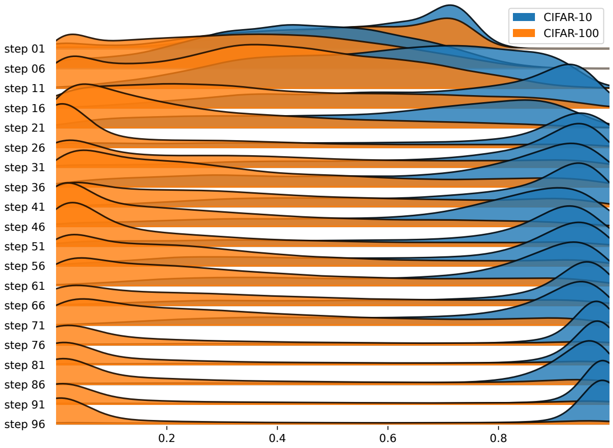


Figure 5: On each training epoch, we plot the distribution of detection confidence for in-distribution set (CIFAR10) and out-of-distribution set (CIFAR100).

Conclusion

In this paper, we analyzed existing Outlier Exposure methods and demonstrated that those methods are detrimental to in-distribution accuracy due to three factors: *contradictory gradient*, *false likelihood*, and *distribution shift*. We thereafter proposed a novel Outlier Exposure method, namely ICE, to address the three defects. ICE replaces a conventional linear discriminator with a Gaussian-like discriminator to prevent false likelihood. Then, the likelihood score generated by the Gaussian-like discriminator is trained via a loss function, enlightened by the classic MLE solution for binary classification. The loss function can prevent distribution shift and does not yield contradictory gradients. ICE achieves *sota* results on multiple OOD detection benchmarks through the collective effects of the above-designed components. Meanwhile, ICE improves in-distribution accuracy by learning from additional OOD features.

Acknowledgments

This work was supported in part by The National Key Research and Development Program of China (Grant Nos: 2018AAA0101400), in part by The National Nature Science Foundation of China (Grant Nos: 62036009, U1909203, 61936006), in part by Innovation Capability Support Program of Shaanxi (Program No. 2021TD-05). Part of this work was supported by the ZJU-Tencent Joint Project.

References

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing Textures in the Wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 3606–3613. IEEE Computer Society.

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 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 41–50. Computer Vision Foundation / IEEE.

Hendrycks, D.; Basart, S.; Mazeika, M.; Zou, A.; Kwon, J.; Mostajabi, M.; Steinhardt, J.; and Song, D. 2022. Scaling Out-of-Distribution Detection for Real-World Settings. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 8759–8773. PMLR.

Hendrycks, D.; and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. G. 2019. Deep Anomaly Detection with Outlier Exposure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Hsu, Y.; Shen, Y.; Jin, H.; and Kira, Z. 2020. Generalized ODIN: Detecting Out-of-Distribution Image Without Learning From Out-of-Distribution Data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 10948–10957. Computer Vision Foundation / IEEE.

Huang, R.; and Li, Y. 2021. MOS: Towards Scaling Out-of-Distribution Detection for Large Semantic Space. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 8710–8719. Computer Vision Foundation / IEEE.

Krizhevsky, A. 2012. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 7167–7177.

Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Lin, Z.; Roy, S. D.; and Li, Y. 2021. MOOD: Multi-Level Out-of-Distribution Detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 15313–15323. Computer Vision Foundation / IEEE.

Liu, W.; Wang, X.; Owens, J. D.; and Li, Y. 2020. Energy-based Out-of-distribution Detection. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. *NIPS*.

Pang, T.; Xu, K.; Dong, Y.; Du, C.; Chen, N.; and Zhu, J. 2020. Rethinking Softmax Cross-Entropy Loss for Adversarial Robustness. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ren, J.; Liu, P. J.; Fertig, E.; Snoek, J.; Poplin, R.; DePristo, M. A.; Dillon, J. V.; and Lakshminarayanan, B. 2019. Likelihood Ratios for Out-of-Distribution Detection. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 14680–14691.

Sun, Y.; Guo, C.; and Li, Y. 2021. ReAct: Out-of-distribution Detection With Rectified Activations. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 144–157.

Torralba, A.; Fergus, R.; and Freeman, W. T. 2008. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11): 1958–1970.

Wang, H.; Li, Z.; Feng, L.; and Zhang, W. 2022. ViM: Out-Of-Distribution with Virtual-logit Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wei, H.; Xie, R.; Cheng, H.; Feng, L.; An, B.; and Li, Y. 2022. Mitigating Neural Network Overconfidence with Logit Normalization. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-*

23 July 2022, Baltimore, Maryland, USA, volume 162 of *Proceedings of Machine Learning Research*, 23631–23644. PMLR.

Xie, S. M.; Kumar, A.; Jones, R.; Khani, F.; Ma, T.; and Liang, P. 2021. In-N-Out: Pre-Training and Self-Training using Auxiliary Information for Out-of-Distribution Robustness. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xu, P.; Ehinger, K. A.; Zhang, Y.; Finkelstein, A.; Kulkarni, S. R.; and Xiao, J. 2015. TurkerGaze: Crowdsourcing Saliency with Webcam based Eye Tracking. *CoRR*, abs/1504.06755.

Yang, J.; Wang, H.; Feng, L.; Yan, X.; Zheng, H.; Zhang, W.; and Liu, Z. 2021. Semantically Coherent Out-of-Distribution Detection. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 8281–8289. IEEE.

Yu, F.; Zhang, Y.; Song, S.; Seff, A.; and Xiao, J. 2015. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *CoRR*, abs/1506.03365.

Zhou, B.; Lapedriza, À.; Khosla, A.; Oliva, A.; and Torralba, A. 2018. Places: A 10 Million Image Database for Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6): 1452–1464.