OpenMix: Exploring Outlier Samples for Misclassification Detection

Fei Zhu^{1,2}, Zhen Cheng^{1,2}, Xu-Yao Zhang^{1,2*}, Cheng-Lin Liu^{1,2}
^{1}MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
^{2}School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China
{zhufei2018, chengzhen2019}@ia.ac.cn, {xyz, liucl}@nlpr.ia.ac.cn

Abstract

Reliable confidence estimation for deep neural classifiers is a challenging yet fundamental requirement in high-stakes applications. Unfortunately, modern deep neural networks are often overconfident for their erroneous predictions. In this work, we exploit the easily available outlier samples, i.e., unlabeled samples coming from non-target classes, for helping detect misclassification errors. Particularly, we find that the well-known Outlier Exposure, which is powerful in detecting out-of-distribution (OOD) samples from unknown classes, does not provide any gain in identifying misclassification errors. Based on these observations, we propose a novel method called OpenMix, which incorporates open-world knowledge by learning to reject uncertain pseudo-samples generated via outlier transformation. OpenMix significantly improves confidence reliability under various scenarios, establishing a strong and unified framework for detecting both misclassified samples from known classes and OOD samples from unknown classes. The code is publicly available at `https://github.com/Impression2805/OpenMix`.

1. Introduction

Human beings inevitably make mistakes, so do machine learning systems. Wrong predictions or decisions can cause various problems and harms, from financial loss to injury and death. Therefore, in risk-sensitive applications such as clinical decision making [14] and autonomous driving [29, 63], it is important to provide reliable confidence to avoid using wrong predictions, in particular for non-specialists who may trust the computational models without further checks. For instance, a disease diagnosis model should hand over the input to human experts when the prediction confidence is low. However, though deep neural networks (DNNs) have enabled breakthroughs in many fields, they are known to be overconfident for their erroneous predictions [25, 62], *i.e.*, assigning high confidence for ① misclassified samples from in-distribution (ID) and ② out-of-distribution (OOD) samples from unknown classes.

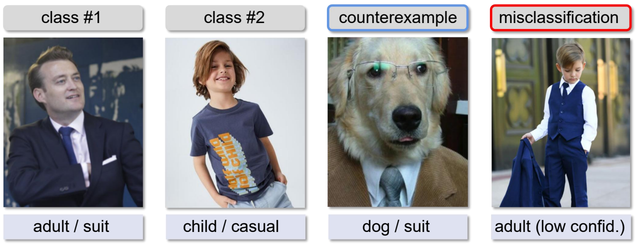

Figure 1. Illustration of advantages of counterexample data for reliable confidence estimation. The misclassified image has the most determinative and shortcut [18] features from class #1 (*i.e.*, suit). Counterexample teaches the model the knowledge of *what is not adult even if it has suit*, which could help reduce model's confidence on wrong predictions.

In recent years, many efforts have been made to enhance the OOD detection ability of DNNs [2, 13, 15, 23, 26, 39], while little attention has been paid to detecting misclassified errors from known classes. Compared with the widely studied OOD detection problem, misclassification detection (MisD) is more challenging because DNNs are typically more confident for the misclassified ID samples than that for OOD data from a different distribution [19]. In this paper, we focus on the under-explored MisD, and propose a simple approach to help decide whether a prediction is likely to be misclassified, and therefore should be rejected.

Towards developing reliable models for detecting misclassification errors, we start by asking a natural question:

Why are human beings good at confidence estimation?

A crucial point is that humans learn and predict in context, where we have abundant prior knowledge about other entities in the open world. According to *mental models* [11, 31, 54] in cognitive science, when assessing the validity or evidence of a prediction, one would retrieve counterexamples, *i.e.*, which satisfy the premise but cannot lead to the conclusion. In other words, exploring counterexamples from open world plays an important role in establishing reliable confidence for the reasoning problem. Inspired by

*Corresponding author.

this, we attempt to equip DNNs with the above ability so that they can reduce confidence for incorrect predictions. Specifically, we propose to leverage outlier data, *i.e.*, unlabeled random samples from non-target classes, as counterexamples for overconfidence mitigation. Fig. 1 presents an intuitive example to illustrate the advantages of outlier samples for reducing the confidence of misclassification.

To leverage outlier samples for MisD, we investigate the well-known Outlier Exposure (OE) [26] as it is extremely popular and can achieve state-of-the-art OOD detection performance. However, we find that OE is more of a hindrance than a help for identifying misclassified errors. Further comprehensive experiments show that existing popular OOD detection methods can easily ruin the MisD performance. This is undesirable as misclassified errors widely exist in practice, and a model should be able to reliably reject those samples rather than only reject OOD samples from new classes. We observe that the primary reason for the poor MisD performance of OE and other OOD methods is that: they often compress the confidence region of ID samples in order to distinguish them from OOD samples. Therefore, it becomes difficult for the model to further distinguish correct samples from misclassified ones.

We propose a *learning to reject* framework to leverage outlier data. ① Firstly, unlike OE and its variants which force the model to output a uniform distribution on all training classes for each outlier sample, we explicitly break the closed-world classifier by adding a separate reject class for outlier samples. ② To reduce the distribution gap between ID and open-world outlier samples, we mix them via simple linear interpolation and assign soft labels for the mixed samples. We call this method *OpenMix*. Intuitively, the proposed OpenMix can introduce the prior knowledge about *what is uncertain and should be assigned low confidence*. We provide proper justifications and show that OpenMix can significantly improve the MisD performance. We would like to highlight that our approach is simple, agnostic to the network architecture, and does not degrade accuracy when improving confidence reliability.

In summary, our primary contributions are as follows:

- For the first time, we propose to explore the effectiveness of outlier samples for detecting misclassification errors. We find that OE and other OOD methods are useless or harmful for MisD.
- We propose a simple yet effective method named OpenMix, which can significantly improve MisD performance with enlarged confidence separability between correct and misclassified samples.
- Extensive experiments demonstrate that OpenMix significantly and consistently improves MisD. Besides, it also yields strong OOD detection performance, serving as a unified failure detection method.

2. Related Work

Misclassification detection. Chow [7] presented an optimal rejection rule for Bayes classifier. For DNNs, a common baseline of MisD is the maximum softmax probability (MSP) score [25]. Some works [8, 41] introduce a separate confidence network to perform binary discrimination between correct and misclassified training samples. One clear drawback of those methods is that DNNs often have high training accuracy where few or even no misclassified examples exist in the training set. Moon *et al.* [43] proposed to learn an ordinal ranking relationship according to confidence for reflecting the historical correct rate during training dynamics. A recent work [64] demonstrates that calibration methods [20, 44, 45, 52] are harmful for MisD, and then reveals a surprising and intriguing phenomenon termed as ***reliable overfitting***: the model starts to irreversibly lose confidence reliability after training for a period, even the test accuracy continually increases. To improve MisD, a simple approach, *i.e.* FMFP [64] was designed by eliminating the reliable overfitting phenomenon. A concurrent work [65] develops *classAug* for reliable confidence estimation by learning more synthetic classes.

Utilizing outlier samples. Auxiliary outlier dataset is commonly utilized in many problem settings. For example, Lee *et al.* [37] leveraged outliers to enhance adversarial robustness of DNNs. Park *et al.* [47] used outliers to improve object localization performance. ODNL [56] uses open-set outliers to prevent the model from over-fitting inherent noisy labels. In addition, outlier samples are also effective for improving few-shot learning [35] and long-tailed classification [57]. In the area of confidence estimation, OE [26] has been the most popular and effective way to improve OOD detection ability by using outlier samples.

OOD detection. This task focuses on judging whether an input sample is from novel classes or training classes. Compared with MisD, OOD detection has been studied extensively in recent years and various methods have been developed, including training-time [4,26,51,55,58] and post-hoc strategies [13, 23, 36, 38, 39]. Cheng *et al.* [6] proposed a AoP (Average of Pruning) framework to improve the performance and stability of OOD detection, which also offers notable gain for MisD. Most existing OOD detection works do not involve detecting misclassified errors. We would like to highlight that both OOD and misclassified samples are failure sources and should be rejected together.

3. Problem Setting and Motivation

3.1. Preliminaries: MisD and OE

Basic notations. Let $\mathcal{X} \in \mathbb{R}^d$ denote the input space and $\mathcal{Y} = \{1, 2, ..., k\}$ represents the label space. Given a sample $(\boldsymbol{x}, y)$ drawn from an unknown distribution $\mathcal{P}$ on $\mathcal{X} \times \mathcal{Y}$,

Table 1. MisD performance can not be improved with OE. AUROC and FPR95 are percentage. AURC is multiplied by 10^3.

Dateset	Method	AURC ↓			AUROC ↑			FPR95 ↓		
		ResNet110	WRNet	DenseNet	ResNet110	WRNet	DenseNet	ResNet110	WRNet	DenseNet
CIFAR-10	MSP [25]	**9.52±0.49**	**4.76±0.62**	**5.66±0.45**	**90.13±0.46**	**93.14±0.38**	**93.14±0.65**	**43.33±0.59**	**30.15±1.98**	**38.64±4.70**
	+ OE [26]	10.10±0.54	4.83±0.13	8.23±0.95	90.02±0.36	93.09±0.15	91.44±0.15	46.89±1.78	38.78±2.59	45.86±2.30
CIFAR-100	MSP [25]	**89.05±1.39**	**46.84±0.90**	**66.11±1.56**	**84.91±0.13**	**88.50±0.44**	**86.20±0.04**	**65.65±1.72**	**56.64±1.33**	**62.79±0.83**
	+ OE [26]	103.06±2.50	58.05±1.21	86.96±2.27	83.81±0.49	86.36±0.20	84.25±0.50	71.11±0.77	62.96±0.38	70.39±0.65

a neural network classifier $f(\cdot) : \mathbb{R}^d \rightarrow \Delta_k$ produces a probability distribution for $\boldsymbol{x}$ on k classes, where Δ_k denotes the $k-1$ dimensional simplex. Specifically, $f_i(\boldsymbol{x})$ denotes the i-th element of the softmax output vector produced by f. Then $\hat{y} =: \arg\max_{y \in \mathcal{Y}} f_y(\boldsymbol{x})$ can be returned as the predicted class and the associated probability $\hat{p} =: \max_{y \in \mathcal{Y}} f_y(\boldsymbol{x})$ can be viewed as the predicted confidence. Denote by $\mathcal{D}_{\text{in}}$ the distribution over $\mathcal{X}$ of ID data. Besides, we can also have access to some unlabeled outlier samples (*i.e.*, $\mathcal{D}_{\text{out}}$) coming from outside target classes. At inference time, most of the inputs are from known classes, and they can be correctly classified or misclassified. We use $\mathcal{D}_{\text{in}}^{\text{test},\checkmark}$ and $\mathcal{D}_{\text{in}}^{\text{test},\times}$ to represent the distribution of correct and misclassified ID samples, respectively.

Misclassification detection. MisD, also known as failure prediction [8,64], is a critical safeguard for safely deploying machine learning models in real-world applications. It focuses on detecting and filtering wrong predictions ($\mathcal{D}_{\text{in}}^{\text{test},\times}$) from correct predictions ($\mathcal{D}_{\text{in}}^{\text{test},\checkmark}$) based on their confidence ranking. Formally, denote κ a confidence-rate function (*e.g.*, the MSP or negative entropy) that assesses the degree of confidence of the predictions, with a predefined threshold $\delta \in \mathbb{R}^+$, the misclassified samples can be detected based on a decision function g such that for a given input $\boldsymbol{x}_i \in \mathcal{X}$:

$$g(\boldsymbol{x}_i) = \begin{cases} \text{correct} & \text{if } \kappa(\boldsymbol{x}_i) \geq \delta, \\ \text{misclassified} & \text{otherwise.} \end{cases} \tag{1}$$

Outlier Exposure. OE [26] leverages auxiliary outliers to help the model detect OOD inputs by assigning low confidence for samples in $\mathcal{D}_{\text{out}}$. Specifically, given a model f and the original learning objective ℓ_{CE} (*i.e.*, cross-entropy loss), OE minimizes the following objective:

$$\mathbb{E}_{\mathcal{D}_{\text{in}}^{\text{train}}}[\ell_{\text{CE}}(f(\boldsymbol{x}), y)] + \lambda\, \mathbb{E}_{\mathcal{D}_{\text{out}}}[\ell_{\text{OE}}(f(\widetilde{\boldsymbol{x}}))], \tag{2}$$

where $\lambda > 0$ is a penalty hyper-parameter, and ℓ_{OE} is defined by Kullback-Leibler (KL) divergence to the uniform distribution: $\ell_{\text{OE}}(f(\boldsymbol{x})) = \text{KL}(\mathcal{U}(y) \| f(\boldsymbol{x}))$, in which $\mathcal{U}(\cdot)$ denotes the uniform distribution. Basically, OE uses the available OOD data $\mathcal{D}_{\text{out}}$ to represent the real OOD data that would be encountered in open environments. Although the limited samples in $\mathcal{D}_{\text{out}}$ can not fully reveal the real-world OOD data, OE surprisingly yields strong performance in OOD detection. The strong effectiveness of outliers for improving OOD detection has been verified by many recent works [39, 40]. This leads us to ask: *Can we use outlier data to help detect misclassification errors?*

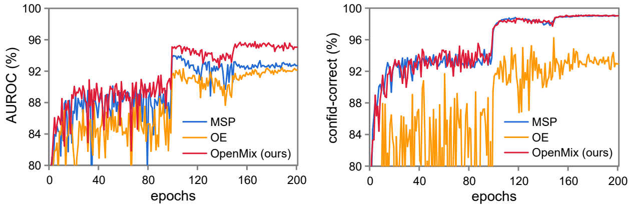

Figure 2. The AUROC and averaged confidence of correct samples under different training epochs. OE results in (left) worse AUROC with (right) under-confident correctly classified samples.

3.2. Motivation: understanding the effect of OE

We start with the empirical experiments of OE, analyzing the role of outlier data for MisD. Throughout this subsection, we perform experiments on CIFAR [34] using standard cross-entropy loss and OE based training, respectively. We use 300K `RandImages` as the OOD auxiliary dataset following [26, 56, 57]. Specifically, all images that belong to CIFAR classes are removed in `RandImages` so that $\mathcal{D}_{\text{in}}$ and $\mathcal{D}_{\text{out}}$ are disjoint. Evaluation metrics include AURC ↓ [17], FPR95 ↓ and AUROC ↑ [10].

OE has negative impact on MisD. Table 1 presents the results of training without/with the auxiliary outlier dataset. We can observe that OE consistently deteriorates the MisD performance under various metrics. For example, when training with OE on CIFAR-10/WRNet, the FPR95↓ increases 8.63 percentages compared with baseline, *i.e.*, MSP. In Fig. 2 (left), we can observe that the AUROC of OE is consistently lower than that of baseline method during training of WRNet on CIFAR-10. Intuitively, to distinguish correct predictions from errors, the model should assign high confidence for correct samples, and low confidence for errors. However, in Fig. 2 (right), we find that OE can significantly deteriorate the confidence of correct samples, which makes it difficult to separate correct and wrong predictions.

Understanding from feature space uniformity. Overconfidence for misclassified prediction implies that the sample is projected into the density region of a wrong class [64]. Intuitively, excessive feature compression would lead to over-tight class distribution, increasing the overlap between correct and misclassified samples. To better understand the negative effect of OE for MisD, we study its impact on the learned deep feature space. Let $z(\cdot)$ represent the feature extractor, we then define and compute the inter-class distances $\pi_{inter} = \frac{1}{Z_{inter}} \sum_{y_l, y_k, l \neq k} d(\boldsymbol{\mu}(Z_{y_l}), \boldsymbol{\mu}(Z_{y_k}))$, and average intra-class distances $\pi_{intra} = \frac{1}{Z_{intra}} \sum_{y_l \in y} \sum_{\boldsymbol{z}_i, \boldsymbol{z}_j \in Z_{y_l}, i \neq j} d(\boldsymbol{z}_i, \boldsymbol{z}_j)$, in which $d(\cdot; \cdot)$ is the distance function. $Z_{y_l} = \{\boldsymbol{z}_i := z(\boldsymbol{x}_i) | y_i = y_l\}$ denotes the set of deep feature vectors of samples in class y_l. $\boldsymbol{\mu}(Z_{y_l})$ is the class mean. Z_{intra} and Z_{inter} are two normalization constants. Finally, the feature space uniformity (FSU) is defined as $\pi_{fsu} = \pi_{intra}/\pi_{inter}$ [49]. Intuitively, large FSU increases the instances in low density regions and encourages the learned features to distribute uniformly (maximal-info preserving) in feature space.

When facing OOD samples from new classes, small FSU (larger inter-class distance and small intra-class distance) could result in less overlap between ID and OOD samples. However, compared to OOD data, misclassified samples are ID and distributed much closer to correct samples of each class. As shown in Fig. 3, the FSU is reduced with OE. By forcing the outliers to be uniformly distributed over original classes, OE introduces similar effect as label-smoothing [45], which leads to over-compressed distributions, losing the important information about the hardness of samples. Consequently, ID samples of each class would be distributed within a compact and over-tight region, making it harder to separate misclassified samples from correct ones. Supp.M provides a unified view on the connection between FSU and OOD detection, MisD performance.

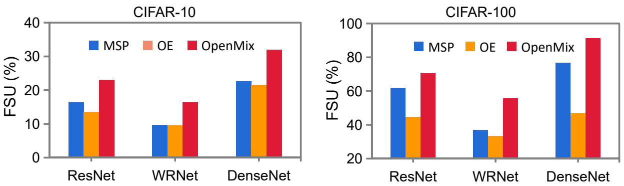

Figure 3. The impact of OE on the deep feature space. With OE, the feature space uniformity (FSU) is reduced, which indicates excessive feature compression and has negative influence for MisD. Our OpenMix leads to less compact feature distributions.

How to use outliers for MisD? Based on the above observations and analysis, we argue that the original OE [26] should be modified from two aspects for MisD:

- *On learning objective.* Simply forcing the model to yield uniform distribution for outliers with ℓ_{OE} would lead to reduced feature space uniformity and worse MisD performance. We suggest that the original ℓ_{OE} loss should be discarded, and a new learning objective to use outliers should be designed.
- *On outlier data.* Outliers from unknown classes are OOD samples and have a large distribution gap with ID misclassified samples, which could weaken the effect for MisD. To overcome this issue, we suggest transforming available outlier data into new outliers that are distributed closer to ID misclassified samples.

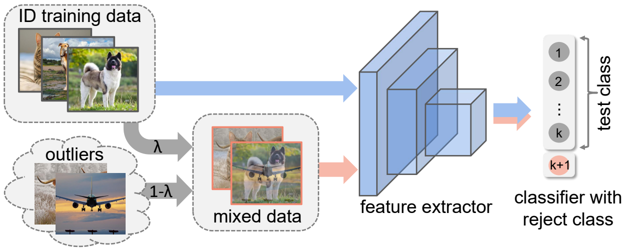

Figure 4. The pipeline of OpenMix.

Motivated by the above observations and analysis, we propose to modify OE from the perspective of learning objective and outlier data, respectively.

4. Proposed Method: OpenMix

Learning with reject class. Different from OE that forces the model to output uniform distribution, we propose to predict the outliers as an additional reject class. Specifically, for a k-class classification problem, we extend the label space by explicitly adding a separate class for outlier samples. Formally, denote $\mathbb{I}^{y_i} := (0, ..., 1, .., 0)^\top \in \{0, 1\}^{k+1}$ is a one-hot vector and only the y_i-th entry is 1. For the outlier dataset, we map the samples to the $(k+1)$-class. The learning objective is:

$$\mathcal{L}_{\text{total}} = \mathbb{E}_{\mathcal{D}_{\text{in}}^{\text{train}}}[\ell(f(\boldsymbol{x}), y)] + \gamma \mathbb{E}_{\mathcal{D}_{\text{out}}}[\ell(f(\widetilde{\boldsymbol{x}}), \widetilde{y})], \quad (3)$$

where $\widetilde{y} = k+1$ and γ denotes a hyper-parameter. With reject class, the negative effect of outliers for MisD could be alleviated. However, there is little performance gain compared with baseline method, as will be shown in Sec. 5.2. Intuitively, the best auxiliary samples are the misclassified examples. However, OOD outliers can not represent misclassified ID samples well due to the distribution gap.

Outlier transformation via Mixup. The distribution gap existing between misclassified ID samples and the OOD outliers significantly limits the effectiveness of learning with reject class. To address this issue, we propose a simple yet powerful strategy to shrink the gap by transforming the original outliers to be near the ID distribution. Specifically, inspired by the well-known Mixup technique [61], we perform simple linear interpolation between ID training samples and OOD outliers. Formally, Given a pair of examples $(\boldsymbol{x}, y)$ and $(\widetilde{\boldsymbol{x}}, \widetilde{y})$ respectively sampled from the ID training set and outlier data, we apply linear interpolation to produce transformed outlier $(\breve{\boldsymbol{x}}, \breve{y})$ as follows:

$$\breve{\boldsymbol{x}} = \lambda \boldsymbol{x} + (1 - \lambda)\widetilde{\boldsymbol{x}}, \quad \mathbb{I}^{\breve{y}} = \lambda \mathbb{I}^{y} + (1 - \lambda)\mathbb{I}^{\widetilde{y}}. \quad (4)$$

The $\lambda \in [0, 1]$ is a parameter sampled as $\lambda \sim \text{Beta}(\alpha, \alpha)$ for $\alpha \in (0, \infty)$. $y \in \{1, ..., k\}$, $\widetilde{y} = k+1$ and $\mathbb{I}^{\breve{y}}$ denotes

Algorithm 1: OpenMix for MisD

```
Input: Training dataset $\mathcal{D}_{\text{in}}^{\text{train}}$. Outlier dataset $\mathcal{D}_{\text{out}}$.
for each iteration do
    Sample a mini-batch of ID training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ from $\mathcal{D}_{\text{in}}^{\text{train}}$;
    Sample a mini-batch of OOD outlier data $\{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{n}$ from $\mathcal{D}_{\text{out}}$;
    Generate transformed outlier data $\{(\breve{\boldsymbol{x}}_i, \breve{y}_i)\}_{i=1}^{n}$ based on Eq. 4;
    Perform common gradient descent on $f$ with $\mathcal{L}_{\text{total}}$ based on Eq. 5;
```

the one-hot label. Compared with Mixup [61], our method involves outliers and makes sure that one of the interpolated labels always belongs to the added class, *i.e.*, the $(k+1)$-th class. As shown in Sec. 5.2, other interpolation strategies like CutMix [59] and Manifold Mixup [53] can also be used.

Final learning objective. Combining reject class with outlier transformation, the final training objective of our OpenMix is as follows:

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \mathbb{E}_{\mathcal{D}_{\text{in}}^{\text{train}}}[\ell(f(\boldsymbol{x}), y)] + \gamma \mathbb{E}_{\mathcal{D}_{\text{out}}^{\text{mix}}}[\ell(f(\breve{\boldsymbol{x}}), \breve{y})] \\ &= \mathbb{E}_{\mathcal{D}_{\text{in}}^{\text{train}}}[-\mathbb{I}^{y} \log f(\boldsymbol{x})] + \gamma \mathbb{E}_{\mathcal{D}_{\text{out}}^{\text{mix}}}[-\mathbb{I}^{\breve{y}} \log f(\breve{\boldsymbol{x}})].\end{aligned} \tag{5}$$

In practice, we do not produce all mixed samples beforehand, but apply the outlier transformation in each mini-batch during training like Mixup. The details of OpenMix are provided in Algorithm 1, and Fig. 4 illustrates the overall framework of OpenMix.

Inference. Our method focuses on detecting misclassified samples from known classes. Therefore, only the original k classes are evaluated in test phase. Specifically, the predicted label of an input $\hat{y} =: \arg\max_{y \in \mathcal{Y}} f_y(\boldsymbol{x})$ and the corresponding confidence is the common MSP score, *i.e.*, $\hat{p} =: \max_{y \in \mathcal{Y}} f_y(\boldsymbol{x})$, in which $\mathcal{Y} = \{1, 2, ..., k\}$.

Why OpenMix is beneficial for MisD? Here we provide an interpretation: ***OpenMix increases the exposure of low density regions***. In standard training, it is difficult for reliable confidence learning because the low density regions (uncertain regions) are often under-explored, where few data points are mapped to those regions. This is expected as cross-entropy loss forces all samples to be correctly classified by matching their probability distributions with one-hot labels. As a result, the low density regions with rich uncertainty are largely ignored, leading to overconfidence for incorrect predictions. With OpenMix, the samples synthesized via outlier transformation, *i.e.*, mixup of the outlier and ID regions, could reflect the property of low density regions, and soft labels teach the model to be uncertain for those samples. The results in Fig. 3 confirm that OpenMix can effectively enlarge the FSU with increased exposure of low density regions. Besides, by keeping one of the classes in soft labels always belonging to the $(k+1)$ class, OpenMix can keep the confidence of correct samples over original k classes, as shown in Fig. 2 (right). Supp.M provides a theoretical justification showing that OpenMix increases the exposure of low density regions.

5. Experiments

Datasets and networks. We conduct a thorough empirical evaluation on benchmark datasets CIFAR-10 and CIFAR-100 [34]. For network architectures, we consider a wide range of DNNs such as ResNet110 [22], WideResNet [60] and DenseNet [27]. We use 300K `RandImages` [26] as the auxiliary outlier data and more discussions on the different choices of outlier datasets are presented in Sec. 5.2. Besides, the results of large-scale experiments on ImageNet [12] with ResNet-50 [21] are also reported.

Training configuration. All models are trained using SGD with a momentum of 0.9, an initial learning rate of 0.1, and a weight decay of 5e-4 for 200 epochs with the mini-batch size of 128 for CIFAR. The learning rate is reduced by a factor of 10 at 100, and 150 epochs. For experiments on ImageNet, we perform the automatic mixed precision training. Implementation details are provided in Supp.M.

Evaluation metrics. ① **AURC.** The area under the risk-coverage curve (AURC) [17] depicts the error rate computed by using samples whose confidence is higher than some confidence thresholds. ② **AUROC.** The area under the receiver operating characteristic curve (AUROC) [10] depicts the relationship between true positive rate (TPR) and false positive rate (FPR). ③ **FPR95.** The FPR at 95% TPR denotes the probability that a misclassified example is predicted as a correct one when the TPR is as high as 95%. ④ **ACC.** Test accuracy (ACC) is also an important metric.

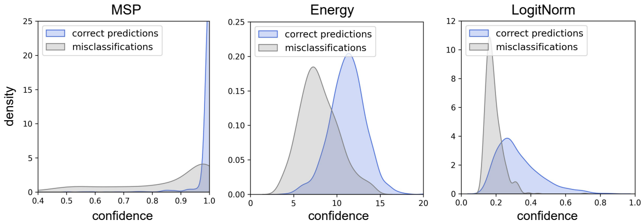

Figure 5. OOD detection methods lead to worse confidence separation between correct and wrong samples.

5.1. Comparative Results

OOD detection methods failed in detecting misclassification errors. As shown in Table 2, we observe that the simple MSP can consistently outperform Energy [39], MaxLogit [23], ODIN [38] and LogitNorm [58], which are strong OOD detection methods. The illustration in Fig. 5 shows that those methods lead to more overlap between

Table 2. Mean and standard deviations of MisD performance on CIFAR benchmarks. The experimental results are reported over three trials. The best mean results are bolded. AUROC, FPR95 and Accuracy are percentages. AURC is multiplied by 10^3.

Network	**Method**	**CIFAR-10**				**CIFAR-100**			
		AURC ↓	**AUROC ↑**	**FPR95 ↓**	**ACC ↑**	**AURC ↓**	**AUROC ↑**	**FPR95 ↓**	**ACC ↑**
ResNet110	MSP [ICLR17] [25]	9.52±0.49	90.13±0.46	43.33±0.59	94.30±0.06	89.05±1.39	84.91±0.13	65.65±1.72	73.30±0.25
	Doctor [NeurIPS21] [19]	9.51±0.49	90.15±0.44	42.95±0.78	94.30±0.06	89.84±1.12	84.94±0.09	64.75±1.37	73.30±0.25
	ODIN [ICLR18] [38]	20.82±1.09	79.45±0.75	59.32±1.08	94.30±0.06	167.53±9.93	68.95±1.95	79.64±1.43	73.30±0.25
	Energy [NeurIPS20] [39]	15.13±0.85	84.72±0.80	53.89±0.65	94.30±0.06	128.66±5.05	76.80±1.07	73.54±0.73	73.30±0.25
	MaxLogit [ICML22] [23]	14.93±0.87	85.00±0.80	53.01±1.13	94.30±0.06	125.38±4.54	77.73±0.96	70.61±0.70	73.30±0.25
	LogitNorm [ICML22] [58]	12.57±1.32	88.82±0.84	56.27±2.61	92.64±0.23	118.00±3.17	79.56±0.16	73.09±0.18	71.68±0.34
	Mixup [NeurIPS18] [61]	16.27±1.33	86.21±0.83	40.71±0.88	94.69±0.31	87.39±1.83	84.60±0.88	64.95±3.28	75.08±0.30
	RegMixup [NeurIPS22] [48]	7.88±0.64	89.40±0.64	50.91±1.47	**95.10±0.23**	75.76±2.00	84.80±0.48	64.75±1.16	**76.15±0.14**
	OpenMix (ours)	**6.31±0.32**	**92.09±0.36**	**39.63±2.36**	94.98±0.20	**73.84±1.31**	**85.83±0.22**	**64.22±1.35**	75.77±0.35
WRNet	MSP [ICLR17] [25]	4.76±0.62	93.14±0.38	30.15±1.98	95.91±0.07	46.84±0.90	88.50±0.44	56.64±1.33	80.76±0.18
	Doctor [NeurIPS21] [19]	4.75±0.61	93.13±0.38	30.46±1.90	95.91±0.07	47.34±1.31	88.41±0.23	57.64±0.64	80.76±0.18
	ODIN [ICLR18] [38]	20.37±3.36	74.70±2.67	62.04±2.86	95.91±0.07	72.58±0.69	81.02±0.37	65.22±0.53	80.76±0.18
	Energy [NeurIPS20] [39]	6.91±0.66	90.47±0.51	39.13±2.07	95.91±0.07	57.30±1.24	85.05±0.34	64.15±0.26	80.76±0.18
	MaxLogit [ICML22] [23]	6.85±0.66	90.60±0.52	37.01±2.38	95.91±0.07	56.07±1.24	85.62±0.32	61.57±0.56	80.76±0.18
	LogitNorm [ICML22] [58]	5.81±0.45	91.06±0.26	46.06±2.24	95.50±0.33	72.05±1.32	82.23±0.28	66.32±0.11	79.11±0.09
	Mixup [NeurIPS18] [61]	5.30±2.02	90.79±2.64	29.68±3.26	96.71±0.05	46.91±2.43	87.61±0.46	56.05±2.50	82.51±0.18
	RegMixup [NeurIPS22] [48]	3.36±0.27	92.31±0.34	37.48±4.96	97.10±0.14	40.36±1.71	88.33±0.35	56.44±0.95	82.50±0.30
	OpenMix (ours)	**2.32±0.15**	**94.81±0.34**	**22.08±1.86**	**97.16±0.10**	**39.61±0.54**	**89.06±0.11**	**55.00±1.29**	**82.63±0.06**
DenseNet	MSP [ICLR17] [25]	5.66±0.45	93.14±0.65	38.64±4.70	94.78±0.16	66.11±1.56	86.20±0.04	62.79±0.83	76.96±0.20
	Doctor [NeurIPS21] [19]	5.64±0.45	93.19±0.63	38.29±4.90	94.78±0.16	67.45±1.34	86.30±0.05	63.47±0.34	76.96±0.20
	ODIN [ICLR18] [38]	15.37±1.98	82.02±2.22	61.77±3.53	94.78±0.16	110.50±5.09	75.71±0.72	76.37±0.89	76.96±0.20
	Energy [NeurIPS20] [39]	8.60±0.84	89.21±1.18	51.31±2.69	94.78±0.16	100.13±3.47	78.03±0.55	74.46±0.65	76.96±0.20
	MaxLogit [ICML22] [23]	8.38±0.81	89.57±1.15	48.96±2.48	94.78±0.16	96.69±3.26	79.14±0.49	70.52±0.57	76.96±0.20
	LogitNorm [ICML22] [58]	10.89±0.71	88.70±0.27	56.59±3.07	93.59±0.34	116.35±3.22	78.14±0.60	74.81±0.89	73.13±0.48
	Mixup [NeurIPS18] [61]	9.55±0.19	89.87±0.47	37.21±1.09	94.92±0.08	63.76±3.28	86.09±0.81	63.94±2.86	77.82±0.42
	RegMixup [NeurIPS22] [48]	5.20±0.45	92.02±0.95	41.50±3.45	95.50±0.03	55.81±1.40	87.14±0.22	63.98±1.36	78.68±0.45
	OpenMix (ours)	**4.68±0.72**	**93.57±0.81**	**33.57±3.70**	**95.51±0.23**	**53.83±0.93**	**87.45±0.18**	**62.22±1.15**	**78.97±0.31**

Table 3. Comparison with other methods using VGG-16. Results with "†" are from [9]. E-AURC is also reported following [9].

Method	**AURC ↓**	**E-AURC ↓**	**FPR95 ↓**	**AUROC ↑**
	CIFAR-10			
MSP [ICLR17] [25] †	12.66±0.61	8.71±0.50	49.19±1.42	91.18±0.32
MCDropout [ICML16] [16] †	13.31±2.63	9.46±2.41	49.67±2.66	90.70±1.96
TrustScore [NeurIPS18] [30] †	17.97±0.45	14.02±0.34	54.37±1.96	87.87±0.41
TCP [TPAMI21] [9] †	11.78±0.58	7.88±0.44	45.08±1.58	92.05±0.34
SS [NeurIPS21] [41]	-	-	44.69	92.22
OpenMix (ours)	**6.31±0.18**	**4.41±0.15**	**38.48±1.30**	**93.56±0.26**
	CIFAR-100			
MSP [ICLR17] [25]†	113.23±2.98	51.93±1.20	66.55±1.56	85.85±0.14
MCDropout [ICML16] [16]†	101.41±3.45	46.45±1.91	63.25±0.66	86.71±0.30
TrustScore [NeurIPS18] [30] †	119.41±2.94	58.10±1.09	71.90±0.93	84.41±0.15
TCP [TPAMI21] [9]†	108.46±2.62	47.15±0.95	62.70±1.04	87.17±0.21
OpenMix (ours)	**73.44±0.65**	**36.41±0.45**	**61.58±0.94**	**87.47±0.12**

misclassified and correct ID data compared with MSP. This is surprising and undesirable because in practice both OOD and misclassified samples result in significant loss, and therefore should be rejected and handed over to humans. This observation points out an interesting future research direction of developing confidence estimation methods that consider OOD detection and MisD in a unified manner.

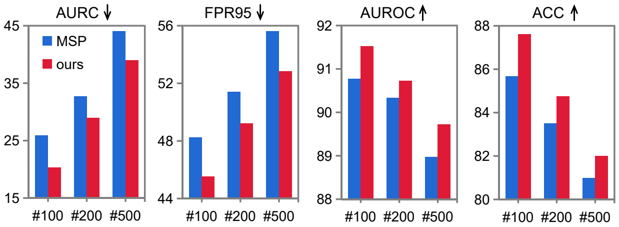

Figure 6. Large-scale experiments on ImageNet.

OpenMix improves the reliability of confidence. ① *Comparison with MSP*. The results in Table 2 show that OpenMix widely outperforms the strong baseline MSP. For instance, compared with MSP, ours successfully reduces the FPR95 from 30.14% to 22.08% under the CIFAR-10/WRNet setting. ② *Comparison with Mixup variants*. We compare OpenMix with the original Mixup [61] and its recently developed variant RegMixup [48]. We can find that they can also be outperformed by OpenMix. ③ *Comparison with TCP and other methods*. Since TCP [8] is based on misclassified training samples, it can not be used for models with high training accuracy. Therefore, we make comparison on VGG-16 [50]. In Table 3, OpenMix outperforms TCP, SS [41], MCDropout [16] and TrustScore [30].

Large-scale experiments on ImageNet. To demonstrate the scalability of our method, in Fig. 6, we report the results on ImageNet. Specifically, three settings which consist of random 100, 200, and 500 classes from ImageNet are conducted. For each experiment, we randomly sample another set of disjoint classes from ImageNet as outliers. As can be seen, OpenMix consistently boosts the MisD performance of baseline, improving the confidence reliability remarkably. Detailed training setups are provided in Supp.M.

Further analysis on accuracy-rejection curves. Fig. 7 plots the accuracy against rejection rate, *i.e.*, accuracy-

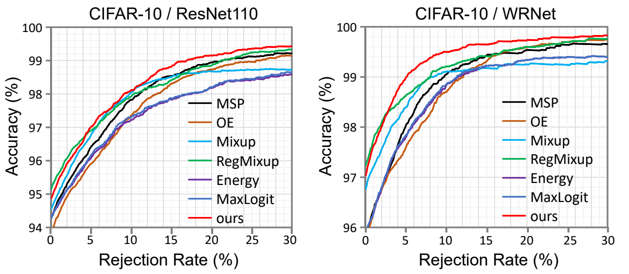

Figure 7. Accuracy-rejection curves analysis: ① ***Diverging*** between OOD detection methods (OE, Energy, MaxLogit) and MSP. ② ***Crossing-over*** between Mixup/RegMixup and MSP. ③ ***Evenly spaced*** between our method and MSP.

rejection curve (ARC) [46], to straightway and graphically make comparison among several models. Particularly, we identify three different types of relationships described in [46], *i.e.*, *diverging*, *crossing-over*, and *evenly spaced*. For selection of the best model by ARCs, ① if the desired accuracy is known, one can move horizontally across the ARC plot and select the model with the lowest rejection rate. ② Conversely, if the acceptable rejection rate is known, we select the model with the highest accuracy. The results in Fig. 7 recommend our method as the best in both cases.

Table 4. Ablation Study of each component in our method.

Method	**CIFAR-10**				**CIFAR-100**			
	AURC	**AUROC**	**FPR95**	**ACC**	**AURC**	**AUROC**	**FPR95**	**ACC**
MSP	9.52	90.13	43.33	94.30	89.05	84.91	65.65	73.30
+ RC	9.55	91.15	40.03	94.02	94.31	85.53	65.78	71.44
+ OT	12.38	87.13	61.83	93.84	99.86	82.51	72.94	72.62
OpenMix	**6.31**	**92.09**	**39.63**	**94.98**	**73.84**	**85.83**	**64.22**	**75.77**

5.2. Ablation Study

The effect of each component of OpenMix. Our method is comprised of two components: *learning with reject class* (**RC**) and *outlier transformation* (**OT**). ① With only RC, the original outlier samples are used and labeled as the $k+1$ class. ② With only OT, it is reasonable to assign the following soft label to the mixed data: $\mathbb{I}^{\tilde{y}} = \lambda \mathbb{I}^{y} + (1-\lambda)\mathcal{U}$. From Table 4, we have three key observations: Firstly, RC performs slightly better or comparable with MSP, indicating that directly mapping OOD outliers to a reject class offers limited help. Secondly, OT alone can observably harm the performance. We expect this is because the interpolation between ID labels and uniform distribution suffers from the same issue as OE. Thirdly, OpenMix integrates them in a unified and complementary manner, leading to significant and consistent improvement over baseline. Supp.M provides more results on WRNet and DenseNet.

The choices of outlier dataset. Fig. 8 reports results of using different outlier datasets. First, we can observe that using simple noises like `Gaussian noise` in OpenMix can lead to notable improvement. This verifies our insight that exposing low density regions is beneficial for MisD. Secondly, real-world datasets with semantic information yield better performance.

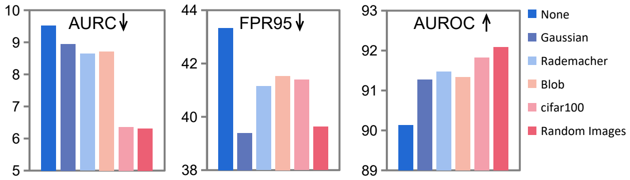

Figure 8. Ablation study on the effect of different outlier datasets.

Comparison of different interpolation strategies. We use Mixup for outlier transformation due to its simplicity. Table 5 (CIFAR/ResNet110) shows that CutMix [59] and Manifold Mixup [53] are also effective, further improving the performance of OpenMix.

Table 5. Comparison of different interpolation strategies.

Method	CIFAR-10				CIFAR-100			
	AURC	**AUROC**	**FPR95**	**ACC**	**AURC**	**AUROC**	**FPR95**	**ACC**
ours w/Mixup	6.31	92.09	39.63	94.98	73.84	85.83	**64.22**	75.77
ours w/CutMix	6.74	**93.45**	**36.82**	93.73	76.28	**86.49**	64.78	74.15
ours w/Manifold	**5.67**	92.46	36.91	**95.21**	**71.71**	85.83	66.11	**76.13**

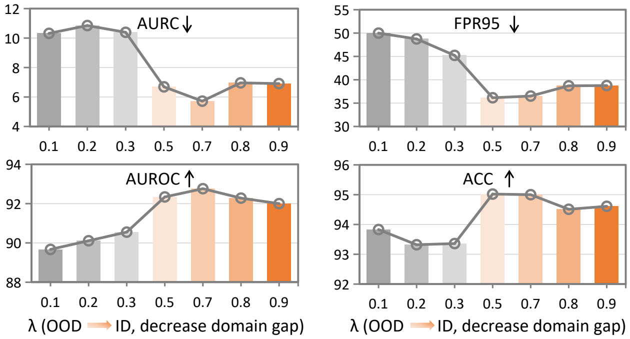

Figure 9. Relationship between domain gap and performance gain. CIFAR-10/ResNet110, the used outlier dataset is `RandImages`.

5.3. Further Experiments and Analysis

The relationship between domain gap and performance gain. Given a specific outlier dataset, the proposed outlier transformation can control and adjust the domain gap flexibly: if the outlier set is far, we can increase the ID information by enlarging λ in Eq. 4, and vice versa. In Fig. 9, we can observe that decreasing the domain gap firstly increases the performance gain and then reduces the gain.

OpenMix improves CRL and FMFP. CRL [43] ranks the confidence to reflect the historical correct rate. FMFP [64] improves confidence reliability by seeking flat minima. Differently, OpenMix focuses on the complementary strategy to exploit the unlabeled outlier data. We show in Table 6 that our method can consistently boost the performance of

Table 6. Integrating OpenMix with CRL [43] and FMFP [64]. Our method can remarkably improve their MisD performance on CIFAR-10.

Method	**AURC ↓**			**AUROC ↑**			**FPR95 ↓**			**ACC ↑**		
	ResNet110	WRNet	DenseNet	ResNet110	WRNet	DenseNet	ResNet110	WRNet	DenseNet	ResNet110	WRNet	DenseNet
CRL [43]	6.60±0.12	3.99±0.17	5.71±0.24	93.59±0.05	94.37±0.21	93.70±0.14	41.00±0.28	32.83±1.17	39.03±0.69	93.63±0.08	95.42±0.20	94.33±0.13
+ ours	**4.48±0.10**	**2.02±0.05**	**4.02±0.38**	**94.43±0.02**	**95.40±0.11**	**94.48±0.51**	**33.20±0.43**	**25.50±0.93**	**44.43±2.21**	**94.85±0.10**	**96.95±0.07**	**95.36±0.09**
FMFP [64]	5.33±0.15	2.28±0.03	4.09±0.11	94.07±0.09	95.71±0.12	94.82±0.10	39.37±0.77	25.20±1.23	30.35±1.72	94.36±0.09	96.55±0.08	95.11±0.16
+ ours	**3.94±0.11**	**1.70±0.12**	**3.58±0.16**	**94.32±0.10**	**95.90±0.13**	**94.64±0.12**	**30.41±0.83**	**18.78±2.13**	**29.36±0.68**	**95.43±0.08**	**97.33±0.11**	**95.71±0.13**

CRL and FMFP, demonstrating the complementarity effectiveness of OpenMix.

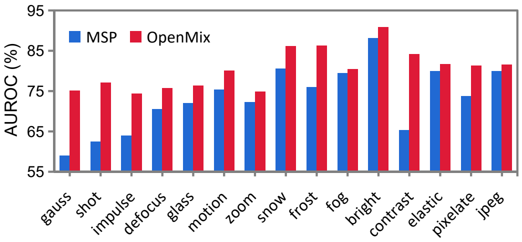

Figure 10. MisD under distribution shift. Performance on 15 types of corruption under the severity level of 5 is reported. The model is trained on CIFAR-10/ResNet110 and tested on C10-C [24].

MisD under distribution shift. In practice, environments can be easily changed, *e.g.*, weather change from sunny to cloudy then to rainy. The model still needs to make reliable decisions under such distribution or domain shift conditions. To mimic those scenarios, we test the model on corruption datasets like C10-C [24]. Fig. 10 shows that OpenMix significantly improves the MisD performance under various corruptions, and the averaged AUROC can be improved from 73.28% to 80.44%. Supp.M provides averaged results under different severity levels and corruptions on C10-C and C100-C.

Table 7. OpenMix improves MisD in long-tailed recognition.

Method	**CIFAR-10-LT**				**CIFAR-100-LT**			
	AURC	**AUROC**	**FPR95**	**ACC**	**AURC**	**AUROC**	**FPR95**	**ACC**
LA [42]	62.13	84.52	69.77	79.02	347.43	78.46	76.47	41.69
+ CRL	63.81	85.30	63.05	78.50	345.05	78.74	76.19	41.58
+ ours	**38.07**	**87.21**	**64.14**	**83.60**	**284.77**	**81.22**	**73.80**	**46.52**
VS [33]	58.45	84.47	70.15	80.11	343.48	78.20	77.25	42.20
+ CRL	62.06	83.98	67.19	79.69	345.06	78.29	77.44	41.88
+ ours	**41.52**	**87.12**	**63.31**	**83.02**	**277.34**	**81.42**	**72.93**	**47.16**

MisD in long-tailed recognition. The class distributions in real-world settings often follow a long-tailed distribution [3,42]. For example, in a disease diagnosis system, the normal samples are typically more than the disease samples. In such failure-sensitive applications, reliable confidence estimation is especially crucial. We use long-tailed classification datasets CIFAR-10-LT and CIFAR-100-LT [3] with an imbalance ratio $\rho = 100$. The network is ResNet-32. We built our method on two long-tailed recognition methods LA [42] and VS [33]. Table 7 shows our method remarkably improves MisD performance and long-tailed classification accuracy. More results can be found in Supp.M.

OpenMix improves OOD detection. A good confidence estimator should help separate both the OOD and misclassified ID samples from correct predictions. Therefore, besides MisD, we explore the OOD detection ability of our method. The ID dataset is CIFAR-10. For the OOD datasets, we follow recent works that use ***six*** common benchmarks: Textures, SVHN, Place365, LSUN-C, LSUN-R and iSUN. Metrics are AUROC, AUPR and FPR95 [25]. Table 8 shows that OpenMix also achieves strong OOD detection performance along with high MisD ability, which is not achievable with OE and other OOD detection methods. Results on CIFAR-100 can be found in Supp.M.

Table 8. OOD detection performance. All values are percentages and are averaged over ***six*** OOD test datasets.

Method	**FPR95 ↓**			**AUROC ↑**			**AUPR ↑**		
	ResNet	WRN	DenseNet	ResNet	WRN	DenseNet	ResNet	WRN	DenseNet
MSP [25]	51.69	40.83	48.60	89.85	92.32	91.55	97.42	97.93	98.11
LogitNorm [58]	29.72	12.97	19.72	94.29	97.47	96.19	98.70	99.47	99.11
ODIN [38]	35.04	26.94	30.67	91.09	93.35	93.40	97.47	97.98	98.30
Energy [39]	33.98	25.48	30.01	91.15	93.58	93.45	97.49	98.00	98.35
MaxLogit [23]	34.61	26.72	30.99	91.13	93.14	93.44	97.46	97.78	98.35
OE [26]	**5.28**	**3.49**	**5.25**	**98.04**	**98.59**	**98.20**	**99.55**	**99.71**	**99.62**
FMFP [64]	39.50	26.83	35.12	93.83	96.22	94.88	98.73	99.23	98.95
OpenMix (ours)	39.72	16.86	32.75	93.22	96.92	94.85	98.46	99.34	98.84

6. Conclusive Remarks

MisD is an important but under-explored area of research. In this paper, we propose OpenMix, a simple yet effective approach that explores outlier data for helping detect misclassification errors. Extensive experiments demonstrate that OpenMix significantly improves the confidence reliability of DNNs and yields strong performance under distribution shift and long-tailed scenarios. Particularly, recent works [1,5,28,32] claim that none of the existing methods performs well for both OOD detection and MisD. Fortunately, the proposed OpenMix can detect OOD and misclassified samples in a unified manner. We hope that our work opens possibilities to explore unified methods that can detect both OOD samples and misclassified samples.

Acknowledgement. This work has been supported by the National Key Research and Development Program (2018AAA0100400), National Natural Science Foundation of China (61836014, 62222609, 62076236, 61721004), Key Research Program of Frontier Sciences of Chinese Academy of Sciences (ZDBS-LY-7004), Youth Innovation Promotion Association of Chinese Academy of Sciences (2019141).

References

[1] Mélanie Bernhardt, Fabio De Sousa Ribeiro, and Ben Glocker. Failure detection in medical image classification: A reality check and benchmarking testbed. *Transactions on Machine Learning Research*. 8

[2] Julian Bitterwolf, Alexander Meinke, Maximilian Augustin, and Matthias Hein. Breaking down out-of-distribution detection: Many methods based on ood training data estimate a combination of the same core quantities. In *ICML*, pages 2041–2074, 2022. 1

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019. 8

[4] Senqi Cao and Zhongfei Zhang. Deep hybrid models for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4733–4743, 2022. 2

[5] Jun Cen, Di Luan, Shiwei Zhang, Yixuan Pei, Yingya Zhang, Deli Zhao, Shaojie Shen, and Qifeng Chen. The devil is in the wrongly-classified samples: Towards unified open-set recognition. In *ICLR*, 2023. 8

[6] Zhen Cheng, Fei Zhu, Xu-Yao Zhang, and Cheng-Lin Liu. Average of pruning: Improving performance and stability of out-of-distribution detection. *arXiv preprint arXiv:2303.01201*, 2023. 2

[7] C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970. 2

[8] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *NeurIPS*, pages 2898–2909, 2019. 2, 3, 6

[9] Charles Corbière, Nicolas Thome, Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Perez. Confidence estimation via auxiliary models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 6

[10] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML*, pages 233–240, 2006. 3, 5

[11] Wim De Neys, Walter Schaeken, and Géry d'Ydewalle. Working memory and everyday conditional reasoning: Retrieval and inhibition of stored counterexamples. *Thinking & Reasoning*, 11(4):349–381, 2005. 1

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5

[13] Xin Dong, Junfeng Guo, Ang Li, Wei-Te Ting, Cong Liu, and HT Kung. Neural mean discrepancy for efficient out-of-distribution detection. In *CVPR*, pages 19217–19227, 2022. 1, 2

[14] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017. 1

[15] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. In *NeurIPS*, volume 34, pages 7068–7081, 2021. 1

[16] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, volume 48, pages 1050–1059, 2016. 6

[17] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *NeurIPS*, pages 4878–4887, 2017. 3, 5

[18] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 1

[19] Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, and Pablo Piantanida. Doctor: A simple method for detecting misclassification errors. In *NeurIPS*, volume 34, pages 5669–5681, 2021. 1, 6

[20] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, pages 1321–1330, 2017. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016. 5

[23] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. 2022. 1, 2, 5, 6, 8

[24] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 8

[25] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 1, 2, 3, 6, 8

[26] Dan Hendrycks, Mantas Mazeika, and Thomas G Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019. 1, 2, 3, 4, 5, 8

[27] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269, 2017. 5

[28] Paul F Jaeger, Carsten T Lüth, Lukas Klein, and Till J Bungert. A call to reflect on evaluation practices for failure detection in image classification. In *ICLR*, 2023. 8

[29] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020. 1

[30] Heinrich Jiang, Been Kim, and Maya R. Gupta. To trust or not to trust a classifier. In *NeurIPS*, 2018. 6

[31] Philip N Johnson-Laird. Mental models and human reasoning. *Proceedings of the National Academy of Sciences*, 107(43):18243–18250, 2010. 1

[32] Jihyo Kim, Jiin Koo, and Sangheum Hwang. A unified benchmark for the unknown detection capability of deep neural networks. *arXiv preprint arXiv:2112.00337*, 2021. 8

[33] Ganesh Ramachandra Kini, Orestis Paraskevas, Samet Oymak, and Christos Thrampoulidis. Label-imbalanced and group-sensitive classification under overparameterization. In *NeurIPS*, pages 18970–18983, 2021. 8

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 3, 5

[35] Duong Le, Khoi Duc Nguyen, Khoi Nguyen, Quoc-Huy Tran, Rang Nguyen, and Binh-Son Hua. Poodle: Improving few-shot learning via penalizing out-of-distribution samples. *NeurIPS*, 34:23942–23955, 2021. 2

[36] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, pages 7167–7177, 2018. 2

[37] Saehyung Lee, Changhwa Park, Hyungyu Lee, Jihun Yi, Jonghyun Lee, and Sungroh Yoon. Removing undesirable feature contributions using out-of-distribution data. In *ICLR*, 2020. 2

[38] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018. 2, 5, 6, 8

[39] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *NeurIPS*, 33:21464–21475, 2020. 1, 2, 3, 5, 6, 8

[40] Philipp Liznerski, Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Klaus Robert Muller, and Marius Kloft. Exposing outlier exposure: What can be learned from few, one, and zero outlier images. *Transactions on Machine Learning Research*. 3

[41] Yan Luo, Yongkang Wong, Mohan S Kankanhalli, and Qi Zhao. Learning to predict trustworthiness with steep slope loss. *NeurIPS*, 2021. 2, 6

[42] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021. 8

[43] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *ICML*, pages 7034–7044, 2020. 2, 7, 8

[44] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. Calibrating deep neural networks using focal loss. In *NeurIPS*, 2020. 2

[45] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *NeurIPS*, pages 4696–4705, 2019. 2, 4

[46] Malik Sajjad Ahmed Nadeem, Jean-Daniel Zucker, and Blaise Hanczar. Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In *Machine Learning in Systems Biology*, pages 65–81, 2009. 7

[47] Dongmin Park, Hwanjun Song, MinSeok Kim, and Jae-Gil Lee. Task-agnostic undesirable feature deactivation using out-of-distribution data. *NeurIPS*, 34:4040–4052, 2021. 2

[48] Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip HS Torr, and Puneet K Dokania. Regmixup: Mixup as a regularizer can surprisingly improve accuracy and out distribution robustness. In *NeurIPS*, 2022. 6

[49] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Björn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *ICML*, 2020. 4

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6

[51] Engkarat Techapanurak, Masanori Suganuma, and Takayuki Okatani. Hyperparameter-free out-of-distribution detection using cosine similarity. In *ACCV*, 2020. 2

[52] Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NeurIPS*, pages 13888–13899, 2019. 2

[53] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR, 2019. 5, 7

[54] Niki Verschueren, Walter Schaeken, and Gery d'Ydewalle. Everyday conditional reasoning: A working memory-dependent tradeoff between counterexample and likelihood use. *Memory & Cognition*, 33(1):107–119, 2005. 1

[55] Haotao Wang, Aston Zhang, Yi Zhu, Shuai Zheng, Mu Li, Alex J Smola, and Zhangyang Wang. Partial and asymmetric contrastive learning for out-of-distribution detection in long-tailed recognition. In *ICML*, pages 23446–23458. PMLR, 2022. 2

[56] Hongxin Wei, Lue Tao, Renchunzi Xie, and Bo An. Open-set label noise can improve robustness against inherent label noise. *NeurIPS*, 34:7978–7992, 2021. 2, 3

[57] Hongxin Wei, Lue Tao, Renchunzi Xie, Lei Feng, and Bo An. Open-sampling: Exploring out-of-distribution data for re-balancing long-tailed datasets. In *ICML*, pages 23615–23630. PMLR, 2022. 2, 3

[58] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *ICML*, 2022. 2, 5, 6, 8

[59] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *CVPR*, pages 6023–6032, 2019. 5, 7

[60] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 5

[61] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 4, 5, 6

[62] Xu-Yao Zhang, Cheng-Lin Liu, and Ching Y Suen. Towards robust pattern recognition: A review. *Proceedings of the IEEE*, 108(6):894–922, 2020. 1

[63] Xu-Yao Zhang, Guo-Sen Xie, Xiuli Li, Tao Mei, and Cheng-Lin Liu. A survey on learning to reject. *Proceedings of the IEEE*, 111(2):185–215, 2023. 1

[64] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Rethinking confidence calibration for failure prediction. In *ECCV*, pages 518–536. Springer, 2022. 2, 3, 7, 8

[65] Fei Zhu, Xu-Yao Zhang, Rui-Qi Wang, and Cheng-Lin Liu. Learning by seeing more classes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2