Diagnosing Pretrained Models for Out-of-distribution Detection

Haipeng Xiong Kai Xu Angela Yao
National University of Singapore, Singapore
haipeng,kxu,ayao@comp.nus.edu.sg

Abstract

This work questions a common assumption of OOD detection, that models with higher in-distribution (ID) accuracy tend to have better OOD performance. Recent findings show this assumption doesn't always hold. A direct observation is that the later version of torchvision models improves ID accuracy but suffers from a significant drop in OOD performance. We systematically diagnose torchvision training recipes and explain this effect by analyzing the maximal logits of ID and OOD samples. We then propose post-hoc and training-time solutions to mitigate the OOD decrease by fixing problematic augmentations in torchvision recipes. Both solutions enhance OOD detection and maintain strong ID performance.

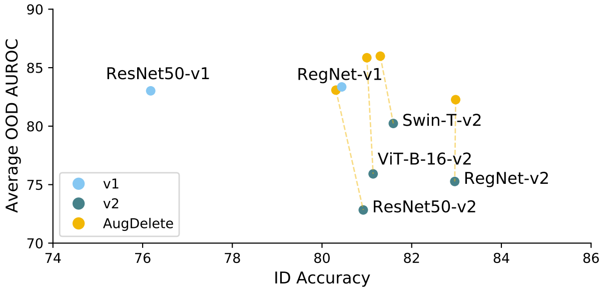


Figure 1. OOD and ID performance comparison between `torchvision` v1, v2 and AugDelete (ours) models on ImageNet-1K. AUROC is averaged among near-OOD and far-OOD datasets. Despite better ID accuracy, v2 models deliver worse OOD performance than v1 models. AugDelete can improve the OOD performance of v2 models while maintaining strong ID accuracy.

1. Introduction

Out-of-distribution (OOD) detection identifies input samples that differ from the in-distribution (ID) training data. Detecting such samples avoids overconfident or incorrect predictions on data outside the training scope. It is important for sensitive domains such as healthcare, autonomous driving, and security. Previous works [25, 38] have shown that a model's ID and OOD detection performance are correlated - the higher the ID classification accuracy (on CIFAR, ImageNet, *etc*.), the better it is at distinguishing OOD versus ID samples. It is assumed that a stronger separation of ID classes will lead to a natural separation of OOD from ID classes. Improvements may come from the learning rate schedule or model ensemble, though data augmentation is the most effective [25]. RandAugment [3], Style Augment [8], and AugMix [13] are all effective for improving both ID and OOD performance. These strategies use a combination of techniques, such as image rotation, translation, or color transformation.

`Torchvision` is one of the most widely used model libraries in OOD detection [45]. Curiously, some observations of `torchvision` models challenge the conventional understanding of correlation between ID and OOD performance. Specifically, `torchvision v2` models have worse OOD performance than `v1` models, despite better ID accuracy (see Fig. 1). Similar trends hold across a wide range of convolutional and transformer network architectures, including ResNet, MobileNet, ResNetXt, WideResNet, RegNet, SWIN-T, and ViT (See Fig. 1, Appendix 8.9).

To understand this unusual phenomenon, we systematically diagnose `torchvision` models and find that label-related augmentation strategies - label smoothing [34] and mixup [44] are mainly responsible for the poor OOD performance of `torchvision` models. In contrast, other training augmentations in `torchvision` models have little effect on OOD detection. We further analyze the impact of different augmentations on the logit space and find that only label smoothing and mixup significantly reduce the maximal logit values. The reduction of the maximal logit is more pronounced for in-distribution (ID) samples than out-of-distribution (OOD) samples, thereby diminishing their separability. In turn, logit-based and hybrid methods that rely on the logit values for OOD separation, such as the maximal logit score (MLS) [15] or nearest neighbor guidance score (NNGuide) [29], are compromised.

Our findings are in line with previous observations, albeit on a different task of failure detection in ID setups [41, 48], and the impact of mixup on OOD detection [30]. However, our findings are more comprehensive, as we uncover a distinction on the impacts of several data-based versus label-based augmentations on OOD detection. Our work covers a wide range of data augmentations and training techniques featured in `torchvision` models.

To counter the degradation of OOD detection, we propose two novel methods: Augmentation Deletion (AugDelete), for fine-tuning pre-trained models, and Augmentation Revision (AugRevise), for training models from scratch. AugDelete mitigates the negative effects of data augmentations by removing them from the training recipe and finetuning only the final layer of the network. In contrast, AugRevise introduces a revised data augmentation method paired with a corresponding training strategy to enhance OOD detection while preserving in-distribution generalization.

Both AugDelete and AugRevise demonstrate improvements over baselines in OOD detection (see Fig. 1 and Table 4). While the OOD performance of post-hoc methods [5, 32, 45] heavily rely on the quality of ID pretrained networks, AugDelete reduces the dependence by fixing pre-training issues in a post-hoc manner. AugRevise, as a training-time fix, enlarges the ID-OOD separation with a novel virtual separation loss. It improves upon related methods like MOS [17] and RegMixup [30], and outperforms state-of-the-art training-based methods regarding OOD detection and ID accuracy. Our contributions are as follows:

- We are the first to systematically diagnose pre-trained `torchvision` models and relate their poor OOD performance to their pre-training process on ID data.
- From our analysis, we propose an efficient post-hoc fix, AugDelete, to fix pretraining issues of `torchvision` models without retraining; it improves OOD detection of various pre-trained CNNs and transformers.
- We also propose a training-time fix, AugRevise, which can enhance OOD detection while improving in-distribution performance. AugRevise shows superior OOD results on challenging Openood v1.5 Benchmarks.

Our work on `torchvision` models can benefit future research as these models are widely used in OOD detection benchmarks [45]. Code is available at `https://github.com/xhp-hust-2018-2011/DiagPMOOD`.

2. Related Works

Post-Hoc OOD Detection methods often use pre-trained models; the main research focus is to define new score functions or post-hoc adjustments to improve detection capabilities. Most methods are derived from output logits[5, 10, 24, 32] and modify the logits by reshaping the feature activation. Feature-based methods model the behavior of internal feature representations. For instance, Mahalanobis distance-based methods [23] calculate the distance of feature vectors from class-conditional Gaussian distributions, effectively identifying OOD samples by measuring feature space uncertainty. In addition to logit-based and feature-based techniques, recent work like NNGuide [29] combines the two to derive more robust OOD detection scores.

Training-based OOD Detection methods adjust model training to improve the model's ability to distinguish between ID and OOD samples. One strategy is through explicit supervision, either from true OOD samples [12] or synthesized virtual ones [17, 30]. Synthesized samples are more appealing, since real OOD data is typically not available for training. For example, RegMixup [30] treat mixed-up samples as virtual outliers, the cross-entropy loss of which serves as regularizers for strengthening the decision boundary between ID and OOD data. Other training-based techniques, like LogitNorm [40] and T2FNorm [31], aim to improve the separability of feature representations between ID and OOD samples. After the model training, a compatible post-hoc score function is still required for OOD detection.

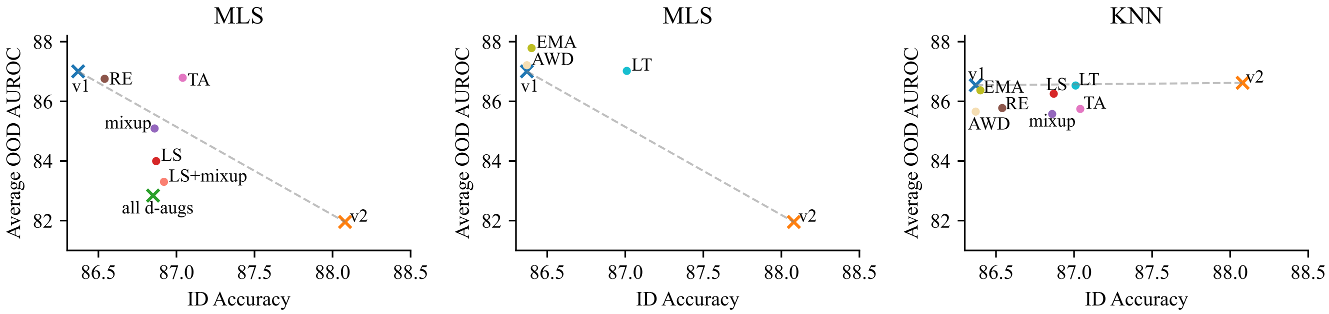
Relationships between ID and OOD data has been widely explored [18, 25, 38]. [38] found that a good closed-set classifier can identifying semantically novel classes. Similarly, [18] observed that ID and OOD accuracy are positively correlated, at least for correctly predicted ID samples. [25] found that data augmentations including AugMix and RandAugment improve both ID and OOD performance. However, we diagnose torchvision models and find that label-based augmentations–label smoothing (LS) and mixup contribute to the degraded OOD performance of torchvision models despite improving ID accuracy. Our findings on mixup are partly similar to [30] but more general because we analyze a broader range of data augmentations and training techniques in torchvision pre-training recipes. Besides, we provide an post-hoc fix, AugDelete, that is more efficient than retraining-based regmixup in [30]. Beyond OOD detection, [41, 48] observed LS and Mixup's negative impacts on ID failure detection. However, poor failure detection of ID samples does not indicate poor ID-OOD separation or degraded OOD performance. In fact, [48] has even noticed a negative correlation between ID failure detection and OOD detection. Specifically, [48] found that approaches with good OOD performance may shrink the distribution of ID samples, leading to poor ID failure detection.

3. Preliminaries

3.1. OOD Detection

A commonly used setup for OOD detection is to identify semantic shifts in image classification [15, 17, 42]. During training, only in-distribution (ID) data $\{(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{D}_{\text{ID}}, \boldsymbol{y} \in \mathcal{Y}_{\text{ID}}\}$ are observed, where $\mathcal{Y}_{\text{ID}}$ has C classes. Samples from semantically novel classes unseen in training are considered OOD. During testing, OOD samples $\{(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{D}_{\text{OOD}}, y \in \mathcal{Y}_{\text{OOD}}, \mathcal{Y}_{\text{OOD}} \cap \mathcal{Y}_{\text{ID}} = \emptyset\}$ are encountered.

To separate ID and OOD samples, a score function $S(\boldsymbol{x})$



(a)

(b)

(c)

Figure 2. ID Accuracy and OOD AUROC for various training techniques and data augmentations on ImageNet200, using a ResNet18 backbone as per [45]. (a) The impact of data augmentations on OOD detection with MLS score. Label-based augmentations, label smoothing and mixup, greatly decrease OOD performance of logit-based scores, *i.e*. MLS. "all d-augs" denotes combining all data augmentations (RE/TA/LS/mixup). (b) The impact of training techniques on OOD detection using MLS scores. (c) Impacts of both data augmentations and training techniques on OOD detection using KNN scores. Compared to OOD detection using MLS score, these strategies have less impact (≤ 1 AUROC) on OOD performance when employing KNN score.

is designed to output higher values for ID samples. Based on a threshold τ, an OOD indicator $\mathbb{1}(\boldsymbol{x};\tau)$ can be defined as

$$\mathbb{1}(\boldsymbol{x};\tau) = \begin{cases} \text{ID} & \text{if } S(\boldsymbol{x}) \geq \tau, \\ \text{OOD} & \text{if } S(\boldsymbol{x}) < \tau. \end{cases} \tag{1}$$

The score function $S(\boldsymbol{x})$ is derived from an ID classification network F. F can be decomposed as a feature extractor G sub-network and a linear layer ($\mathbf{W} \in R^{C\times D}, \boldsymbol{b} \in R^{C}$):

$$\boldsymbol{v} = F(\boldsymbol{x}) = \mathbf{W} \cdot G(\boldsymbol{x}) + \boldsymbol{b}, \qquad \boldsymbol{f} = G(\boldsymbol{x}), \tag{2}$$

where $\boldsymbol{f} \in R^{D}$ is the feature vector of the penultimate layer. Typically, network F is trained with an ID training set using the standard cross-entropy loss L_{CE}:

$$L_{CE}(\boldsymbol{v}, \boldsymbol{y}) = -\boldsymbol{y}^{T}\log(\sigma(\boldsymbol{v})), \qquad \boldsymbol{v} = F(\boldsymbol{x}), \tag{3}$$

where σ is the softmax and $\boldsymbol{v} \in R^{C}$ is the output logit.

Post-hoc OOD detection methods [5, 15, 24] use pre-trained networks, off-the-shelf to feed directly into the scoring function. They focus on post-hoc adjustment to the features $\boldsymbol{f}$ and/or designing more effective score functions $S(\boldsymbol{x})$. On the other hand, training-based methods train a novel F from scratch to improve ID/OOD separation, *e.g*. by adding regularizers [30, 40] or data augmentations [3, 14], though they still require a compatible score function $S(\boldsymbol{x})$.

Typical score functions are based on the logits $\boldsymbol{v}$, the features $\boldsymbol{f}$, or a combination of the two. For example, the maximal logit score (MLS) [15] and energy-based score [24]are defined respectively as as:

$$S_{MLS}(\boldsymbol{x}) = \max_{j=1,\dots,C} \boldsymbol{v}[j], S_{EBO}(\boldsymbol{x}) = log(\sum_{j=1}^{C} e^{\boldsymbol{v}[j]}) \tag{4}$$

where $\boldsymbol{v}[j]$ denotes the j-th element of the logit prediction. S_{EBO} is a soft approximation of S_{MLS}, and other logit-based scores such as ASH [5] or FSEBO [9] are also related to S_{MLS} since they modify logits by reshaping feature activation. A typical feature-based scoring function is the k-th nearest neighbor distance score (KNN) [33],

$$S_{KNN}(x) = -||\boldsymbol{f} - \boldsymbol{f}_{k^*}||_2, \tag{5}$$

where f_{k^*} denotes the feature of the k-th nearest neighbor in the training set. The nearest neighbor guidance score (NNGuide) [29] combines both features and logits:

$$\begin{aligned} S_{NNGuide}(\boldsymbol{x}) &= S_{EBO}(\boldsymbol{x}) \cdot \text{Guide}(\boldsymbol{x}), \\ \text{Guide}(\boldsymbol{x}) &= \frac{1}{k}\sum_{i=1}^{k} S_{EBO}(\boldsymbol{x}_{i^*}) \cdot \cos(G(\boldsymbol{x}_{i^*}), \boldsymbol{f}), \end{aligned} \tag{6}$$

where x_{i^*} denotes the i-th nearest sample in the training set and $\cos(\cdot)$ the cosine similarity function.

3.2. Advanced Recipe of `Torchvision v2` Models

`torchvision v1` and `v2` models are similar, though the latter are trained with additional data augmentations and data-independent training techniques. The augmentations can be further categorized into data-based and label-based augmentation, where the latter also adjusts the label $\boldsymbol{y}$,

Data-Based Augmentation 1: Random Erasing (RE) [46] applies random zero masking in the input sample $\boldsymbol{x}$ with a probability p^{er}. It reduces over-fitting and improve the generalization of neural networks. Typically, $p^{er} = 0.1$.

Data-Based Augmentation 2: Trivial Augment (TA) [27] is a parameter-free set of image transformations to the input sample $\boldsymbol{x}$ such as solarize, posterize, brightness adjustment, *etc*. During training, TA randomly selects a single augmentation and an augmentation strength from a pre-defined set.

Label-Based Augmentation 1: Label Smoothing (LS) [34] limits overconfidence by adding a uniform vector to label $\boldsymbol{y}$:

$$\begin{aligned} L_{CE}^{ls}(\boldsymbol{v}, \boldsymbol{y}^{ls}) &= -(\boldsymbol{y}^{ls})^{T}\log(\sigma(\boldsymbol{v})), \\ \boldsymbol{y}^{ls} &= (1-\beta)\boldsymbol{y} + \beta\boldsymbol{u}, \qquad 0 \leq \beta < 1, \end{aligned} \tag{7}$$

where $\boldsymbol{u} \in R^C$ is a uniform vector with all elements equal to 1, β is the label smoothing strength, and σ is the softmax function. A larger β denotes smoother learning targets; typically, $\beta = 0.1$.

Label-Based Augmentation 2: Mixup [44] interpolates new samples (x^{mix},y^{mix}) by linearly combining two samples in both the data and label spaces:

$$\boldsymbol{x}^{mix} = (1-\lambda)\boldsymbol{x}+\lambda\boldsymbol{x}_1, \qquad \boldsymbol{y}^{mix} = (1-\lambda)\boldsymbol{y}+\lambda\boldsymbol{y}_1. \quad (8)$$

The cross-entropy loss is applied to the mixed samples (x^{mix},y^{mix}) in a standard fashion:

$$\begin{gathered} L_{CE}^{mix}(\boldsymbol{v}^{mix}, \boldsymbol{y}^{mix}) = -(\boldsymbol{y}^{mix})^T \log(\sigma(\boldsymbol{v}^{mix})), \\ \boldsymbol{v}^{mix} = F(\boldsymbol{x}^{mix}). \end{gathered} \quad (9)$$

Mixup creates a smooth transition between different classes and can improve ID generalization.

Training techniques: Compared to `v1` models, `v2` models add data-independent training techniques such as longer training (LT), adjusted weight decay (AWD), and Exponential Moving Average (EMA) of parameters [36]. The details are listed in Appendix 8.3.

4. Diagnosing Torchvision Training Recipes

This section systematically investigates the influence of `torchvision` training recipes on OOD detection. It starts with a case study in Sec. 4.1 to identify the cause of degraded OOD performance, before explaining the reason through derivations in Sec. 4.2. Finally, we analyze mixup from the perspective of virtual sample generation in Sec. 4.3. Our analysis suggests adding mixup and label smoothing reduces the distinction between ID and mixed samples in the logit space. Less separable ID and mixed samples will result in poor ID/OOD separation because the mixed samples are closer to OOD samples.

4.1. An Empirical Study on `torchvision` Models

This paper's contributions are motivated by a case study based on the protocols of OpenOOD V1.5 [25]. OpenOOD V1.5 is currently the largest OOD detection benchmark. The findings released by the authors are in line with previous literature showing the correlation between ID and OOD performance. A curious discrepancy that we noticed is that state-of-the-art methods for OOD almost all rely on `torchvision v1` models, even though they lag in ID performance compared to `v2` models with the same backbones. The improved performance is brought about by the augmentations and training techniques described in Sec 3.2.

We begin by comparing the performance of the v1 and v2 models using a ResNet50 backbone in ImageNet-1k [4]. ResNet50-v2 improves accuracy by 4% compared to ResNet50-v1 but results in a 12% decrease in the OOD AUROC (see Fig. 1). Such a change in the OOD AUROC is significant because it surpasses the improvements that most post-hoc OOD methods achieve [5, 24]. Similar trends hold for other v1 and v2 models (see Fig. 1).

A key difference between `v1` and `v2` is the different training recipes. `v2` uses data augmentations (see Sec. 3.2) and training techniques on top of the simple augmentations (*e.g.* random resizing, cropping, and horizon flipping) used by `v1`. To pinpoint each strategy's influence, we train models from scratch on ImageNet200 [25] with a single strategy each time. Figure 2 compares the ID vs. OOD accuracy based on the MLS score and KNN score. More experimental details and results for CIFAR10/100 [20] are given in Appendix 8.3.

The augmentation effects are split in Fig. 2a. Regarding the OOD performance with MLS score, adding data augmentations to `v1` (all d-augs) results in a critical drop similar to that brought by `v2` models. Among data augmentations, the data-only augmentations *i.e.*, the Random Erasing (RE) and the Trivial Augment (TA), have minimal impact on the OOD accuracy. The label-based augmentations, *i.e.*, label smoothing (LS) and mixup, however, greatly decrease OOD performance. Combining LS and mixup (LS+mixup) likely compounds together into the significant drop in OOD for `v2`. Unlike data augmentations, training techniques (AWD/EMA/LT) have little impact on OOD performance in Fig. 2b, suggesting that the OOD performance degradation in v2 models is likely from data augmentations rather than training techniques. These negative trends are most prominent at the logit level, where the MLS scores are derived, but less pronounced at the feature level in Fig. 2c, where the KNN score is computed.

4.2. Analysis of Data Augmentations

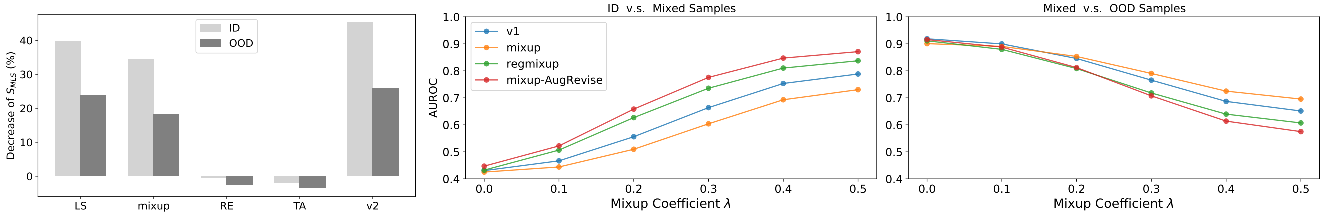
This section analyzes how different data augmentations influence OOD detection with the MLS scoring function S_{MLS} (see equation 4). Prop. 4.1 shows that adding label smoothing and mixup will decrease the maximal logits S_{MLS}. Then we link the decrement of S_{MLS} to degraded OOD performance by Prop. 4.2 and experimental verification.

Proposition 4.1. *Let i^* denote the index of the maximal logit, $\Delta \boldsymbol{v}[i^*]$ denote the increment of the maximal logit after one-step gradient descent, L_{CE}, L_{CE}^{ls} and L_{CE}^{mix} are defined as equation 3,7, and 9. We have*

$$\Delta \boldsymbol{v}[i^*] - \Delta \boldsymbol{v}^{aug}[i^*] \propto (\frac{\partial L_{CE}^{aug}}{\partial \boldsymbol{v}} - \frac{\partial L_{CE}}{\partial \boldsymbol{v}})[i^*] \geq 0, \quad (10)$$

where "aug" can be LS or mixup, and "$[j]$" denotes take the j-th element of a vector.

Remark: Proposition 4.1 suggests that label smoothing and mixup tend to decrease the gradient updation to the maximal logits during each step, thus decreasing S_{MLS}. Detailed proof can be found in Appendix Sec. 9. Figure 3a (left)



(a)

(b)

Figure 3. (a) Relative decrease (see Prop. 4.2) of the maximal logit S_{MLS} with different data augmentations. Label smoothing and mixup will reduce S_{MLS} more on ID than OOD samples. (b) AUROC between ID/OOD and mixed samples with different mixup coefficients λ. With the increasing λ, the mixed samples will be more inseparable from OOD samples, while more separable from ID samples.

visualizes the decrement of S_{MLS} of ID and OOD after data augmentations in ImageNet200. It can be observed that LS and Mixup will reduce S_{MLS}, while RE and TA do not significantly influence S_{MLS}.

Decreased maximal logits do not directly lead to a reduction in OOD performance. A trivial case is that the OOD performance remains the same if all logits decrease by a constant value. To understand the connection between the reduction in the maximal logits and the degradation in OOD performance, we derive Prop. 4.2.

Proposition 4.2. *Let s_i (s_o) denote the maximal logits of ID (OOD) samples, s_i^{aug} (s_o^{aug}) denote that of ID (OOD) samples after augmentations, and $r_i = \frac{s_i - s_i^{aug}}{s_i}$ ($r_o = \frac{s_o - s_o^{aug}}{s_o}$) denote the relative decrement of s_i^{aug} (s_o^{aug}). Suppose $s_i \geq 0$, if $1 > r_i > r_o$, we have:*

$$\delta P = Prob(s_i \geq s_o) - Prob(s_i^{aug} \geq s_o^{aug}) \geq 0, \quad (11)$$

and δP monotonically increases w.r.t $\frac{r_i - r_o}{1 - r_o}$, where "Prob(e)" denotes the probability of event e.

Remark: i) Prob($s_i \geq s_o$) is the probabilistic form of AUROC [7, 11], larger value of which indicates better OOD performance. ii) Proposition 4.2 suggests that larger logit decrement in ID than OOD ($r_i > r_o$) will cause poor OOD performance (lower $Prob(s_i^{aug} \geq s_o^{aug})$). This is the case of LS/Mixup shown in Fig. 3a. iii) LS/mixup have larger $r_i - r_o$ and r_o than RE/TA, suggesting a larger $\frac{r_i - r_o}{1 - r_o}$ in LS/mixup. According to Prop. 4.2, the decrement of OOD performance δP will be more pronounced in mixup/LS, in accordance with the observation in Fig. 2a.

4.3. Analysis of Mixup as Sample Generation

Different from label smoothing, mixup creates virtual samples from ID data. We compare the maximal logit of mixed samples to that of ID and OOD samples in Figure 3b. We find that: i) With the increasing λ, the AUROC becomes lower between mixed and OOD samples while higher between mixed and ID samples, meaning that the mixed samples will be inseparable from OOD samples. This suggests that the mixed samples can also serve as virtual OOD samples. ii) After adding mixup to the basic recipe v1, the AUROC of mixed and OOD samples will decrease for each λ, indicating that *adding mixup decreases the separability between ID and mixed samples.* As mixed samples get closer to OOD samples, less separable ID and mixed samples will likely cause less separability between ID and OOD.

Summary of Section 4: i) In `torchvision` training recipes, label-based data augmentations (LS/mixup) reduce the distinction between ID and OOD in logits during gradient updation. The negative influence will also be propagated into the feature space, though much smaller than on logits. ii) In contrast, data-based augmentations (RE/TA) and training techniques (AWD/EMA/LT) have relatively small impact on both logit and feature spaces (see Fig. 2).

5. Method

Based on the analysis of `torchvision` models, we devise two methods for fixing impaired logits. The first, augmentation deletion (Sec. 5.1), fixes the impaired logits by finetuning the last fully connected layer without problematic data augmentations. The second, augmentation revision (Sec. 5.2), revises the problematic data augmentations in the `torchvision` v2 recipe for training models from scratch.

5.1. AugDelete for Pretrained Models

Empirically, the impact of label smoothing and mixup is the greatest on the output logits. The effects gradually diminish with back-propagation into the feature layers. The results of figure 2a show less impact on OOD detection when adopting a feature-based score S_{KNN} rather than a logit-based score S_{MLS}. These empirical results suggest that a simple way to fix the logits $\boldsymbol{v}$ is to fine-tune the last fully connected layer $W, \boldsymbol{b}$ without label smoothing and mixup.

To make finetuning efficient, we extract the features $\boldsymbol{f}$ in a single forward pass and then train $W, \boldsymbol{b}$. Alg. 1 and Figure 4 show the pipeline of this simple approach termed as AugDelete. AugDelete improves the logit-based OOD detectors with minimal training cost and maintains the ID accuracy since the feature extractor G is fixed.

By retraining the last layer, AugDelete improves `torchvision` v2 models in terms of OOD detection. However, its OOD performance is simply comparable to

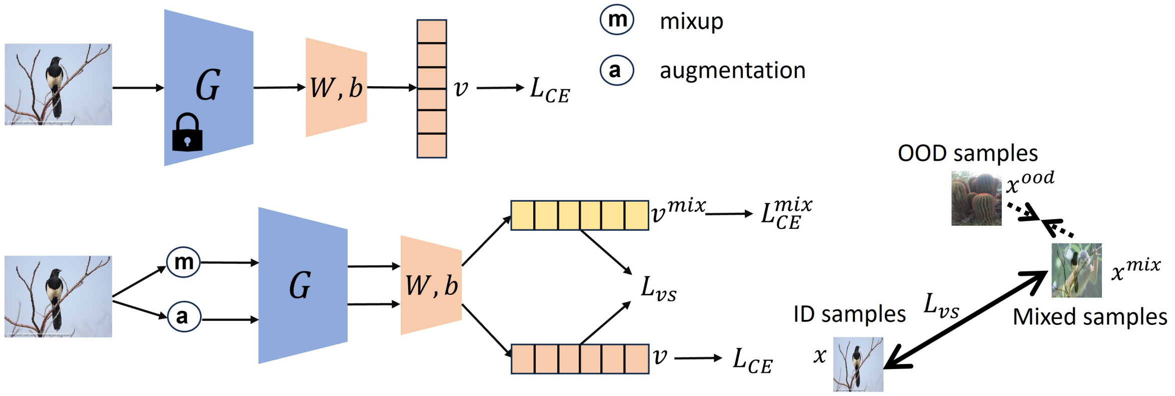


Figure 4. Pipelines of AugDelete (top) and AugRevise (bottom). In AugRevise, L_{vs} is added to enforce the separation between ID and mixed samples. As mixed samples are close to OOD samples, a better separation with ID samples also leads to better ID/OOD separation.

```
Algorithm 1: AugDelete
Input: ID training set $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$, pre-trained network
  with $G$,$\mathbf{W}$,$\boldsymbol{b}$
Output: Finetuned linear layer $\mathbf{W}$,$\boldsymbol{b}$

Extract features $\boldsymbol{f}_i$ with $G$ as equation 2
while Training not end do
    Sample a batch of $(\boldsymbol{f}_i, \boldsymbol{y}_i)$
    Compute $L_{CE}$ as equation 3
    Perform gradient descent to update $\mathbf{W}, \boldsymbol{b}$
end while
return $\mathbf{W}, \boldsymbol{b}$
```

v1 models (see ResNet or RegNet in Fig. 1) as the features themselves are left untouched. Next, we aim to surpass the v1 models in both ID and OOD by revising the v2 training recipe when training models from scratch.

5.2. AugRevise for Models Trained from Scratch

Following the analysis from Sec. 4, we make the following design decisions. We keep the data-based augmentations (Random Erasure and Trivial Augment) while removing label smoothing; the former does not harm OOD detection while the latter does. Finally, we adjust the mixup scheme to ensure that ID samples are sufficiently separable from the mixed samples. Ideally, S_{MLS} of ID samples should be larger than mixed samples. The closer λ is to 0.5, the greater the gap in S_{MLS} between ID and mixed samples.

To improve mixup for OOD detection, [30] propose regmixup, which treats mixup loss as an OOD regularizer as

$$L^{rmix}_{CE} = L_{CE}(\boldsymbol{v}, \boldsymbol{y}) + L^{mix}_{CE}(\boldsymbol{v}^{mix}, \boldsymbol{y}^{mix}). \tag{12}$$

However, we find that regmixup cannot ensure that ID samples are separable from that of mixed samples, as shown in Figure 3b. As mixed samples are close to OOD samples, poor separation between ID and mixed samples will degrade ID/OOD separation. To ensure a clear separation between mixed and ID samples, we propose a virtual separation loss:

```
Algorithm 2: AugRevise
Input: ID training set $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$, initialized network
  with $G$,$\mathbf{W}$,$\boldsymbol{b}$
Output: Trained Network $G$,$\mathbf{W}$,$\boldsymbol{b}$

while Training not end do
    Sample a batch of $(\boldsymbol{x}_i, \boldsymbol{y}_i)$
    Perform mixup to get $(\boldsymbol{x}^{mix}_i, \boldsymbol{y}^{mix}_i)$
    Perform other data augmentations to $\boldsymbol{x}_i$
    Compute logits $\boldsymbol{v}_i$,$\boldsymbol{v}^{mix}_i$ as equation 3, 8, 9
    Compute $L^{rvmix}_{CE}$ as equation 13~14
    Perform gradient descent to update $G, \mathbf{W}, \boldsymbol{b}$
end while
Call AugDelete to update linear layer $\mathbf{W}, \boldsymbol{b}$
return $G$,$\mathbf{W}$,$\boldsymbol{b}$
```

$$L_{vs}(\boldsymbol{v}, \boldsymbol{v}^{mix}) = -(1 - P_\lambda)\log(1 - P_{\boldsymbol{v}}) - P_\lambda \log(P_{\boldsymbol{v}}),$$
$$P_{\boldsymbol{v}} = \frac{\sum_{i=1}^{C} e^{\boldsymbol{v}^{mix}[i]}}{\sum_{i=1}^{C} e^{\boldsymbol{v}[i]} + e^{\boldsymbol{v}^{mix}[i]}}, P_\lambda = \frac{max(\lambda, 1-\lambda)}{max(\lambda, 1-\lambda) + 1}, \tag{13}$$

L_{vs} optimize the LogSumExp(LSE) approximation of S_{MLS} since this approximation provides dense gradients. It ensures the ratios between the maximal logits of ID and mixed samples ($\frac{S^{id}_{MLS}}{S^{mixup}_{MLS}}$) equals $\frac{1}{max(\lambda,1-\lambda)}$. $\frac{S^{id}_{MLS}}{S^{mixup}_{MLS}} \geq 1$ ensures that S_{MLS} of ID samples is larger than that of OOD samples. Moreover, $\frac{S^{id}_{MLS}}{S^{mixup}_{MLS}}$ increases as λ becomes closer to 0.5, ensuring the increasing distinction between mixed and ID samples. Overall, the final revised mixup adopts the

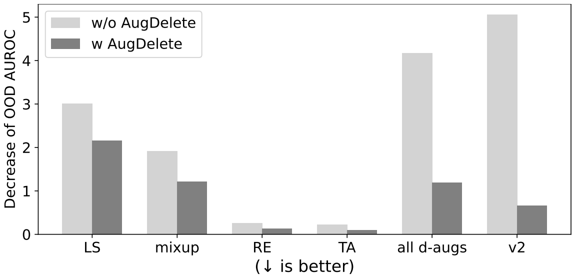


Figure 5. AugDelete for models trained with different data augmentations on ImageNet200. AugDelete improves models trained with various data augmentations, especially label smoothing and mixup.

loss L_{CE}^{rvmix}:

$$L_{CE}^{rvmix} = L_{CE}^{rmix} + L_{vs}, \tag{14}$$

This augmentation revision approach is termed as AugRevise, the pipeline of which is shown in Alg. 2 and Figure 4. Note that AugRevise still requires AugDelete after training the whole network to mitigate the influence of data augmentation in the fully connected layers.

6. Experiments

6.1. Ablation Studies

We do ablation studies on ImageNet200 to verify critical elements of AugDelete and AugRevise on OOD Detection. By default, the maximal logit score S_{MLS} is chosen as the OOD score function. More ablations are provided in Appendix Sec. 8.

AugDelete for Different Data Augmentation. Figure 5 shows the OOD detection results before and after applying AugDelete under various data augmentations. AugDelete improves models with label smoothing and mixup by a large margin. AugDelete can also slightly improve the OOD Detection performance of RE and TA. However, with AugDelete, models trained with label smoothing and mixup are still worse than the v1 model. Because AugDelete keeps the pretrained features, the negative impact of label smoothing and mixup cannot be mitigated.

Fixing Mixup for OOD Detection. Mixup is fixed in AugRevise with L_{vs} loss to increase the separability between ID and mixed samples. Table 1 shows the quantitative results of fixing mixup. Regmixup improves the vanilla mixup but cannot outperform the v1 model in OOD detection. Adopting mixup in AugRevise can outperform the v1 model in both ID classification and OOD detection. To explain the superior OOD performance of mixup-AugRevise to regmixup, we visualize the separability between ID, OOD, and mixed samples in Figure. 3b. Mixup-AugRevise delivers higher auroc between ID and mixed samples, while lower auroc between mixed and OOD samples. It suggests better separation between ID and OOD samples and mixed samples as better virtual OOD samples, thus improving the separation between ID and OOD samples.

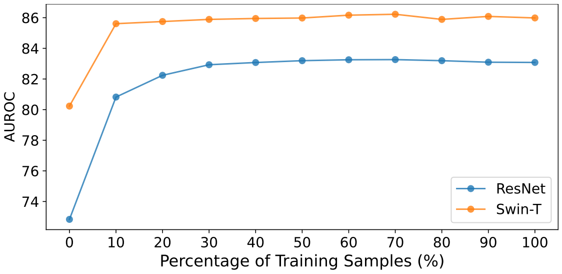


Figure 6. OOD performance of AugDelete with a fraction of training samples. 0% means do not apply AugDelete . With a small percentage (10%) of the training samples, AugDelete already delivers a significant improvement.

AugDelete vs. AugRevise Table 2 shows that AugRevise outperforms AugDelete and vanilla v1 models in both the ID classification and OOD detection. However, adding label smoothing in AugRevise will decrease the OOD performance of OOD detection, suggesting that label smoothing should be removed in AugRevise.

6.2. AugDelete for Pretrained ImageNet-1k Models

OOD detection with Pretrained Network Architectures. Figure 1 visualizes the ID accuracy and OOD performance with and without AugDeletefor different pre-trained network architectures. AugDelete improves the OOD detection of both CNNs and transformers while maintaining ID accuracy. Models with better ID accuracy show higher or at least comparable AUROC after applying AugDelete.

AugDelete with Various Percentages of Training Samples As AugDelete only finetunes the last layer, it is effective with only a fraction of the training samples. Fig. 6 shows that AugDelete , with only 10% of the training samples, already delivers a significant improvement. The OOD performance saturated with 20% $\sim$ 30% training samples, suggesting that the entire improvement of AugDelete can be achieved with only a small fraction of training samples.

AugDelete for various OOD score functions. Table 3 shows the results of applying AugDelete with various score functions $S(\boldsymbol{x})$ on `torchvision` v2 pretrained models. AugDelete improves ResNet50-v2 in all $S(\boldsymbol{x})$ by a large margin, except KNN scores. AugDelete performs comparably to ResNet50-v1 with logit-based $S(\boldsymbol{x})$ in terms of OOD detection, while having much better ID accuracy than ResNet50-v1. However, AugDelete shows worse OOD detection performance than ResNet50-v1 when adopting feature-based $S(\boldsymbol{x})$, KNN or NNGuide. This is because AugDel does not fix the impaired features of ResNet50-v2.

6.3. AugRevise for Training-Time Enhancement

We train models from scratch with AugRevise on ImageNet200/1k and CIFAR100 datasets. Following the same

Training Recipe	**Loss**	AUROC	FPR@95	ID ACC
		↑	↓	↑
v1	L_{CE}	87.00	46.90	86.37
v1+mixup	L_{CE}^{mix}	84.00	57.81	86.87
v1+regmixup	L_{CE}^{rmix}	86.97	48.03	**87.58**
v1+mixup-AugRevise	L_{CE}^{rvmix}	**87.72**	**42.09**	87.28

Table 1. Fixing mixup on ImageNet200. Fixing mixup with AugRevise can improve both ID and OOD performance of the v1 model.

Training Recipe	AUROC	FPR@95	ID ACC
	↑	↓	↑
v1	87.00	46.90	86.37
v2*	82.78	62.47	86.74
v2*+AugDelete	85.57	51.87	86.68
AugRevise	**87.88**	**41.72**	**87.67**
AugRevise+LS	87.17	43.87	87.33

Table 2. Comparison on ImageNet200. AugRevise outperforms other models. v2* is trained for 100 epochs as other models.

Method	ResNet50-v1			ResNet50-v2			ResNet50-v2 + AugDelete		
	AUROC	FPR@95	ID ACC	AUROC	FPR@95	ID ACC	AUROC	FPR@95	ID ACC
	↑	↓	↑	↑	↓	↑	↑	↓	↑
MLS [15]	83.02	53.02	76.18	72.84	84.75	80.92	83.08	63.11	80.31
ASH [5]	83.97	49.62	76.18	53.53	90.59	80.92	81.70	65.11	80.31
KNN [33]	80.64	52.50	76.18	79.91	55.09	80.92	79.91	55.09	80.31
NNGuide [29]	86.68	44.81	76.18	65.77	72.07	80.92	77.54	58.22	80.31

Table 3. AugDelete for various OOD score functions on ImageNet-1k.

Method	**CIFAR100**			**ImageNet200**			**ImageNet-1k**		
	AUROC	FPR@95	ID ACC	AUROC	FPR@95	ID ACC	AUROC	FPR@95	ID ACC
	↑	↓	↑	↑	↓	↑	↑	↓	↑
v1+MLS [15]	80.36	56.09	77.26	87.00	46.90	86.37	83.02	53.02	76.18
v1+FSEBO [9]	79.97	57.24	77.26	86.75	48.87	86.37	86.82	45.14	76.18
v1+KNN [33]	81.29	57.44	77.26	86.54	44.70	86.37	80.64	52.50	76.18
v1+NNGuide [29]	80.84	57.51	77.26	87.83	46.90	86.37	86.68	44.81	76.18
LogitNorm+MSP [40]	80.00	58.25	76.34	87.85	40.28	86.04	83.08	49.94	76.45
NPOS+KNN [35]	80.32	57.24	—	86.94	41.93	—	—	—	—
MOS+MOS [17]	80.29	56.67	76.98	75.15	61.58	85.60	77.80	64.47	72.81
AugMix+MSP [14]	78.27	57.33	76.45	87.09	44.20	87.01	82.08	55.70	77.63
RegMixup+MSP [30]	79.94	56.81	79.32	87.47	49.62	87.25	81.68	57.12	76.68
AugRevise +MLS	83.69	50.86	**82.10**	87.88	41.72	**87.67**	84.77	49.06	**77.70**
AugRevise +MSP	82.50	52.12	**82.10**	87.89	41.65	**87.67**	84.78	49.06	**77.70**
AugRevise +KNN	83.88	53.46	**82.10**	87.00	41.66	**87.67**	82.56	49.25	**77.70**
AugRevise +NNGuide	**84.51**	**49.52**	**82.10**	**89.31**	**37.56**	**87.67**	**87.17**	**43.64**	**77.70**

Table 4. Comparison with SOTA methods on CIFAR100 and ImageNet200/1k. AugRevise improves both logit-based and feature-based methods and delivers SOTA results.

settings as OpenoodV1.5, all the AugRevise models are trained for 100 epochs with a learning rate starting at 0.1. ResNet18 is adopted for CIFAR100 and ImageNet200, while ResNet50 is for ImageNet200. We choose logit-based (MLS), feature-based (KNN), and logit and a combination of both (NNGuide) OOD score functions for AugRevise. Table 4 compares AugRevise with state-of-the-art (SOTA) methods in Openood V1.5 Benchmark. AugRevise improves both logit-based and feature-based methods since it improves both features and logits. AugRevise also improves ID accuracy and outperforms comparing methods. Overall, AugRevise delivers SOTA results in both ID and OOD.

7. Conclusion

In this paper, we systematically diagnose torchvision models in OOD detection and find that label-based data augmentations, label smoothing (LS) and mixup, contribute to the degraded OOD performance of torchvision models with enhanced ID accuracy. Through careful analysis, we find that LS and mixup reduce ID-OOD separation in the logit space, thus hurting OOD detection. To mitigate the negative impact, we proposed AugDelete for post-hoc fix and AugRevise for training-time fix. Both approaches can improve OOD detection performance of logit-based and hybrid methods while maintaining strong ID accuracy.

References

[1] Julian Bitterwolf, Maximilian Müller, and Matthias Hein. In or out? Fixing ImageNet out-of-distribution detection evaluation. In *ICML*, 2023. 1

[2] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 1

[3] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, pages 18613–18624. Curran Associates, Inc., 2020. 1, 3

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 4, 1

[5] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. In *ICLR*, 2023. 2, 3, 4, 8

[6] Xuefeng Du, Zhaoning Wang, Mu Cai, and Sharon Li. Towards unknown-aware learning with virtual outlier synthesis. In *International Conference on Learning Representations*, 2022. 12

[7] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006. 5

[8] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 1

[9] Xiaoyuan Guan, Jiankang Chen, Shenshen Bu, Yuren Zhou, Wei-Shi Zheng, and Ruixuan Wang. Exploiting discrepancy in feature statistic for out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 19858–19866, 2024. 3, 8

[10] Dan Hendrycks and Kevin Gimpel. 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, 2017. 2

[11] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. 5

[12] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. 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, 2019. 2

[13] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020. 1

[14] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*, 2020. 3, 8, 12

[15] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *ICML*, 2022. 1, 2, 3, 8

[16] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The INaturalist species classification and detection dataset. In *CVPR*, 2018. 1

[17] Rui Huang and Yixuan Li. Mos: Towards scaling out-of-distribution detection for large semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8710–8719, 2021. 2, 8, 12

[18] Galadrielle Humblot-Renaux, Sergio Escalera, and Thomas B Moeslund. A noisy elephant in the room: Is your out-of-distribution detector robust to label noise? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22626–22636, 2024. 2

[19] P Izmailov, AG Wilson, D Podoprikhin, D Vetrov, and T Garipov. Averaging weights leads to wider optima and better generalization. In *Uncertainty in Artificial Intelligence*, pages 876–885, 2018. 2

[20] Alex Krizhevsky. Learning multiple layers of features from tiny images. In *arXiv*, 2009. 4, 1

[21] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 1

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

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

[24] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020. 2, 3, 4

[25] Shuo Lu, YingSheng Wang, LuJun Sheng, AiHua Zheng, LinXiao He, and Jian Liang. Recent advances in ood detection: Problems and approaches. *arXiv preprint arXiv:2409.11884*, 2024. 1, 2, 4

[26] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations*, 2023. 12

[27] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 774–782, 2021. 3, 1

[28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop*, 2011. 1

[29] Jaewoo Park, Yoon Gyo Jung, and Andrew Beng Jin Teoh. Nearest neighbor guidance for out-of-distribution detection. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 1686–1695. IEEE, 2023. 1, 2, 3, 8

[30] Francesco Pinto, Harry Yang, Ser Nam Lim, Philip H. S. Torr, and Puneet K. Dokania. Using mixup as a regularizer can surprisingly improve accuracy & out-of-distribution robustness. In *NeurIPS*, 2022. 1, 2, 3, 6, 8, 12

[31] Sudarshan Regmi, Bibek Panthi, Sakar Dotel, Prashnna K Gyawali, Danail Stoyanov, and Binod Bhattarai. T2fnorm: Extremely simple scaled train-time feature normalization for ood detection. *arXiv preprint arXiv:2305.17797*, 2023. 2, 12

[32] Yiyou Sun, Chuan Guo, and Yixuan Li. ReAct: Out-of-distribution detection with rectified activations. In *NeurIPS*, 2021. 2

[33] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR, 2022. 3, 8

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 1, 3

[35] Leitian Tao, Xuefeng Du, Jerry Zhu, and Yixuan Li. Non-parametric outlier synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. 8, 12

[36] Torchvision. How to train state-of-the-art models using torchvision latest primitives, 2024. 4

[37] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *ICLR*, 2022. 1

[38] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*, 2022. 1, 2

[39] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. ViM: Out-of-distribution with virtual-logit matching. In *CVPR*, 2022. 1

[40] Hongxin Wei, Renchunzi Xie an Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *ICML*, 2022. 2, 3, 8, 12

[41] Guoxuan Xia, Olivier Laurent, Gianni Franchi, and Christos-Savvas Bouganis. Understanding why label smoothing degrades selective classification and how to fix it. *arXiv preprint arXiv:2403.14715*, 2024. 1, 2

[42] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. OpenOOD: Benchmarking generalized out-of-distribution detection. In *NeurIPS*, 2022. 2

[43] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 1, 2

[44] Hongyi Zhang. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1, 4, 2

[45] Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Yixuan Li, Ziwei Liu, Yiran Chen, and Hai Li. OpenOOD v1.5: Enhanced benchmark for out-of-distribution detection. In *arXiv*, 2023. 1, 2, 3

[46] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13001–13008, 2020. 3, 1

[47] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 2018. 1

[48] Fei Zhu, Xu-Yao Zhang, Zhen Cheng, and Cheng-Lin Liu. Revisiting confidence estimation: Towards reliable failure prediction. *IEEE TPAMI*, 2023. 1, 2

[49] Yingtian Zou, Vikas Verma, Sarthak Mittal, Wai Hoh Tang, Hieu Pham, Juho Kannala, Yoshua Bengio, Arno Solin, and Kenji Kawaguchi. Mixupe: Understanding and improving mixup from directional derivative perspective. In *Uncertainty in Artificial Intelligence*, pages 2597–2607. PMLR, 2023. 14