Diagnosing Pretrained Models for Out-of-distribution Detection

Supplementary Material

8. Experiment Details

8.1. Openoodv1.5 Dataset Setting

We conduct experiments on OpenOOD v1.5 [45] benchmark. It consists of 4 ID datasets, CIFAR10/100 [20], ImageNet200 [45], and ImageNet-1k [4]. Each ID dataset contains several near-OOD and far-OOD test sets, where the near-OOD test sets are more challenging than the far-OOD ones.

CIFAR10/100 are relatively small datasets. They have the same far-OOD test set containing MNIST [22], SVHN [28], Textures [2], and Places365 [47]. TinyImageNet [21] and CIFAR100 are adopted as near-OOD evaluation sets for CIFAR10; while TinyImageNet and CIFAR10 are adopted for CIFAR100. For network architecture, we adopt the same network backbone, ResNet18 as OpenoodV1.5 [45].

ImageNet-1k is a large-scale dataset consisting of 1281167 training images of 1000 classes. It has 2 near-OOD dataset, SSB-hard [37] and NINCO [1], and 3 far-OOD datasets, iNaturalist [16], Textures [2], and OpenImage-O [39]. To be consistent with Openood V1.5 benchmark, ResNet50, Swin Transformer (Swin-T), and Vision Transformers (ViT) are adopted as pretrained backbone. Besides these models, some additional `torchvision` pre-trained models are evaluated.

ImageNet200 is a subset of ImageNet-1k, which contains 200 classes. It has the same near-OOD and far-OOD datasets as ImageNet-1k. Following OpenoodV1.5 [45], ResNet-18 is adopted as the network backbone.

Evaluation Metrics. FPR@95 and AUROC are adopted to evaluate the OOD performance. FPR@95 is the false positive rate when the true positive rate is 95%, while AUROC is the Area under the receiver operating characteristic curve. Lower FPR@95 and higher AUROC deliver better separation between ID and OOD samples.

8.2. Training Settings

We follow the training setting of Openoodv1.5 [45]. All models are trained 100 epochs with learning rates starting from 0.1. The same Cosine learning rate schedule is adopted as OpenOODv1.5. For CIFAR10/100 and ImageNet200, batch size is 128; for ImageNet-1k, batch size is 512. All the models are repeated with 3 random seeds and the mean results are reported. For AugDelete models, we retrain the fc layers for 15 epochs with learning rates starting from 0.01. Following `torchvision`, both vanilla mixup [44] and cutmix [43] are adopted for models with mixup. We use mixup in the main text of this paper to represent the combination of both vanilla mixup and cutmix. The mixup strength λ of vanilla mixup and cutmix follows Beta distribution $Beta(0.2, 0.2)$ and $Beta(1, 1)$, respectively. The separated impacts of vanilla mixup and cutmix on OOD detection are shown in Table 7, both vanilla mixup and cutmix will degrade OOD performance, which is in accordance with observations in the main text of this paper.

We compare our baseline results (recipe v1) with the OpenOOD v1.5 checkpoint with receipt v1 in Table 5. All the models are repeated with 3 random seeds and the mean results are reported. Our re-implementation shows similar results as OpenOOD v1.5.

8.3. Impact of Augmentations of Torchvision training recipes on OOD Detection

As shown in Table 6, v2 models perform worse than v1 models in OOD detection with logit-based (MLS, GEN, EBO) and hybrid (VIM, NNGuide) score functions. Compared to torchvision v1 models, v2 models adopts several additional data augmentations and training techniques as follows:

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 improves the generalization of neural networks. In experiments, $p^{er} = 0.1$.

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 $\boldsymbol{y}$:

$$\begin{aligned} L_{CE}^{ls}(\boldsymbol{v}, \boldsymbol{y}^{ls}) &= -(\boldsymbol{y}^{ls})^T \log(\sigma(\boldsymbol{v})), \\ \boldsymbol{y}^{ls} &= (1-\beta)\boldsymbol{y} + \beta \boldsymbol{u}, \qquad 0 \leq \beta < 1, \end{aligned} \tag{15}$$

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; in experiments, $\beta = 0.1$.

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:

$$\boldsymbol{x}^{mix} = (1-\lambda)\boldsymbol{x} + \lambda \boldsymbol{x}_1, \qquad \boldsymbol{y}^{mix} = (1-\lambda)\boldsymbol{y} + \lambda \boldsymbol{y}_1. \tag{16}$$

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

$$\begin{aligned} L_{CE}^{mix}(\boldsymbol{v}^{mix}, \boldsymbol{y}^{mix}) &= -(\boldsymbol{y}^{mix})^T \log(\sigma(\boldsymbol{v}^{mix})), \\ \boldsymbol{v}^{mix} &= F(\boldsymbol{x}^{mix}). \end{aligned} \tag{17}$$

Dataset	Implemention	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
CIFAR10	Ours	87.54	60.39	91.04	40.17	94.59
	Openood V1.5	87.52	61.32	91.10	41.68	95.06
CIFAR100	Ours	81.16	55.69	80.75	54.49	77.22
	Openood V1.5	81.05	55.47	79.67	56.73	77.25
ImageNet200	Ours	82.43	60.21	90.84	34.40	86.40
	Openood V1.5	82.90	59.76	91.11	34.03	86.37

Table 5. **Comparison between OpenoodV1.5 models and our re-implementation.** The same receipt v1 is adopted in both implementations.

	VIM	NNguide	GEN	EBO	MLS
ResNet50-v1	82.38	86.68	83.31	82.68	83.02
ResNet50-v2	57.01	65.77	82.46	52.88	72.84
RegNet-v1	83.13	87.09	85.10	82.94	83.36
RegNet-v2	71.49	42.82	84.22	68.70	75.27
MobileNet-v1	79.83	82.59	79.98	79.98	80.35
MobileNet-v2	70.40	69.47	78.36	69.98	77.32

Table 6. **AUROC of ImageNet-1k v1 and v2 models across various backbones and OOD functions.** v2 models show worse AUROC.

Mixup creates a smooth transition between different classes and can improve ID generalization. Following torchvision recipe v2, two variants of mixups, vanilla mixup [44] and cutmix [43], are adopted in experiments. The mixup strength λ of vanilla mixup and cutmix follows Beta distribution $Beta(0.2, 0.2)$ and $Beta(1, 1)$, respectively. The separated impacts of vanilla mixup and cutmix on OOD detection are shown in Table 7 and 9.

Training technique 1: Longer Training (LT) means increasing the training epochs. In experiments, we increase epoch numbers from 100 to 200 when using LT.

Training technique 2: Adjusted Weight Decay means adjust the weight decay value. Specifically, weight decay for the parameters of the normalization layers is set to be 0 in v2 recipes.

Training technique 3: Exponentially Moving Average (EMA) of model parameter: EMA [19] applies the moving average process to model parameters θ following:

$$\theta^{(t+1)} = \gamma\theta^t + (1-\gamma)\theta \tag{18}$$

Where θ^t denotes the model parameters at the t-th step. In experiments, the EMA coefficients $\gamma = 0.99998$ as the torchvision v2 recipes.

Table 10, 11 and 9 show the influence of each data augmentation and training techniques on CIFAR10/100 and ImageNet200, with a single augmentation for each time. Follow the same training setting as [42], each training configure is repeated 3 times, and mean results are reported. Besides the MLS and KNN score function, we also report the results of more OOD score functions in Table 12.

We further analyze the influence of the mixup coefficients (α) and label smoothing (LS) coefficients (β) on OOD detection. Table 7 and 8 show the result of mixup strength and LS coefficients on ImageNet200, respectively. We can observe that i) for both vanilla mixup and cutmixup, larger mixup coefficients α will lead to worse OOD performance; ii) Larger label smoothing coefficients β will also cause larger decrements in OOD AUROC.

	Vanilla Mixup			CutMix		
α	0.2	1.0	10.0	0.2	1.0	10.0
AUROC	86.48	84.70	81.82	83.93	82.45	81.58
ID Acc	87.11	86.27	82.04	86.68	86.30	85.89

Table 7. **Compare Vanilla Mixup and Cutmix on ImageNet200.** AUROC is averaged among near-OOD and far-OOD datasets.

β	0.00	0.10	0.25	0.50	0.95
AUROC	87.01	84.00	84.54	84.72	81.30
ID Acc	86.37	86.87	86.45	86.51	75.30

Table 8. **LS with different smoothing β on ImageNet200.** AUROC is averaged among near-OOD and far-OOD datasets.

8.4. AugDelete for Different Data Augmentation

Table 14, 15 and 13 shows the OOD detection results before and after applying AugDelete under various data augmentations. On all 3 datasets, We observe that AugDelete improves models with label smoothing and mixup by a large margin

Score	Data Augmentaion	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
MLS	v1	82.90	59.76	91.11	34.03	86.37
	v1 +LS	80.74	67.81	87.25	47.82	86.87
	v1 + vanilla mixup	82.47	62.21	90.49	35.81	87.11
	v1 + cutmix	78.65	73.57	86.25	54.45	86.30
	v1 + mixup (both mixups)	81.48	65.49	88.70	43.87	86.86
	v1 + RE	82.57	60.40	90.94	34.52	86.54
	v1 + TA	82.43	59.81	91.15	33.04	87.04
	v1+LS+mixup	79.85	71.43	86.75	51.04	86.92
	v1 + all data augs	79.18	72.55	86.50	52.88	86.85
	v1 + adjusted weight decay (AWD)	82.60	59.98	91.82	30.93	86.37
	v1+ EMA	82.74	59.12	92.83	27.71	86.40
	v1 + LT	83.00	59.94	91.04	33.74	87.01
	v2	78.11	77.39	85.80	61.35	88.08
KNN	v1	81.59	58.26	91.49	31.15	86.37
	v1+LS	81.37	58.46	91.14	31.49	86.87
	v1+vanilla mixup	81.39	59.31	90.84	32.55	87.11
	v1+cutmix	80.49	60.82	89.81	36.24	86.30
	v1+mixup (both mixups)	80.98	60.31	90.17	35.26	86.86
	v1+RE	81.24	58.45	90.31	34.35	86.54
	v1+TA	81.06	59.36	90.43	33.56	87.04
	v1+LS+mixup	81.03	59.88	90.29	34.21	86.92
	v1+all data augs	81.24	58.49	89.70	36.64	86.85
	v1 + adjusted weight decay (AWD)	80.22	62.58	91.09	33.78	86.37
	v1+ EMA	80.53	61.56	92.21	29.20	86.40
	v1 + LT	81.78	58.08	91.28	31.63	87.01
	v2	82.32	57.04	90.92	33.59	88.08

Table 9. **OOD detection results w.r.t data augmentations on ImageNet200.** Label-based data augmentations (LS and mixup) cause a huge OOD performance drop in logits (MLS score), while the negative impact on the feature space is much smaller. Unlike Label-based data augmentations, data-based augmentations (RE and TA) and training techniques (LT, AWD and EMA) have relatively small impact on both logit and feature spaces.

while maintaining the ID accuracy. AugDelete can also slightly improve the OOD Detection performance of RE and TA. However, with AugDelete, models trained with mixup or LS are still worse than the v1 model. This is because AugDelete keeps the pretrained features, thus the negative impact of label smoothing and mixup are not mitigated.

8.5. Finetune the Fully Connected (FC) Layer or the Whole Network in AugDelete

AugDelete deletes all data augmentations and finetunes the FC layer while keeping features fixed, since label smoothing and mixup influence logits more than features. If further finetuning the feature extractor G, can we get additional gain in OOD detection? We finetune the entire network or FC layer for 15 epochs. Table 16 compares the results of finetuning FC or the whole network. Finetuning the entire network only gets marginal or no gains compared to finetuning FC layers. However, finetuning the whole network requires more time and delivers worse ID accuracy, because deleting all data augmentation during finetuning will hurt pretrained features.

8.6. AugRevise with Finetuning or Re-training on ImageNet-1k

AugRevise are assumed to be trained from scratch in the main text. Here we explore whether AugRevise can work by finetuning v2 models. Table 17 shows the results of finetuning v2 models with AugRevise for 10 epochs. Finetuning can significantly improve OOD performance but is worse than re-training, suggesting a trade-off between training burden and OOD performance.

8.7. Fixing Mixup for OOD Detection

Mixup is fixed in AugRevise with L_{vs} loss to increase the separability between ID and mixed samples. Table 18 and 19 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.

We further compare regmixup [30] and mixup-AugRevise

Score	Data Augmentaion	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
MLS	v1	87.52	61.32	91.10	41.68	95.06
	v1 + LS	82.13	93.76	86.05	85.40	95.23
	v1 + mixup	84.39	76.20	88.89	58.10	95.95
	v1 + RE	88.47	56.22	91.99	39.53	95.36
	v1 + TA	92.15	32.37	95.28	19.68	95.51
	v1+all data augs	85.88	77.50	92.07	45.65	95.86
	v2	87.05	72.20	91.22	50.00	96.51
KNN	v1	90.64	33.99	92.96	24.28	95.06
	v1+LS	90.03	36.81	93.12	21.48	95.23
	v1+mixup	91.58	33.38	94.33	21.47	95.95
	v1+RE	91.30	33.01	93.96	22.15	95.36
	v1+TA	92.32	28.88	95.26	18.83	95.51
	v1+all data augs	92.82	27.92	96.34	16.84	95.86
	v2	93.30	27.57	96.66	14.94	96.51

Table 10. **OOD detection results w.r.t data augmenations on CIFAR10.**

Score	Data Augmentaion	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
MLS	v1	81.05	55.47	79.67	56.73	77.25
	v1 + LS	80.35	58.06	78.44	60.88	77.78
	v1 + mixup	77.57	73.10	72.68	78.65	79.69
	v1 + RE	80.96	56.48	82.31	52.20	76.91
	v1 + TA	81.92	55.78	82.91	50.57	78.78
	v1+all data augs	78.34	71.64	74.85	71.46	79.89
	v2	78.44	74.01	72.39	76.96	81.19
KNN	v1	80.18	61.23	82.40	53.65	77.26
	v1+LS	78.84	61.34	81.24	56.14	77.78
	v1+mixup	78.99	60.55	83.08	52.68	79.69
	v1+RE	79.91	62.27	83.87	50.99	76.91
	v1+TA	79.98	63.89	85.53	46.86	78.78
	v1+all data augs	79.69	60.98	85.09	48.70	79.89
	v2	80.06	60.34	81.64	54.37	81.19

Table 11. **OOD detection results w.r.t data augmentations on CIFAR100.**

Data Augmentaion	VIM	NNGuide	GEN	EBO	MLS
v1	86.98	87.83	87.49	86.68	87.01
v1+RE	86.56	86.44	87.31	86.43	86.76
v1+TA	86.53	86.68	87.34	86.44	86.79
v1+LS	83.31	71.48	85.41	82.15	84.00
v1+mixup	84.52	78.43	86.09	84.06	85.09

Table 12. **AUROC of ResNet18 with various data augmentations and OOD functions on ImageNet200.** The average values of near-OOD and far-OOD AUROCs are reported. LS/mixup harms the AUROC of various OOD functions.

Data Augmentaion	AugDelete	Near-OOD		Far-OOD		ID ACC
		AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	↑
v1	✗	82.90	59.76	91.11	34.04	86.37
	✓	82.92	59.24	90.51	35.82	86.26
v1+LS	✗	80.74	67.81	87.25	47.82	86.87
	✓	81.86	64.21	87.85	45.22	86.94
v1+mixup	✗	81.48	65.49	88.70	43.87	86.86
	✓	82.51	62.44	89.08	41.88	86.93
v1+RE	✗	82.57	60.40	90.94	34.52	86.54
	✓	83.06	59.43	90.69	35.49	86.63
v1+TA	✗	82.43	59.81	91.15	33.04	87.04
	✓	82.97	58.91	90.85	33.90	87.04
v1+all data augs	✗	79.18	72.55	86.50	52.88	86.85
	✓	82.33	62.70	89.30	40.78	87.14
v1+LS+mixup	✗	79.85	71.43	86.75	51.04	86.92
	✓	81.83	65.66	88.09	44.92	87.15
v2	✗	78.11	77.39	85.80	61.35	88.08
	✓	82.79	62.83	89.91	41.54	88.03

Table 13. **AugDelete for models trained with different data augmentations on ImageNet200.**

Data Augmentaion	AugDelete	Near-OOD		Far-OOD		ID ACC
		AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	↑
v1	✗	87.52	61.32	91.10	41.67	95.06
	✓	88.01	56.84	91.40	37.56	95.01
v1+LS	✗	84.39	76.20	88.89	58.10	95.95
	✓	90.04	45.44	92.13	36.80	95.92
v1+mixup	✗	82.13	93.76	86.05	85.40	95.23
	✓	89.91	40.30	92.54	26.72	95.21
v1+RE	✗	88.47	56.22	91.99	39.53	95.36
	✓	89.13	50.47	92.48	34.87	95.42
v1+TA	✗	92.15	32.37	95.28	19.68	95.51
	✓	92.34	29.88	95.14	19.50	95.46
v1+all data augs	✗	85.88	77.50	92.07	45.65	95.86
	✓	91.61	34.61	94.66	24.81	95.85
v2	✗	87.05	72.20	91.22	50.00	96.51
	✓	92.57	30.86	94.86	25.39	96.54

Table 14. **AugDelete for models trained with different data augmentations on CIFAR10.**

w.r.t different mixup coefficients α in Table 20. It can be observed that RegMixup delivers worse performance than the v1 model when $\alpha < 10$ (lower α indicates lower mixup strength). In contrast, AugRevise-mixup always outperforms v1 models, suggesting the benefits of L_{vs}.

8.8. Compare AugDelete and AugRevise

We compare AugDelete and AugRevise in Table 21 and Table 22. AugRevise outperforms AugDelete and vanilla v1 models in both the ID classification and OOD detection. However, adding label smoothing in AugRevise will

Data Augmentaion	AugDelete	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
v1	✗	81.05	55.47	79.67	56.73	77.25
	✓	80.96	55.60	80.26	56.01	77.16
v1 + LS	✗	80.35	58.06	78.44	60.88	77.78
	✓	80.81	55.89	79.67	58.17	77.93
v1 + mixup	✗	77.57	73.10	72.68	78.65	79.69
	✓	80.46	62.21	77.81	64.60	79.75
v1 + RE	✗	80.96	56.48	82.31	52.20	76.91
	✓	81.04	56.31	82.84	51.10	76.81
v1 + TA	✗	81.92	55.78	82.91	50.57	78.78
	✓	81.93	55.84	82.51	51.03	78.62
v1+all data augs	✗	78.34	71.64	74.85	71.46	79.89
	✓	80.89	62.21	80.33	56.96	80.03
v2	✗	78.44	74.01	72.39	76.96	81.19
	✓	82.31	56.06	77.87	60.96	81.17

Table 15. **AugDelete for models trained with different data augmentations on CIFAR100.**

Dataset	Models	Finetuning	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
ImageNet-1k	ResNet50-v2	✗	69.20	86.00	76.47	83.49	80.92
		FC	78.69	70.05	87.47	56.18	80.31
		Network	78.47	70.74	88.98	45.45	80.08
ImageNet-1k	Swin-T-v2	✗	75.66	80.76	84.80	67.81	81.59
		FC	81.01	69.06	90.96	37.79	81.30
		Network	77.58	72.76	87.42	51.26	80.45

Table 16. **Comparing finetuning the fully connected (FC) layer and finetuning the whole network.**

decrease the OOD performance in both near-OOD and far-OOD detection, suggesting that label smoothing should be removed in AugRevise.

8.9. OOD detection with Various Pretrained Network Architectures

We apply AugDelete to pretrained models with different network architectures including convolutional neural networks (CNNs) and transformers. Table 24 presents the ID accuracy and OOD performance with/without AugDelete. We see that AugDelete improves the OOD detection of both CNNs and transformers while maintaining ID accuracy. In this paper, we focus on torchvision because it is widely used in OOD research. Preliminary results on TIMM show our findings do generalize (Table 23), but there are no baselines for comparison because the most common OOD benchmark (OpenOOD) uses mainly torchvision.

8.10. Comparison with various post-hoc OOD Detection methods.

We combine AugDelete into various post-hoc OOD detection methods on Openood V1.5. Both logit-based and feature-and-logit-based methods are considered for AugDelete. Note that AugDelete has no effect on the feature-based method since it does not change the features. Table 25 shows the results with ResNet-v2, Swin-T-v2 and ViT-B-16 prerained networks. More results concerning network architectures are in the appendix. We can see that AugDelete can improve both methods by a large margin since AugDelete fixes the logits hurt by label smoothing and mixup.

	Near-OOD		Far-OOD		ID ACC	Epochs
	AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	↑	
ResNet50-v2	69.20	86.00	76.47	83.49	80.92	600
ResNet50-v2+Finetune	79.33	66.17	89.88	41.85	79.86	600+10
ResNet50-v2+Re-training	79.23	62.37	90.30	35.74	77.70	100

Table 17. **Finetune/Re-training ResNet50-v2 models on ImageNet-1k with AugRevise.**

Data Augmentaion	**Loss**	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
v1	L_{CE}	87.52	61.32	91.10	41.68	95.06
v1 + mixup	L_{CE}^{mix}	84.39	76.20	88.89	58.10	95.95
v1 + regmixup	L_{CE}^{rmix}	89.19	53.81	93.18	32.93	96.27
v1 + mixup-AugRevise	L_{CE}^{rvmix}	90.56	44.71	94.20	25.55	96.58
v1	L_{CE}	81.05	55.47	79.67	56.73	77.25
v1 + mixup	L_{CE}^{mix}	77.57	73.10	72.68	78.65	79.69
v1 + regmixup	L_{CE}^{rmix}	82.22	56.53	82.40	54.80	80.43
v1 + mixup-AugRevise	L_{CE}^{rvmix}	83.34	51.56	85.20	45.93	81.22

Table 18. **The results of fixing mixup on CIFAR10/100.** The top/bottom halves are for CIFAR10/CIFAR100, seperately.

Training Recipe	**Loss**	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
v1	L_{CE}	82.90	59.76	91.11	34.04	86.37
v1+mixup	L_{CE}^{mix}	80.74	67.81	87.25	47.82	86.87
v1+regmixup	L_{CE}^{rmix}	82.85	61.58	91.10	34.48	87.58
v1+mixup-AugRevise	L_{CE}^{rvmix}	83.88	54.26	91.57	29.91	87.28

Table 19. **The results of fixing mixup on ImageNet200.**

Training Recipe	**Loss**	**mixup coefficients** α	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
v1+regmixup	L_{CE}^{rmix}	0.2	78.28	77.58	84.71	59.84	87.01
v1+regmixup	L_{CE}^{rmix}	1	78.88	76.29	85.36	59.26	87.20
v1+regmixup	L_{CE}^{rmix}	10	82.85	61.58	91.10	34.48	87.58
v1+mixup-AugRevise	L_{CE}^{rvmix}	0.2	83.75	54.65	90.92	32.37	87.04
v1+mixup-AugRevise	L_{CE}^{rvmix}	1	83.86	54.35	91.14	31.47	87.21
v1+mixup-AugRevise	L_{CE}^{rvmix}	10	83.88	54.26	91.57	29.91	87.28

Table 20. **The results of fixing mixup on ImageNet200.**

8.11. Detailed Results of AugRevise for Training-Time Model Enhancement

We train models from scratch with AugRevise on ImageNet200/1k and CIFAR10/100 datasets, following the same training setting as OpenoodV1.5. ResNet18 is adopted for CIFAR10/100 and ImageNet200, while ResNet50 is for ImageNet200. Note that AugRevise for ImageNet-1k is trained 100 epochs as ResNet50-v1 instead of 600 epochs as ResNet50-v2. We choose logit-based, feature-based, and logit-and-feature-based OOD score functions for AugRevise. Table 26 and 27 compares AugRevise with state-of-the-art (SOTA) methods in Openood V1.5 Benchmark. AugRevise

Train Recipe	Near-OOD AUROC ↑	Near-OOD FPR@95 ↓	Far-OOD AUROC ↑	Far-OOD FPR@95 ↓	ID ACC ↑
v1	82.90	59.76	91.11	34.04	86.37
v2*	78.92	72.84	86.64	52.10	86.74
v2*+AugDelete	81.87	63.29	89.27	40.44	86.68
v2*+AugRevise	84.07	54.02	91.70	29.41	87.67
v2*+AugRevise +LS	83.88	54.64	90.47	33.10	87.33
v2	78.11	77.39	85.80	61.35	88.08
v2+AugDelete	82.79	62.83	89.91	41.54	88.03
v2+AugRevise	84.49	53.84	92.06	29.66	88.14

Table 21. **Compare AugDelete and AugRevise on ImageNet200.** For fair comparison, v2* is trained for 100 epochs as v1. v2 is trained for 200 epochs.

	VIM	NNGuide	GEN	EBO	MLS
ResNet50-v2	57.01	65.77	82.46	52.88	72.84
ResNet50+AugDelete	83.10	77.54	83.10	81.83	83.08
ResNet50+AugRevise	84.78	87.17	84.87	84.84	84.77

Table 22. **AUROC of AugDelete and AugRevise across different OOD functions on ImageNet-1k with ResNet50 backbone.**

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 outperforms both post-hoc and training-based methods in ID and OOD.

	VIM	NNGuide	GEN	EBO	MLS
RN50-a1	81.66	82.95	84.07	81.81	82.43
RN50-a1+AugDelete	88.93	86.11	89.55	89.60	88.14

Table 23. **Applying AugDelete to ResNet50.a1_in1k in timm library.** AugDelete can improve the AUROC of ResNet50.a1_in1k.

Pre-trianed Models	**Near-OOD**		**Far-OOD**		ID ACC
	AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	↑
ResNet50-v1	76.46	67.84	89.58	38.20	76.18
ResNet50-v2	69.20	86.00	76.47	83.49	80.92
MobileNetv2-v1	72.01	73.01	88.68	38.54	71.91
MobileNetv2-v2	73.89	74.33	80.74	62.88	72.22
ResNetXt50-v1	78.49	67.73	89.37	40.64	77.64
ResNetXt50-v2	72.06	84.55	79.28	82.66	81.22
WideResNet50-v1	78.69	67.93	89.02	41.22	78.50
WideResNet50-v2	66.56	87.58	68.32	91.84	81.64
RegNet-v1	78.58	70.92	88.13	45.16	80.44
RegNet-v2	72.13	89.20	78.41	91.49	82.96
ConvNext-v2	76.44	74.10	84.58	53.83	83.59
Swin-T-v2	75.66	80.76	84.80	67.81	81.59
ViT-B-16-v2	68.30	92.25	83.54	79.23	81.14
ResNet50-v2 + AugDelete	78.69	70.05	87.47	56.18	80.31
MobileNetv2-v2 + AugDelete	75.45	69.93	83.63	56.23	70.24
ResNetXt50-v2 + AugDelete	80.27	69.48	88.64	50.59	80.92
WideResNet50-v2 + AugDelete	76.67	77.19	83.84	71.38	81.45
RegNet-v2 + AugDelete	76.74	84.57	87.79	66.05	82.86
ConvNext-v2 + AugDelete	79.41	67.19	88.88	45.69	82.98
Swin-T-v2 + AugDelete	81.01	69.06	90.96	37.79	81.30
VIT-B-16-v2 + AugDelete	79.83	69.84	91.87	30.31	81.00

Table 24. **AugDelete w.r.t various pretrained Networks on ImageNet-1k.** The top half is before AugDelete while the bottom half is after AugDelete.

Pre-trianed Models	Method	AugDelete	Near-OOD		Far-OOD	
			AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓
ResNet50-v2	MLS	✗	69.20	86.00	76.47	83.49
		✓	78.69	70.05	87.47	56.18
	EBO	✗	54.39	89.23	51.36	90.72
		✓	76.84	72.46	86.82	58.72
	MSP	✗	72.53	82.21	81.48	72.28
		✓	77.64	66.92	85.58	53.98
	ASH	✗	54.75	90.01	52.32	91.16
		✓	76.50	72.32	86.90	57.91
	KNN	✗	70.76	73.48	89.07	36.71
	NNGuide	✗	61.16	82.25	70.37	61.89
		✓	71.38	70.84	83.71	45.60
Swin-T-v2	MLS	✗	75.66	80.76	84.80	67.81
		✓	81.01	69.06	90.96	37.79
	EBO	✗	73.23	83.31	81.32	75.59
		✓	80.78	71.74	91.40	38.32
	MSP	✗	76.75	71.06	86.30	49.16
		✓	78.88	64.08	88.28	43.68
	ASH	✗	67.91	85.86	71.93	82.68
		✓	78.46	76.89	89.30	45.75
	KNN	✗	71.62	71.76	89.37	34.12
	NNGuide	✗	67.92	84.99	85.36	50.77
		✓	71.95	83.48	90.07	42.44
ViT-B-16-v2	MLS	✗	68.30	92.25	83.54	79.23
		✓	79.83	69.84	91.87	30.31
	EBO	✗	62.41	93.19	78.98	85.35
		✓	80.13	70.90	92.69	27.94
	MSP	✗	73.52	81.85	86.04	51.69
		✓	77.77	65.34	89.00	39.64
	ASH	✗	57.82	93.65	73.08	85.39
		✓	79.71	71.99	93.01	27.96
	KNN	✗	74.11	70.47	90.81	31.93
	NNGuide	✗	60.40	89.89	81.74	59.86
		✓	69.83	85.66	90.36	43.40

Table 25. **AugDelete w.r.t various OOD scores on ImageNet-1k.**

ID Dataset	Method	AugRevise	Near-OOD		Far-OOD		ID ACC
			AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	↑
CIFAR10	MLS	✗	87.52	61.32	91.10	41.67	95.06
		✓	92.78	30.37	95.28	20.16	96.73
	EBO	✗	87.58	61.32	91.21	41.70	95.06
		✓	92.86	30.43	95.46	20.11	96.73
	MSP	✗	88.03	48.18	90.73	31.72	95.06
		✓	92.44	28.62	94.56	19.59	96.73
	ASH	✗	87.54	61.23	91.13	41.60	95.06
		✓	92.82	30.42	95.35	20.11	96.73
	KNN	✗	90.64	33.99	92.96	24.28	95.06
		✓	93.69	27.46	96.22	16.20	96.73
	NNGuide	✗	83.54	78.57	86.95	65.86	95.06
		✓	92.83	32.41	95.60	20.48	96.73
CIFAR100	MLS	✗	81.05	55.46	79.67	56.72	77.26
		✓	84.05	51.49	83.33	50.22	82.10
	EBO	✗	80.91	55.60	79.77	56.58	77.26
		✓	83.88	51.66	84.29	49.59	82.10
	MSP	✗	80.27	54.79	77.76	58.70	77.26
		✓	83.37	52.06	81.63	52.17	82.10
	ASH	✗	81.07	55.99	79.92	55.69	77.26
		✓	83.78	52.15	84.51	49.19	82.10
	KNN	✗	80.18	61.23	82.40	53.65	77.26
		✓	81.88	60.24	85.88	46.68	82.10
	NNGuide	✗	80.27	58.36	81.41	56.66	77.26
		✓	83.92	52.89	85.10	46.14	82.10
ImageNet200	MLS	✗	82.90	59.76	91.11	34.04	86.37
		✓	84.07	54.02	91.70	29.41	87.67
	EBO	✗	82.50	60.22	90.86	34.86	86.37
		✓	83.68	54.49	91.85	29.66	87.67
	MSP	✗	83.34	54.83	90.13	35.43	86.37
		✓	84.09	53.91	91.69	29.38	87.67
	ASH	✗	82.76	59.82	91.63	32.68	86.37
		✓	83.94	53.82	92.68	26.85	87.67
	KNN	✗	81.59	58.26	91.49	31.15	86.37
		✓	81.34	56.45	92.65	26.88	87.67
	NNGuide	✗	82.54	63.10	93.11	30.70	86.37
		✓	84.34	54.15	94.29	20.98	87.67
ImageNet-1k	MLS	✗	76.46	67.84	89.58	38.20	76.18
		✓	79.23	62.37	90.30	35.74	77.70
	EBO	✗	75.89	68.56	89.47	38.40	76.18
		✓	78.96	62.59	90.72	34.57	77.70
	MSP	✗	76.02	65.67	85.23	51.47	76.18
		✓	79.25	62.36	90.31	35.75	77.70
	ASH	✗	76.41	66.85	91.52	32.39	76.18
		✓	79.34	61.09	91.93	30.91	77.70
	KNN	✗	71.10	70.87	90.18	34.13	76.18
		✓	72.60	69.82	92.52	28.67	77.70
	NNGuide	✗	78.80	63.89	94.56	25.73	76.18
		✓	80.90	59.92	93.44	27.35	77.70

Table 26. **AugRevise w.r.t various OOD scores on CIFAR10/100 and ImageNet200/1k.**

Dataset	Methods	Near-OOD		Far-OOD		ID ACC
		AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	↑
CIFAR10	T2FNorm+T2FNorm [31]	92.79	26.47	96.98	12.75	94.69
	LogitNorm+MSP [40]	92.33	29.34	96.74	13.81	94.30
	VOS+EBO [6]	87.70	57.03	90.83	40.43	94.31
	NPOS+KNN [35]	89.78	32.64	94.07	20.59	—
	CIDER+KNN [26]	90.71	32.11	94.71	20.72	—
	MOS+MOS [17]	71.45	78.72	76.41	62.90	94.83
	AugMix+MSP [14]	89.43	37.68	91.66	27.00	95.01
	RegMixup+MSP [30]	87.47	48.78	90.25	36.30	95.75
	AugRevise +MLS	92.78	30.37	95.28	20.16	96.73
	AugRevise +KNN	93.69	27.46	96.22	16.20	96.73
	AugRevise +NNGuide	92.83	32.41	95.60	20.48	96.73
CIFAR100	T2FNorm+T2FNorm [31]	79.84	58.47	82.73	51.25	76.43
	LogitNorm+MSP [40]	78.47	62.89	81.53	53.61	76.34
	VOS+EBO [6]	80.93	55.56	81.32	53.70	77.20
	NPOS+KNN [35]	78.35	63.35	82.29	51.13	—
	CIDER+MSP [26]	73.10	72.02	80.49	54.22	—
	MOS+MOS [17]	80.40	56.05	80.17	57.28	76.98
	AugMix+MSP [14]	79.36	56.30	77.18	58.36	76.45
	RegMixup+MSP [30]	80.83	56.12	79.04	57.50	79.32
	AugRevise +MLS	84.05	51.49	83.33	50.22	82.10
	AugRevise +KNN	81.88	60.24	85.88	46.68	82.10
	AugRevise +NNGuide	83.92	52.89	85.10	46.14	82.10
ImageNet200	T2FNorm+T2FNorm [31]	83.00	55.01	93.55	25.73	86.87
	LogitNorm+MSP [40]	82.66	54.46	93.04	26.11	86.04
	VOS+EBO [6]	82.51	59.89	91.00	34.01	86.23
	NPOS+KNN [35]	79.40	62.09	94.49	21.76	—
	CIDER+KNN [26]	80.58	60.10	90.66	30.17	—
	MOS+MOS [17]	69.84	71.60	80.46	51.56	85.60
	AugMix+MSP [14]	83.49	54.97	90.68	33.42	87.01
	RegMixup+MSP [30]	84.13	68.92	90.81	30.31	87.25
	AugRevise +MLS	84.07	54.02	91.70	29.41	87.67
	AugRevise +KNN	81.34	56.45	92.65	26.88	87.67
	AugRevise +NNGuide	84.34	54.15	94.29	20.98	87.67
ImageNet-1k	T2FNorm+T2FNorm [31]	73.08	69.14	91.92	31.24	76.76
	LogitNorm+MSP [40]	74.62	68.56	91.54	31.32	76.45
	VOS+EBO [6]	—	—	—	—	—
	NPOS+KNN [35]	—	—	—	—	—
	CIDER+KNN [26]	68.97	71.69	92.18	28.69	—
	MOS+MOS [17]	72.85	76.31	82.75	52.63	72.81
	AugMix+MSP [14]	77.49	64.45	86.67	46.94	77.63
	RegMixup+MSP [30]	77.04	65.33	86.31	48.91	76.68
	AugRevise +MLS	79.23	62.37	90.30	35.74	77.70
	AugRevise +KNN	72.60	69.82	92.52	28.67	77.70
	AugRevise +NNGuide	80.90	59.92	93.44	27.35	77.70

Table 27. **Comapre AugRevise with training-based OOD detection methods on CIFAR10/100 and ImageNet200/1k.**

9. Derivation of Proposition 4.1

We use i^* denote the index of the maximal logit, $\Delta \boldsymbol{v}[i^*]$ to denote 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.

The derivation contains 2 steps. First, we illustrate the relationship between the one-step update of the maximal logit ($\Delta \boldsymbol{v}[i^*]$) and the gradient. Then, we compute the difference between gradients. With the results of the previous steps, we finally prove the proposition.

9.1. Relating the Increment of the Maximal Logit to Gradients

We follow the loss and network definition as equation 3 and 2. Let θ denote the parameter of the feature extraction network G, and η denote the learning rate. When applying one-step gradient descent, the network parameters $\mathbf{W}$, $\boldsymbol{b}$, and θ are directly updated, then the update of the network parameters will be reflected on the logits. According to the chain rule, the total derivative $\Delta \boldsymbol{v}[i^*]$ of the maximal logits $\boldsymbol{v}[i^*]$ is:

$$
\begin{aligned}
\Delta \boldsymbol{v}[i^*] &= \boldsymbol{f}^T \Delta \mathbf{W}[i^*,:] + \mathbf{W}[i^*,:]^T \Delta \boldsymbol{f} + \Delta \boldsymbol{b}[i^*] \\
&= \boldsymbol{f}^T \Delta \mathbf{W}[i^*,:] + \mathbf{W}[i^*,:]^T (\frac{\partial \boldsymbol{f}}{\partial \theta})^T \Delta \theta + \Delta \boldsymbol{b}[i^*] \\
&= \boldsymbol{f}^T (-\eta \frac{\partial L_{ce}}{\partial \mathbf{W}[i^*,:]}) + \mathbf{W}[i^*,:]^T (\frac{\partial \boldsymbol{f}}{\partial \theta})^T (-\eta \frac{\partial L_{ce}}{\partial \theta}) + (-\eta \frac{\partial L_{ce}}{\partial \boldsymbol{b}[i^*]}) \\
&= -\eta \{ \boldsymbol{f}^T \frac{\partial \boldsymbol{v}[i^*]}{\partial \mathbf{W}[i^*,:]} \frac{\partial L_{ce}}{\partial \boldsymbol{v}[i^*]} + \mathbf{W}[i^*,:]^T (\frac{\partial \boldsymbol{f}}{\partial \theta})^T (\frac{\partial \boldsymbol{f}}{\partial \theta} \sum_{k=1}^{C} \frac{\partial \boldsymbol{v}[k]}{\partial \boldsymbol{f}} \frac{\partial L_{ce}}{\partial \boldsymbol{v}[k]}) + \frac{\partial \boldsymbol{v}[i^*]}{\partial \boldsymbol{b}[i^*]} \frac{\partial L_{ce}}{\partial \boldsymbol{v}[i^*]} \} \\
&= -\eta \{ (\boldsymbol{f}^T \boldsymbol{f} + 1) \frac{\partial L_{ce}}{\partial \boldsymbol{v}[i^*]} + \sum_{k=1}^{C} \mathbf{W}[i^*,:]^T (\frac{\partial \boldsymbol{f}}{\partial \theta})^T (\frac{\partial \boldsymbol{f}}{\partial \theta}) \mathbf{W}[k,:] \frac{\partial L_{ce}}{\partial \boldsymbol{v}[k]}) \} \\
&\approx -\eta \{ \boldsymbol{f}^T \boldsymbol{f} + 1 + \mathbf{W}[i^*,:]^T (\frac{\partial \boldsymbol{f}}{\partial \theta})^T (\frac{\partial \boldsymbol{f}}{\partial \theta}) \mathbf{W}[i^*,:] \} \frac{\partial L_{ce}}{\partial \boldsymbol{v}[i^*]} \\
&\propto -\frac{\partial L_{ce}}{\partial \boldsymbol{v}[i^*]}
\end{aligned}
\tag{19}
$$

where "$[j]$" denotes take the j-th element of a vector, and "$[k,:]$" denotes take the k-th row of a matrix.

Remark: During training, the gradient of the maximal logits tends to be much larger than that of the other logits, *i.e.* $|\frac{\partial L_{ce}}{\partial \boldsymbol{v}[i^*]}| >> |\frac{\partial L_{ce}}{\partial \boldsymbol{v}[k]}| (k \neq i^*)$.

According to equation 19, we have

$$
\begin{aligned}
\Delta \boldsymbol{v}[i^*] - \Delta \boldsymbol{v}^{ls/mix}[i^*] &\approx \eta (\boldsymbol{f}^T \boldsymbol{f} + 1) (\frac{\partial L_{ce}^{ls/mix}}{\partial \boldsymbol{v}[i^*]} - \frac{\partial L_{ce}}{\partial \boldsymbol{v}[i^*]}) \\
&\propto \frac{\partial L_{ce}^{ls/mix}}{\partial \boldsymbol{v}[i^*]} - \frac{\partial L_{ce}}{\partial \boldsymbol{v}[i^*]}
\end{aligned}
\tag{20}
$$

9.2. Difference of Gradiants

For cross entropy loss L_{ce} defined in equation 3, we can compute the partial derivative w.r.t the j-th logits $v[j]$ as:

$$
\frac{\partial L_{ce}}{\partial \boldsymbol{v}}[j] = -\boldsymbol{y}[j] + \boldsymbol{p}[j], \qquad \boldsymbol{p}[i] = \frac{e^{\boldsymbol{v}[j]}}{\sum_{k=1}^{C} e^{\boldsymbol{v}[k]}},
\tag{21}
$$

Similarly, the gradient for label smoothing w.r.t the j-th logits $v[j]$ is

$$
\frac{\partial L_{ce}^{ls}}{\partial \boldsymbol{v}}[j] = -\boldsymbol{y}^{ls}[j] + \boldsymbol{p}[j],
\tag{22}
$$

compare equation 21 and 22, we can get the gradient difference

$$
\frac{\partial L_{ce}^{ls}}{\partial \boldsymbol{v}}[j] - \frac{\partial L_{ce}}{\partial \boldsymbol{v}}[j] = \boldsymbol{y}[j] - \boldsymbol{y}^{ls}[j],
\tag{23}
$$

For mixup, we adopt the first-order approximation derived by [49], i.e.,

$$L_{ce}^{mix} \approx L_{ce} + (\boldsymbol{y} - \sigma(\boldsymbol{v}))^T \boldsymbol{v} \tag{24}$$

With equation 24, we can compute the gradient difference $\frac{\partial L_{ce}^{mix}}{\partial \boldsymbol{v}[j]} - \frac{\partial L_{ce}}{\partial \boldsymbol{v}[j]}$:

$$\frac{\partial L_{ce}^{mix}}{\partial \boldsymbol{v}}[j] - \frac{\partial L_{ce}}{\partial \boldsymbol{v}}[j] = (\boldsymbol{y}[j] - \boldsymbol{p}[j]) + p[j] \sum_{k=1}^{C} p[k](\boldsymbol{v}[k] - \boldsymbol{v}[j]) \tag{25}$$

Now we analyze the gradient of the maximal logit $\boldsymbol{v}[i^*]$. Take $j = i^*$ into equation 23 and combining the definition of label smoothing in equation 7, we have

$$\frac{\partial L_{ce}^{ls}}{\partial \boldsymbol{v}}[i^*] - \frac{\partial L_{ce}}{\partial \boldsymbol{v}}[i^*] = \beta(1 - \frac{1}{C}) > 0, \tag{26}$$

where β is the label smoothing coefficient and C is the number of classes. Similarly, take $j = i^*$ into equation 25, we have

$$\begin{aligned}
\frac{\partial L_{ce}^{mix}}{\partial \boldsymbol{v}}[i^*] - \frac{\partial L_{ce}}{\partial \boldsymbol{v}}[i^*] &= (1 - \boldsymbol{p}[i^*]) + p[i^*] \sum_{k=1}^{C} p[k](\boldsymbol{v}[k] - \boldsymbol{v}[i^*]) \\
&= \sum_{k=1, k\neq i^*}^{C} \boldsymbol{p}[k] + \sum_{k=1, k\neq i^*}^{C} p[i^*]p[k](\boldsymbol{v}[k] - \boldsymbol{v}[i^*]) \\
&= \sum_{k=1, k\neq i^*}^{C} \boldsymbol{p}[k] + p[i^*]p[k](\boldsymbol{v}[k] - \boldsymbol{v}[i^*]) \\
&= \sum_{k=1, k\neq i^*}^{C} \boldsymbol{p}[k] + p[i^*]p[k] \log(\frac{\boldsymbol{p}[k]}{\boldsymbol{p}[i^*]}) \\
&= \sum_{k=1, k\neq i^*}^{C} \boldsymbol{p}[k]\{1 + p[i^*] \log(\frac{\boldsymbol{p}[k]}{\boldsymbol{p}[i^*]})\} \geq 0
\end{aligned} \tag{27}$$

Remark: The informative gradients come from the wrong predicted logits, i.e., $p[k] \geq \boldsymbol{p}[i^*]$. When $p[k] \geq \boldsymbol{p}[i^*]$ holds, $\boldsymbol{p}[k]\{1 + p[i^*] \log(\frac{\boldsymbol{p}[k]}{\boldsymbol{p}[i^*]})\}$ is large than 0. On the contrary, for the correct predicted logits, $p[k] << \boldsymbol{p}[i^*]$, then the term $\boldsymbol{p}[k]\{1 + p[i^*] \log(\frac{\boldsymbol{p}[k]}{\boldsymbol{p}[i^*]})\}$ is close to 0.

Combining equation 20, 26, and 27, we can reach the conclusion:

$$\Delta \boldsymbol{v}[i^*] - \Delta \boldsymbol{v}^{ls/mix}[i^*] \propto \frac{\partial L_{ce}^{ls/mix}}{\partial \boldsymbol{v}[i^*]} - \frac{\partial L_{ce}}{\partial \boldsymbol{v}[i^*]} \geq 0. \tag{28}$$

10. Derivation of 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, $r_i = \frac{s_i - s_i^{aug}}{s_i}$ ($r_o = \frac{s_o - s_o^{aug}}{s_o}$) denote the relative decrement of s_i^{aug} (s_o^{aug}), $p_i(x)$ ($p_o(x)$) denote the probability density function of s_i (s_o) at the value of x, and $p_i^{aug}(x)$ ($p_o^{aug}(x)$) denote that of s_i^{aug} (s_o^{aug}) at the value of x. Assume s_i is non-negative.

We can connect s_i (s_o) and s_i^{aug} (s_o^{aug}) by r_i (r_o) as

$$s_i^{aug} = s_i \cdot (1 - r_i), \quad s_o^{aug} = s_o \cdot (1 - r_o); \tag{29}$$

and also connect the probability density function as

$$p_i^{aug}(x) = \frac{1}{(1 - r_i)} p_i(\frac{x}{1 - r_i}), \quad p_o^{aug}(x) = \frac{1}{(1 - r_o)} p_o(\frac{x}{1 - r_o}). \tag{30}$$

We can derive the probability $Prob(s_i \geq s_o)$ as:

$$\begin{aligned} Prob(s_i \geq s_o) &= \int_0^{+\infty} Prob(s_o < x) \cdot p_i(x)dx \\ &= \int_0^{+\infty} p_i(x)dx \int_{-\infty}^{x} p_o(y)dy \\ &= \int_0^{+\infty} dx \int_{-\infty}^{x} p_i(x)p_o(y)dy \end{aligned} \tag{31}$$

Similarly, we can derive the probability $Prob(s_i^{aug} \geq s_o^{aug})$ as:

$$Prob(s_i^{aug} \geq s_o^{aug}) = \int_0^{+\infty} dx \int_{-\infty}^{x} p_i^{aug}(x)p_o^{aug}(y)dy \tag{32}$$

Combining eq. 30 and 31, we have

$$\begin{aligned} Prob(s_i^{aug} \geq s_o^{aug}) &= \int_0^{+\infty} dx \int_{-\infty}^{x} \frac{1}{(1-r_i)} p_i(\frac{x}{1-r_i}) \cdot \frac{1}{(1-r_o)} p_o(\frac{y}{1-r_o})dy \\ &= \int_0^{+\infty} dx^{'} \int_{-\infty}^{\frac{1-r_i}{1-r_o}x} p_i(x^{'}) \cdot p_o(y^{'})dy^{'} \qquad (\text{Let} \quad x^{'} = \frac{x}{1-r_i} \quad \text{and} \quad y^{'} = \frac{y}{1-r_o}) \end{aligned} \tag{33}$$

Combine eq. 31 and 33, we have

$$\begin{aligned} \delta P &= Prob(s_i \geq s_o) - Prob(s_i^{aug} \geq s_o^{aug}) \\ &= \int_0^{+\infty} dx \int_{\frac{1-r_i}{1-r_o}x}^{x} p_i(x)p_o(y)dy \end{aligned} \tag{34}$$

If $1 > r_i > r_o$, we have $0 < \frac{1-r_i}{1-r_o} < 1$ and considering the probability density function is always non-negative ($p_i(x) \geq 0$ and $p_o(y) \geq 0$), we can derive from eq. 34 that:

$$\begin{aligned} \delta P &= \int_0^{+\infty} dx \int_{\frac{1-r_i}{1-r_o}x}^{x} p_i(x)p_o(y)dy \\ &\geq \int_0^{+\infty} p_i(x)dx \cdot (1 - \frac{1-r_i}{1-r_o})x \cdot min_{\frac{1-r_i}{1-r_o}x \leq y \leq x}(p_o(y)) \\ &\geq 0 \end{aligned} \tag{35}$$

Then we prove δP monotonically increases w.r.t $\frac{r_i - r_o}{1-r_o}$ from the perspective of function derivatives. Let $t = \frac{r_i - r_o}{1-r_o}$, we can derive from eq. 34:

$$\begin{aligned} \frac{d\{\delta P\}}{dt} &= \frac{d\{(\frac{1-r_i}{1-r_o})x\}}{dt} \cdot \frac{d\{\delta P\}}{d\{(\frac{1-r_i}{1-r_o})x\}} \\ &= \frac{d\{(\frac{1-r_i}{1-r_o})x\}}{dt} \cdot \{\int_0^{+\infty} p_i(x)dx \frac{d}{d\{(\frac{1-r_i}{1-r_o})x\}} \int_{\frac{1-r_i}{1-r_o}x}^{x} p_o(y)dy\} \\ &= (-x) \cdot \{-\int_0^{+\infty} p_i(x)p_o(\frac{1-r_i}{1-r_o}x)dx\} \\ &= x \cdot \int_0^{+\infty} p_i(x)p_o(\frac{1-r_i}{1-r_o}x)dx \\ &\geq 0. \end{aligned} \tag{36}$$

As $\frac{d\{\delta P\}}{dt} \geq 0$, we have δP monotonically increases w.r.t $\frac{r_i - r_o}{1-r_o}$.
Proof Ends.

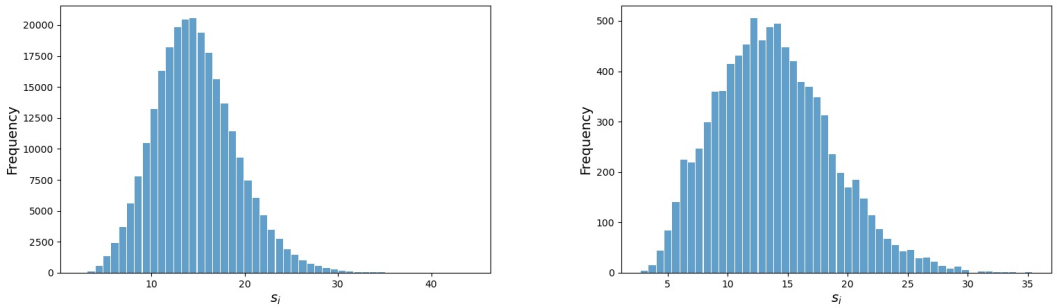

Figure 7. The distribution of maximal logits s_i on ImageNet200 train (left) and test (right) set. It can be observed that s_i is always non-negative.

Remark: i) The assumption $s_i \geq 0$ is easily satisfied. In Figure 7. We visualize the distribution of maximal logits s_i on the ImageNet200. It can be observed that $s_i \geq 0$ holds.
ii) In the proof, we simplified r_i and r_o to be constant values. The proved results can be easily extended to the case where r_i and r_o are not constant values. Specifically, we can prove: $Prob(s_i > s_o) \geq Prob\{(1-r_i^{min})s_i > (1-r_o^{max})s_o\} \geq Prob(s_i^{aug} > s_o^{aug})$, where $r_i^{min} = \min r_i$ and $r_o^{max} = \max r_o$.
First, we have $1 \geq r_i > r_o$. For some specific r_i/r_o, we could have $r_i^{min} > r_o^{max}$. Then, from the original proof, we have:

$$Prob(s_i > s_o) \geq Prob\{(1-r_i^{min})s_i > (1-r_o^{max})s_o\} \tag{37}$$

Then, we want to prove that $Prob\{(1-r_i^{min})s_i > (1-r_o^{max})s_o\} \geq Prob(s_i^{aug} > s_o^{aug})$. We have:

$$\begin{aligned}
Prob\{(1-r_i^{min})s_i > (1-r_o^{max})s_o\} &= Prob\{(1-r_i^{min})s_i > (1-r_o^{max})s_o | s_o < 0\} \cdot Prob(s_o < 0) \\
&+ Prob\{(1-r_i^{min})s_i > (1-r_o^{max})s_o | s_o \geq 0\} \cdot Prob(s_o \geq 0) \\
&= Prob(s_o < 0) + Prob\{(1-r_i^{min})s_i > (1-r_o^{max})s_o | s_o \geq 0\} \cdot Prob(s_o \geq 0) \\
&\geq Prob(s_o < 0) + Prob\{(1-r_i)s_i > (1-r_o^{max})s_o | s_o \geq 0\} \cdot Prob(s_o \geq 0) \\
&\geq Prob(s_o < 0) + Prob\{(1-r_i)s_i > (1-r_o)s_o | s_o \geq 0\} \cdot Prob(s_o \geq 0) \\
&= Prob\{(1-r_i)s_i > (1-r_o)s_o | s_o < 0\} \cdot Prob(s_o < 0) \\
&+ Prob\{(1-r_i)s_i > (1-r_o)s_o | s_o \geq 0\} \cdot Prob(s_o \geq 0) \\
&= Prob\{(1-r_i)s_i > (1-r_o)s_o\} \\
&= Prob(s_i^{aug} > s_o^{aug})
\end{aligned} \tag{38}$$

Combining eq. 37 and 38, we have $Prob(s_i > s_o) \geq Prob(s_i^{aug} > s_o^{aug})$ for inconstant r_i and r_o satisfying $1 \geq r_i > r_o$.