Mitigating Neural Network Overconfidence with Logit Normalization

Hongxin Wei [1] **Renchunzi Xie** [1] **Hao Cheng** [1 2] **Lei Feng** [3] **Bo An** [1] **Yixuan Li** [4]

Abstract

Detecting out-of-distribution inputs is critical for the safe deployment of machine learning models in the real world. However, neural networks are known to suffer from the overconfidence issue, where they produce abnormally high confidence for both in- and out-of-distribution inputs. In this work, we show that this issue can be mitigated through Logit Normalization (**LogitNorm**)—a simple fix to the cross-entropy loss—by enforcing a constant vector norm on the logits in training. Our method is motivated by the analysis that the norm of the logit keeps increasing during training, leading to overconfident output. Our key idea behind LogitNorm is thus to decouple the influence of output's norm during network optimization. Trained with LogitNorm, neural networks produce highly distinguishable confidence scores between in- and out-of-distribution data. Extensive experiments demonstrate the superiority of LogitNorm, reducing the average FPR95 by up to 42.30% on common benchmarks.

1. Introduction

Modern neural networks deployed in the open world often struggle with out-of-distribution (OOD) inputs—samples from a different distribution that the network has not been exposed to during training, and therefore should not be predicted with high confidence at test time. A reliable classifier should not only accurately classify known in-distribution (ID) samples, but also identify as "unknown" any OOD input. This gives rise to the importance of OOD detection, which determines whether an input is ID or OOD and allows the model to take precautions in deployment.

A naive solution uses the maximum softmax probability (MSP)—also known as the softmax confidence score—for OOD detection (Hendrycks & Gimpel, 2016). The operating hypothesis is that OOD data should trigger relatively lower softmax confidence than that of ID data. Whilst intuitive, the reality shows a non-trivial dilemma. In particular, deep neural networks can easily produce overconfident predictions, *i.e.*, abnormally high softmax confidences, even when the inputs are far away from the training data (Nguyen et al., 2015). This has cast significant doubt on using softmax confidence for OOD detection. Indeed, many prior works turned to define alternative OOD scoring functions (Liang et al., 2018; Lee et al., 2018; Liu et al., 2020; Sastry & Oore, 2020; Sun et al., 2021; Huang et al., 2021; Sun et al., 2022). Yet to date, the community still has a limited understanding of the fundamental cause and mitigation of the overconfidence issue.

In this work, we show that the overconfidence issue can be mitigated through a simple fix to the cross-entropy loss—the most commonly used training objective for classification—by enforcing a constant norm on the logit vector (*i.e.*, pre-softmax output). Our method, *Logit Normalization* (dubbed **LogitNorm**), is motivated by our analysis on the norm of the neural network's logit vectors. We find that even when most training examples are classified to their correct labels, the softmax cross-entropy loss can continue to increase the magnitude of the logit vectors. The growing magnitude during training thus leads to the overconfidence issue, despite having no improvement on the classification accuracy.

To mitigate the issue, our key idea behind LogitNorm is to decouple the influence of output's norm from the training objective and its optimization. This can be achieved by normalizing the logit vector to have a constant norm during training. In effect, our LogitNorm loss encourages the direction of the logit output to be consistent with the corresponding one-hot label, without exacerbating the magnitude of the output. Trained with normalized outputs, the network tends to give conservative predictions and results in strong separability of softmax confidence scores between ID and OOD inputs (see Figure 4).

Extensive experiments demonstrate the superiority of LogitNorm over existing methods for OOD detection. First, our method drastically improves the OOD detection perfor-

[1] Nanyang Technological University, Singapore [2] Nanjing University, Nanjing, Jiangsu, China [3] Chongqing University, Chongqing, China [4] University of Wisconsin-Madison, Wisconsin, United States. Correspondence to: Renchunzi Xie <XIER0002@e.ntu.edu.sg>.

Proceedings of the 39^{th} *International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

mance using the softmax confidence score. For example, using CIFAR-10 dataset as ID and SVHN as OOD data, our approach reduces the FPR95 from 50.33% to 8.03%—a **42.30**% of improvement over the baseline (Hendrycks & Gimpel, 2016). Averaged over a diverse collection of OOD datasets, our method reduces the FPR95 by **33.87**% compared to using the softmax score with cross-entropy loss. Beyond MSP, we show that our method not only outperforms, but also boosts more advanced post-hoc OOD scoring functions, such as ODIN (Liang et al., 2018), energy score (Liu et al., 2020), and GradNorm score (Huang et al., 2021). In addition to the OOD detection task, our method improves the calibration performance on the ID data itself by way of post-hoc temperature scaling.

Overall, *using LogitNorm loss achieves strong performance on OOD detection and calibration tasks while maintaining the classification accuracy on ID data.* Our method can be easily adopted in practice. It is straightforward to implement with existing deep learning frameworks, and does not require sophisticated changes to the loss or training scheme. Code and data are publicly available at `https://github.com/hongxin001/logitnorm_ood`.

We summarize our contributions as follows:

1. We introduce LogitNorm – a simple and effective alternative to the cross-entropy loss, which decouples the influence of logits' norm from the training procedure. We show that LogitNorm can effectively generalize to different network architectures and boost different post-hoc OOD detection methods.
2. We conduct extensive evaluations to show that LogitNorm can improve both OOD detection and confidence calibration while maintaining the classification accuracy on ID data. Compared with the cross-entropy loss, LogitNorm achieves an FPR95 reduction of 33.87% on common benchmarks with the softmax confidence score.
3. We perform ablation studies that lead to improved understandings of our method. In particular, we contrast with alternative methods (*e.g.*, GODIN (Hsu et al., 2020), Logit Penalty) and demonstrate the advantages of LogitNorm. We hope that our insights inspire future research to further explore loss function design for OOD detection.

2. Background

2.1. Preliminaries: Out-of-distribution Detection

Setup. We consider a supervised multi-class classification problem. We denote by $\mathcal{X}$ the input space and $\mathcal{Y} = \{1, \dots, k\}$ the label space with k classes. The training dataset $\mathcal{D}_{\text{train}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ consists of N data points, sampled *i.i.d.* from a joint data distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$. We use $\mathcal{P}_{\text{in}}$ to denote the marginal distribution over $\mathcal{X}$, which represents the in-distribution (ID). Given the training dataset, we learn a classifier $f : \mathcal{X} \to \mathbb{R}^k$ with trainable parameter $\theta \in \mathbb{R}^p$, which maps an input to the output space. An ideal classifier can be obtained by minimizing the following expected risk:

$$\mathcal{R}_{\mathcal{L}}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{X}\mathcal{Y}}} \left[\mathcal{L}\left(f(\boldsymbol{x};\theta), y\right)\right],$$

where $\mathcal{L}$ is the commonly used cross-entropy loss with the softmax activation function:

$$\mathcal{L}_{\text{CE}}(f(\boldsymbol{x};\theta), y) = -\log p(y|\boldsymbol{x}) = -\log \frac{e^{f_y(\boldsymbol{x};\theta)}}{\sum_{i=1}^k e^{f_i(\boldsymbol{x};\theta)}}.$$

Here, $f_y(\boldsymbol{x};\theta)$ denotes the y-th element of $f(\boldsymbol{x};\theta)$ corresponding to the ground-truth label y, and $p(y|\boldsymbol{x})$ is the corresponding softmax probability.

Problem statement. During the deployment time, it is ideal that the test data are drawn from the same distribution $\mathcal{P}_{\text{in}}$ as the training data. However, in reality, inputs from unknown distributions can arise, whose label set may have no intersection with $\mathcal{Y}$. Such inputs are termed out-of-distribution (OOD) data and should not be predicted by the model.

The OOD detection task can be formulated as a binary classification problem: determining whether an input $\boldsymbol{x} \in \mathcal{X}$ is from $\mathcal{P}_{\text{in}}$ (ID) or not (OOD). OOD detection can be performed by a level-set estimation:

$$g(\boldsymbol{x}) = \begin{cases} \text{in} & \text{if } S(\boldsymbol{x}) \geq \gamma \\ \text{out} & \text{if } S(\boldsymbol{x}) < \gamma \end{cases}, \qquad (1)$$

where $S(\boldsymbol{x})$ denotes a scoring function and γ is commonly chosen so that a high fraction (e.g., 95%) of ID data is correctly classified. By convention, samples with higher scores $S(\boldsymbol{x})$ are classified as ID and vice versa. In Section 4.2, we will consider a variety of popular OOD scoring functions including MSP (Hendrycks & Gimpel, 2016), ODIN (Liang et al., 2018), energy score (Liu et al., 2020) and GradNorm (Huang et al., 2021).

3. Method: Logit Normalization

3.1. Motivation

In the following, we investigate why neural networks trained with the common softmax cross-entropy loss tend to give overconfident predictions. Our analysis suggests that the large magnitude of neural network output can be a culprit.

For notation shorthand, we use $\boldsymbol{f}$ to denote the network output $f(\boldsymbol{x};\theta)$ for an input $\boldsymbol{x}$. $f(\boldsymbol{x};\theta)$ is also known as the logit, or pre-softmax output. Without loss of generality, the logit vector $\boldsymbol{f}$ can be decomposed into two components:

$$\boldsymbol{f} = \|\boldsymbol{f}\| \cdot \hat{\boldsymbol{f}}, \qquad (2)$$

where $\|\boldsymbol{f}\| = \sqrt{\boldsymbol{f}_1^2 + \boldsymbol{f}_2^2 + \cdots + \boldsymbol{f}_k^2}$ is the Euclidean norm of the logit vector $\|\boldsymbol{f}\|$, and $\hat{\boldsymbol{f}}$ is the unit vector in the same direction as $\boldsymbol{f}$. In other word, $\|\boldsymbol{f}\|$ and $\hat{\boldsymbol{f}}$ represent the *magnitude* and the *direction* of the logit vector $\boldsymbol{f}$, respectively.

During the test stage, the model makes class predictions by $c = \arg\max_i(f_i)$. We have the following propositions.

Proposition 3.1. *For any give constant value $s > 1$, if* $\arg\max_i(f_i) = c$*, then* $\arg\max_i(sf_i) = c$ *always holds.*

Given the above proposition, we find that scaling the magnitude $\|\boldsymbol{f}\|$ of the logit does not change the predicted class c. In the following, we further explore how it impacts the softmax confidence score.

Proposition 3.2. *For the softmax cross-entropy loss, let σ be the softmax activation function. For any given scalar $s > 1$, if $c = \arg\max_i(f_i)$, then $\sigma_c(s\boldsymbol{f}) \geq \sigma_c(\boldsymbol{f})$ holds.*

The proofs of the above propositions are presented in Appendix A and B. From Proposition 3.2, we find that increasing the magnitude $\|\boldsymbol{f}\|$ will cause a higher value for the softmax confidence score but leave the final prediction unchanged.

To analyze the impact on training objective, we provide the following formulation according to Eq. (2):

$$\mathcal{L}_{\text{CE}}(f(\boldsymbol{x};\theta), y) = -\log p(y|\boldsymbol{x}) = -\log \frac{e^{\|\boldsymbol{f}\|\cdot\hat{f}_y}}{\sum_{i=1}^{k} e^{\|\boldsymbol{f}\|\cdot\hat{f}_i}}.$$

We can find that the training loss depends on the magnitude $\|\boldsymbol{f}\|$ and the direction $\hat{\boldsymbol{f}}$. By keeping the direction unchanged, we analyze the influence of the magnitude $\|\boldsymbol{f}\|$ on the training loss. When $y = \arg\max_i(f_i)$, we can see that increasing $\|\boldsymbol{f}\|$ would increase $p(y|\boldsymbol{x})$. It implies that, for those training examples that are already classified correctly, the optimization on the training loss would further increase the magnitude $\|\boldsymbol{f}\|$ of the network output to produce a higher softmax confidence score, thus obtaining a smaller loss.

To provide a straightforward view, we show in Figure 1 the dynamics of logit norm during training. Indeed, softmax cross-entropy loss encourages the model to produce logits with increasingly larger norms for both ID and OOD examples. The large norms directly translate into overconfident softmax scores, leading to difficulty in separating ID vs. OOD data. We proceed by introducing our method, targeting this problem.

3.2. Method

In our previous analysis, we show that the softmax cross-entropy loss encourages the network to produce logits with larger magnitudes, leading to the overconfidence issue that makes it difficult to distinguish ID and OOD examples. To alleviate this issue, our key idea is to decouple the influence of logits' magnitude from network optimization. In other words, our goal is to keep the L_2 vector norm of logits a constant during the training. Formally, the objective can be formulated as:

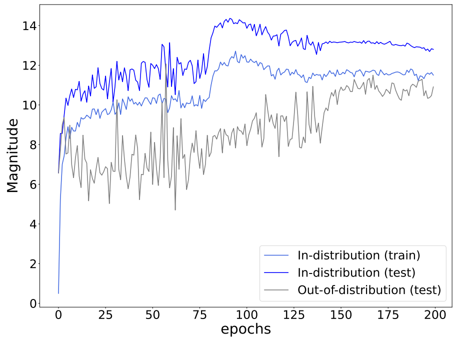


Figure 1. The mean magnitudes of logits under different training epochs. Model is trained on CIFAR-10 with WRN-40-2 (Zagoruyko & Komodakis, 2016). OOD examples are from SVHN dataset.

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}_{\mathcal{P}_{\mathcal{X}\mathcal{Y}}}\left[\mathcal{L}_{\text{CE}}\left(f(\boldsymbol{x};\theta), y\right)\right] \\ \text{subject to} \quad & \|f(\boldsymbol{x};\theta)\|_2 = \alpha. \end{aligned}$$

Performing constrained optimization in the context of modern neural networks is non-trivial. As we will show later in Section 5, simply adding the constraint via Lagrange multiplier (Forst & Hoffmann, 2010) does not work well. To circumvent the issue, we convert the objective into an alternative loss function that can be end-to-end trainable, which strictly enforces a constant vector norm.

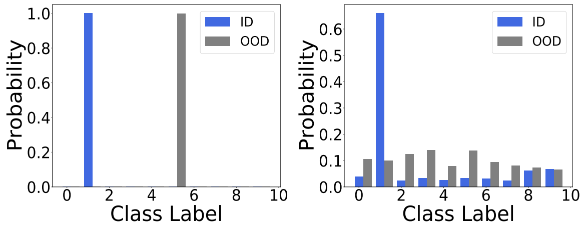
Logit Normalization. We employ *Logit Normalization* (dubbed LogitNorm), which encourages the direction of the logit to be consistent with the corresponding one-hot label, without optimizing the magnitude of the logit. In particular, the logit vector is normalized to be a unit vector with a constant magnitude. The softmax cross-entropy loss is then applied on the normalized logit vector instead of the original output. Formally, the objective function of LogitNorm is given by:

$$\mathcal{R}_{\mathcal{L}}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{X}\mathcal{Y}}}\left[\mathcal{L}_{\text{CE}}\left(\hat{f}(\boldsymbol{x};\theta), y\right)\right], \qquad (3)$$

where $\hat{f}(\boldsymbol{x};\theta) = f(\boldsymbol{x};\theta)/\|f(\boldsymbol{x};\theta)\|$ is the normalized logit vector. In practice, a small positive value (e.g., 10^{-7}) is added to the denominator to ensure numerical stability. Equivalently, the new loss function can be defined as:

$$\mathcal{L}_{\text{logit\_norm}}(f(\boldsymbol{x};\theta), y) = -\log \frac{e^{f_y/(\tau\|\boldsymbol{f}\|)}}{\sum_{i=1}^{k} e^{f_i/(\tau\|\boldsymbol{f}\|)}}, \qquad (4)$$

where the temperature parameter τ modulates the magnitude of the logits. Interestingly, our loss function can be viewed



(a) Cross-entropy Loss (b) LogitNorm Loss (ours)

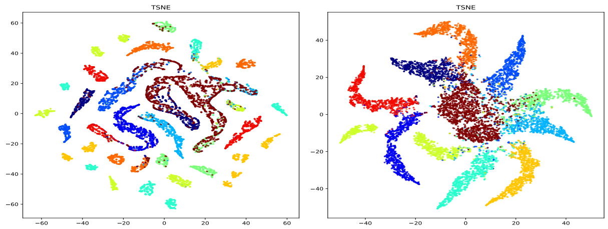
Figure 2. The softmax outputs of two examples on a CIFAR-10 pre-trained WRN-40-2 (Zagoruyko & Komodakis, 2016) with (a) cross-entropy loss and (b) logit normalization loss. For the cross-entropy loss, the softmax confidence scores are 1.0 and 1.0 for the ID and OOD examples. In contrast, the softmax confidence scores of the network trained with LogitNorm loss are 0.66 and 0.14 for the ID and OOD examples. While using cross-entropy loss produces an extremely confident prediction for the OOD example, our method produces almost uniform softmax probabilities (near 0.1), which benefits OOD detection.

as having an *input-dependent* temperature, $\tau\|f(\boldsymbol{x};\theta)\|$, which depends on the input $\boldsymbol{x}$.

By way of logit normalization, the magnitudes of the output vectors are strictly constant (*i.e.*, $1/\tau$). Minimizing the loss in Eq. (4) can only be achieved by adjusting the direction of the logit output f. The resulting model tends to give conservative predictions, especially for inputs that are far away from $\mathcal{P}_{\text{in}}$. We illustrate with an example in Figure 2, where training with logit normalization leads to softmax outputs that are more distinguishable between in- and out-of-distribution samples (right), as opposed to using cross-entropy loss (left). In Figure 3, we present a t-SNE visualization of the softmax outputs using Cross-entropy vs. LogitNorm Loss, where LogitNorm leads to more meaningful information to differentiate ID and OOD samples in the softmax output space. Below we further provide a lower bound of this new loss function in Eq. (4).

Proposition 3.3 (Lower Bound of Loss)**.** *For any input $\boldsymbol{x}$ and any positive number $\tau \in \mathbb{R}^+$, the per-sample loss defined in Eq. (4) has a lower bound: $\mathcal{L}_{logit\_norm} \geq \log\left(1+(k-1)e^{-2/\tau}\right)$, where k is the number of classes.*

The proof of Proposition 3.3 is provided in Appendix C. From Proposition 3.3, we find that the LogitNorm loss has a lower bound that depends on τ and number of classes k. In particular, it implies that the lower bound of the loss value increases with the value of τ. For example, when $k = 10$ and $\tau = 1$, the norm of logits would be linearly scaled to 1 and the lower bound is about 0.7966. The high lower bound can cause optimization difficulty. For this reason, we found it is desirable to have a relatively small $\tau < 1$. We will analyze the effect of τ in detail in Section 5.



(a) Cross-entropy Loss (b) LogitNorm Loss (ours)

Figure 3. t-SNE visualization (Van der Maaten & Hinton, 2008) of the softmax outputs from WRN-40-2 (Zagoruyko & Komodakis, 2016) trained on CIFAR-10 with (a) cross-entropy loss and (b) LogitNorm loss. All colors except for brown indicate 10 different ID classes. Points in *brown* denote out-of-distribution examples from SVHN. Trained with logit normalization, the softmax outputs provide more meaningful information to differentiate in- and out-distribution samples.

4. Experiments

In this section, we verify the effectiveness of LogitNorm loss in OOD detection with several benchmark datasets.

4.1. Experimental Setup

In-distribution datasets. In this work, we use the CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) datasets as in-distribution datasets, which are common benchmarks for OOD detection. Specifically, we use the standard split with 50,000 training images and 10,000 test images. All the images are of size 32×32.

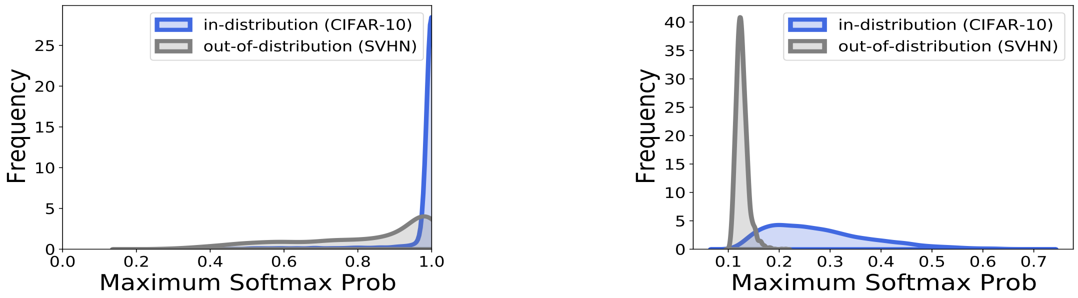
Out-of-distribution datasets. For the OOD detection evaluation, we use six common benchmarks as OOD test datasets $\mathcal{D}_{\text{out}}^{\text{test}}$: `Textures` (Cimpoi et al., 2014), `SVHN` (Netzer et al., 2011), `Places365` (Zhou et al., 2017), `LSUN-Crop` (Yu et al., 2015), `LSUN-Resize` (Yu et al., 2015), and `iSUN` (Xu et al., 2015). For all test datasets, the images are of size 32×32. The detail information of the six datasets is presented in Appendix D.

Evaluation metrics. We evaluate the performance of OOD detection by measuring the following metrics: (1) the false positive rate (FPR95) of OOD examples when the true positive rate of in-distribution examples is 95%; (2) the area under the receiver operating characteristic curve (AUROC); and (3) the area under the precision-call curve (AUPR).

Training details. For main results, we perform training with WRN-40-2 (Zagoruyko & Komodakis, 2016) on CIFAR-10/100. The network is trained for 200 epochs using SGD with a momentum of 0.9, a weight decay of 0.0005, a dropout rate of 0.3, and a batch size of 128. We set the initial learning rate as 0.1 and reduce it by a factor of 10 at 80 and 140 epochs. The hyperparameter τ is selected from the range $\{0.001, 0.005, 0.01, \ldots, 0.05\}$. We set 0.04 for CIFAR-10 by default. For hyperparameter tuning, we use

Table 1. OOD detection performance comparison using softmax cross-entropy loss and LogitNorm loss. We use WRN-40-2 (Zagoruyko & Komodakis, 2016) to train on the in-distribution datasets and use softmax confidence score as the scoring function. All values are percentages. ↑ indicates larger values are better, and ↓ indicates smaller values are better. **Bold** numbers are superior results.

ID datasets	CIFAR-10			CIFAR-100		
OOD datasets	FPR95 ↓	AUROC ↑	AUPR ↑	FPR95 ↓	AUROC ↑	AUPR ↑
	Cross-entropy loss / **LogitNorm loss (ours)**					
Texture	64.13 / **28.64**	86.29 / **94.28**	96.28 / **98.63**	84.11 / **70.67**	74.05 / **78.65**	93.45 / **93.66**
SVHN	50.33 / **8.03**	93.48 / **98.47**	98.70 / **99.68**	79.09 / **45.98**	78.62 / **92.48**	94.99 / **98.45**
LSUN-C	33.34 / **2.37**	95.29 / **99.42**	99.04 / **99.88**	67.94 / **13.93**	83.60 / **97.56**	96.32 / **99.48**
LSUN-R	42.52 / **10.93**	93.74 / **97.87**	98.66 / **99.59**	82.21 / **68.68**	69.45 / **84.77**	91.45 / **96.52**
iSUN	46.56 / **12.28**	93.13 / **97.73**	98.55 / **99.56**	84.50 / **71.47**	69.29 / **83.79**	91.44 / **96.24**
Places365	60.23 / **31.64**	87.36 / **93.66**	96.62 / **98.49**	81.09 / **80.20**	75.61 / **77.14**	93.74 / **94.16**
average	49.52 / **15.65**	91.55 / **96.91**	97.98 / **99.31**	79.82 / **58.49**	75.10 / **85.73**	93.57 / **96.42**



(a) Cross-entropy Loss

(b) LogitNorm Loss (Ours)

Figure 4. Distribution of softmax confidence score from WRN-40-2 (Zagoruyko & Komodakis, 2016) trained on CIFAR-10 with (a) cross-entropy loss and (b) LogitNorm loss.

Gaussian noises as the validation set. All experiments are repeated five times with different seeds, and we report the average performance. We conduct all the experiments on NVIDIA GeForce RTX 3090 and implement all methods with default parameters using PyTorch (Paszke et al., 2019).

4.2. Results

How does logit normalization influence OOD detection performance? In Table 1, we compare the OOD detection performance on models trained with cross-entropy loss and LogitNorm loss respectively. To isolate the effect of the loss function in training, we keep the test-time OOD scoring function to be the same, *i.e.*, softmax confidence score:

$$S(\boldsymbol{x}) = \max_i \frac{e^{f_i(\boldsymbol{x};\theta)}}{\sum_{j=1}^{k} e^{f_j(\boldsymbol{x};\theta)}}.$$

A salient observation is that our method drastically improves the OOD detection performance by employing LogitNorm loss. For example, on the CIFAR-10 model, when evaluated against SVHN as OOD data, our method reduces the FPR95 from 50.33% to 8.03%—a **42.3**% of direct improvement. Averaged across six test datasets, our approach reduces the FPR95 by **33.87**% compared with using MSP on the model trained with cross-entropy loss. On CIFAR-100, our method also improves the performance by a meaningful margin.

To further illustrate the difference between the two losses on OOD detection, we visualize and compare the distribution of softmax confidence score for ID and OOD data, derived from networks trained with cross-entropy vs. LogitNorm losses. With cross-entropy loss, the softmax scores for both ID and OOD data concentrate on high values, as shown in Figure 4a. In contrast, the network trained with LogitNorm loss produces highly distinguishable scores between ID and OOD data. From Figure 4b, we observe that the softmax confidence score for most OOD examples is around 0.1, which indicates that the softmax outputs are close to a uniform distribution. Overall the experiments show that training with LogitNorm loss makes the softmax scores more distinguishable between in- and out-of-distributions and consequently enables more effective OOD detection.

Can the logit normalization improve existing scoring functions? In Table 2, we show that the LogitNorm loss not only outperforms, but also boosts competitive OOD scoring functions. Note that all the OOD scoring functions consid-

Table 2. OOD detection performance comparison with various scoring functions using softmax cross-entropy loss and logit normalization loss. We use WRN-40-2 to train on the in-distribution datasets. All values are percentages. ↑ indicates larger values are better, and ↓ indicates smaller values are better. **Bold** numbers are superior results.

ID datasets	CIFAR-10			CIFAR-100		
Score	FPR95 ↓	AUROC ↑	AUPR ↑	FPR95 ↓	AUROC ↑	AUPR ↑
	Cross-entropy loss / **LogitNorm loss (ours)**					
Softmax	49.52 / **15.65**	91.55 / **96.91**	97.98 / **99.31**	79.82 / **58.35**	75.10 / **85.75**	93.57 / **96.44**
ODIN	40.32 / **12.95**	87.21 / **97.37**	96.24 / **99.40**	70.71 / **58.13**	81.38 / **85.65**	95.37 / **96.36**
Energy	26.82 / **19.14**	93.07 / **96.09**	98.02 / **99.14**	70.87 / **65.46**	81.45 / **84.84**	95.38 / **96.27**
GradNorm	58.98 / **17.78**	72.29 / **96.34**	90.75 / **99.14**	87.01 / **61.89**	52.84 / **81.41**	84.12 / **94.85**

Table 3. OOD detection performance comparison trained on CIFAR-10 with different network architectures: WRN-40-2 (Zagoruyko & Komodakis, 2016), ResNet-34 (He et al., 2016), DenseNet-BC (Huang et al., 2017). All values are percentages. ↑ indicates larger values are better, and ↓ indicates smaller values are better. **Bold** numbers are superior results.

Architecture	ID Accuracy ↑	FPR95 ↓	AUROC ↑	AUPR ↑
	Cross-entropy loss / **LogitNorm loss (ours)**			
WRN-40-2	94.75 / 94.69	49.52 / **15.65**	91.55 / **96.91**	97.98 / **99.31**
ResNet-34	95.01 / 95.14	47.74 / **15.82**	91.15 / **97.01**	97.72 / **99.32**
DenseNet	94.55 / 94.37	50.41 / **18.57**	91.48 / **96.16**	98.04 / **99.10**

ered are originally developed based on models trained with cross-entropy loss, hence are natural choices of comparison. In particular, we consider: 1) **MSP** (Hendrycks & Gimpel, 2016) uses the softmax confidence score to detect OOD samples. 2) **ODIN** (Liang et al., 2018) employs temperature scaling and input perturbation to improve OOD detection. Following the original setting, we use the temperature parameter $T = 1000$ and $\epsilon = 0.0014$. 3) **Energy score** (Liu et al., 2020) utilizes the information in logits for OOD detection, which is the negative log of the denominator in softmax function: $E(\mathbf{x}; f) = -T \cdot \log \sum_{i=1}^{k} e^{f_i(\boldsymbol{x})/T}$. With LogitNorm loss, we set $T = 0.1$ for CIFAR-10 and $T = 0.01$ for CIFAR-100. 4) **GradNorm** (Huang et al., 2021) detects OOD inputs by utilizing information extracted from the gradient space.

Our results in Table 2 suggest that logit normalization can benefit a wide range of downstream OOD scoring functions. Due to space constraints, we report the average performance across six test OOD datasets. The OOD detection performance on each OOD test dataset is provided in Appendix F. For example, we observe that the FPR95 of the ODIN method is reduced from 40.32% to 12.95% when employing logit normalization, establishing strong performance. In addition, we find that logit normalization enables the energy score and GradNorm score to achieve decent OOD detection performance as well.

Table 4. Comparison results of ECE (%) with $M = 15$, using WRN-40-2 trained on CIFAR-10. For temperature scaling (TS), T is tuned on a hold-out validation set by optimization methods.

Dataset		Cross Entropy	LogitNorm (ours)
CIFAR-10	w/o TS	3.20	66.95
	w/ TS	0.66	**0.41**
CIFAR-100	w/o TS	11.69	70.45
	w/ TS	2.18	**1.67**

Logit normalization is effective on different architectures. In Table 3, we show that LogitNorm is effective on a diverse range of model architectures. The results are based on softmax confidence score as test-time OOD score. In particular, our method consistently improves the performance using WRN-40-2 (Zagoruyko & Komodakis, 2016), ResNet (He et al., 2016) and DenseNet (Huang et al., 2017) architectures. For example, on DenseNet, using LogitNorm loss reduces the average FPR95 from 50.41% to 18.57%.

Logit normalization maintains classification accuracy. We also verify whether LogitNorm loss affects the classification accuracy. Our results in Table 3 show that LogitNorm can improve the OOD detection performance, and at the same time, achieve similar accuracy as the cross-entropy loss. For example, when trained on ResNet-34 with CIFAR-10 as the ID dataset, using LogitNorm loss achieves a test accuracy of 95.14% on CIFAR-10, on par with the test accuracy 95.01% using cross-entropy loss. On CIFAR-100, LogitNorm loss and cross-entropy loss also achieves comparable test accuracies (75.12% vs. 75.23%). Overall LogitNorm loss maintains comparable classification accuracy on the ID data while leading to significant improvement in OOD detection performance.

Logit normalization enables better calibration. While the OOD detection task concentrates on the separability between ID and OOD data, the calibration task focuses solely on the ID data—softmax confidence score should represent the true probability of correctness (Guo et al., 2017). In practice, Expected Calibration Error (ECE) (Naeini et al., 2015) is commonly used to measure the calibration performance from finite samples. In Figure 4, we observe that LogitNorm loss leads to a smoother distribution of softmax

Table 5. OOD detection performance comparison with different loss functions. We use WRN-40-2 trained on CIFAR-10. **Bold** numbers are superior results. We report the average norm value of logits as L_2 Norm, where we use SVHN as OOD test dataset.

Loss	FPR95 ↓	AUROC ↑	AUPR ↑	L_2 Norm (ID / OOD)
CrossEntropy	49.52	91.55	97.98	12.80 / 10.91
LogitPenalty	57.62	73.27	87.62	1.90 / 1.74
LogitNorm (ours)	**15.65**	**96.91**	**99.31**	1.48 / 0.49

Table 6. Comparison between GODIN and LogitNorm. We use WRN-40-2 (Zagoruyko & Komodakis, 2016) trained on CIFAR-10. For a fair comparison, we use the ODIN score for LogitNorm loss. **Bold** numbers are superior results.

Loss	FPR95 ↓	AUROC ↑	AUPR ↑
GODIN	25.24	95.07	98.93
LogitNorm (ours)	**15.65**	**96.91**	**99.31**

confidence score for ID examples, in contrast to the spiky distribution induced by cross-entropy loss (*i.e.*, values concentrate around 1). It implies that the model trained with LogitNorm loss preserves distinguishable information for different ID samples, indicating its potential in improving model calibration. Indeed, our results in Table 4 show that the model trained with LogitNorm achieves better calibration performance by way of post-hoc temperature scaling.

5. Discussion

Logit normalization vs. Logit penalty. While our logit normalization has demonstrated strong promise, a question arises: *can a similar effect be achieved by imposing a penalty on the L_2 norm of the logits*? In this ablation, we show that explicitly constraining logit norm via a Lagrangian multiplier (Forst & Hoffmann, 2010) does not work well. Specifically, we consider the alternative loss:

$$\mathcal{L}_{\text{logit\_penalty}}(f(\boldsymbol{x};\theta),y) = \mathcal{L}_{\text{CE}}(f(\boldsymbol{x};\theta),y) + \lambda \|f(\boldsymbol{x};\theta)\|_2.$$

where λ denotes the Lagrangian multiplier that controls the trade-off between the cross-entropy loss and the regularization term.

Our results in Table 5 show that both logit penalty and logit normalization lead to logits with small L_2 norms, compared with using cross-entropy loss. However, unlike LogitNorm, the logit penalty method produces a large L_2 norm for OOD data as well, resulting in the inferior performance of OOD detection. In practice, we notice that the network trained with the logit penalty can suffer from optimization difficulty and sometimes fail to converge if λ is too large (which is needed to regularize the logit norm effectively). Overall, we show that simply constraining the logit norm during training cannot help the OOD detection task while our LogitNorm loss significantly improves the performance.

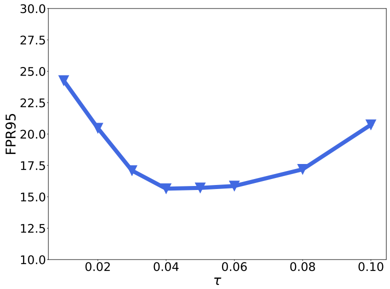


Figure 5. Effect of τ in LogitNorm with CIFAR-10.

Relations to temperature scaling. As introduced in Section 3, the logit normalization can be viewed as an *input-dependent* temperature on the logits. Related to our work, previous work ODIN (Liang et al., 2018) proposes a variant of softmax score by employing temperature scaling with a *constant* T in the testing phase:

$$S(\boldsymbol{x}) = \max_i \frac{e^{f_i(\boldsymbol{x};\theta)/T}}{\sum_{j=1}^{k} e^{f_j(\boldsymbol{x};\theta)/T}},$$

where the temperature is the same for all inputs. In contrast, our method bares two key differences: (1) the effective temperature in LogitNorm is input-dependent rather than a global constant, and (2) the temperature in LogitNorm can be enforced during the training stage. Our method is compatible with test-time temperature scaling in OOD detection and can improve calibration performance with temperature scaling, as we show in Section 4.2.

In Figure 5, we further ablate how the parameter τ in our method (*cf.* Eq. 4) affects the OOD detection performance. The analysis is based on CIFAR-10, with FPR95 averaged across six test datasets. Our results echo the analysis in Proposition 3.3, where a large τ would lead to a large lower bound on the loss, which is less desirable from the optimization perspective.

Relations to other normalization methods. In the literature, normalization method is applied for OOD detection in the form of cosine similarity (Techapanurak & Okatani, 2019; Hsu et al., 2020). In particular, Cosine loss (Techapanurak & Okatani, 2019) and Generalized ODIN (GODIN) (Hsu et al., 2020) decompose the logit $f_i(\mathbf{x};\theta)$ for class i as: $f_i(\mathbf{x};\theta) = \frac{h_i(\mathbf{x};\theta)}{g(\mathbf{x};\theta)}$, where

$$\begin{cases} h_i(\boldsymbol{x}) = \cos(\boldsymbol{w}_i, \phi^p(\boldsymbol{x})) = \frac{\boldsymbol{w}_i^T \phi^p(\boldsymbol{x})}{\|\boldsymbol{w}_i\| \|\phi^p(\boldsymbol{x})\|}, \\ g(\mathbf{x};\theta) = \exp\left\{ \text{BN}\left(\boldsymbol{W}^\top \phi^p(\boldsymbol{x}) + b \right) \right\}. \end{cases}$$

In the testing phase, the maximum cosine similarity is used as the scoring function. Contrastingly, LogitNorm has two key differences: (1) the cosine similarity in $h_i(\boldsymbol{x})$ applies L_2 normalization on the last-layer weight $\boldsymbol{w}$ and the learned feature $\phi^p(\boldsymbol{x})$, while our LogitNorm normalizes the network output $f(\boldsymbol{x};\theta)$; (2) our LogitNorm can boost the performance of common OOD scoring functions, while GODIN and Cosine loss detect OOD examples with their specific scoring functions. In Table 6, we show our LogitNorm with the MSP score achieves better performance than GODIN in OOD detection. More discussion on future work is provided in Appendix. E.

6. Related Work

OOD detection. OOD detection is an increasingly important topic for deploying machine learning models in the open world and has attracted a surge of interest in two directions.

1) Some methods aim to design scoring functions for OOD detection, such as OpenMax score (Bendale & Boult, 2016), maximum softmax probability (Hendrycks & Gimpel, 2016), ODIN score (Liang et al., 2018; Hsu et al., 2020), Mahalanobis distance-based score (Lee et al., 2018), Energy-based score (Liu et al., 2020; Wang et al., 2021b; Morteza & Li, 2022), ReAct (Sun et al., 2021), GradNorm score (Huang et al., 2021), and non-parametric KNN-based score (Sun et al., 2022; Zhu et al., 2022). In this work, we first show that logit normalization can drastically mitigate the overconfidence issue for OOD data, thereby boosting the performance of existing scoring functions in OOD detection.

2) Some works address the out-of-distribution detection problem by training-time regularization (Lee et al., 2017; Bevandić et al., 2018; Hendrycks et al., 2019; Geifman & El-Yaniv, 2019; Malinin & Gales, 2018; Mohseni et al., 2020; Jeong & Kim, 2020; Liu et al., 2020; Chen et al., 2021; Wei et al., 2021; 2022; Ming et al., 2022a). For example, models are encouraged to give predictions with uniform distribution (Lee et al., 2017; Hendrycks et al., 2019) or higher energies (Liu et al., 2020; Du et al., 2022b; Ming et al., 2022a; Du et al., 2022a; Katz-Samuels et al., 2022) for outliers. The energy-based regularization has a direct theoretical interpretation as shaping the log-likelihood, hence naturally suits OOD detection. Contrastive learning methods are also employed for the OOD detection task (Tack et al., 2020; Sehwag et al., 2021; Ming et al., 2022b), which can be computationally more expensive to train than ours. In this work, we focus on exploring classification-based loss functions for OOD detection, which only requires in-distribution data in training. LogitNorm is easy to implement and use, and maintains the same training scheme as standard cross-entropy loss.

Normalization in deep learning. In the literature, normalization has been widely used in metric learning (Sohn, 2016; Wu et al., 2018; van den Oord et al., 2018), face recognition (Ranjan et al., 2017; Liu et al., 2017; Wang et al., 2017; 2018; Deng et al., 2019; Zhang et al., 2019), and self-supervised learning (Chen et al., 2020). L_2-constrained softmax (Ranjan et al., 2017) applies the L_2 normalization on features and SphereFace (Liu et al., 2017) normalizes the weights of the last inner-product later only. Cosine loss (Wang et al., 2017; 2018) normalizes both the features and weights to achieve better performance for face verification. LayerNorm (Xu et al., 2019) normalizes the distributions of intermediate layers for better generalization accuracy. In self-supervised learning, SimCLR (Chen et al., 2020) adopts cosine similarity to measure the feature distances between positive pair of examples. A recent study (Kornblith et al., 2021) shows that several loss functions, including logit normalization and cosine softmax, lead to higher accuracy on ImageNet but degrade the performance of transfer tasks. In addition, GODIN (Hsu et al., 2020) and Cosine loss (Techapanurak & Okatani, 2019) adopt cosine similarity for better performance on OOD detection. As discussed in Section 5, our method is superior to these cosine-based methods, since it is applicable to existing scoring functions and achieves strong performance in OOD detection.

Confidence calibration. Confidence calibration has been studied in various contexts in recent years. Some works address the miscalibration problem by post-hoc methods, such as Temperature Scaling (Platt et al., 1999; Guo et al., 2017) and Histogram Binning (Zadrozny & Elkan, 2001). Besides, some regularization methods are also proposed to improve the calibration quality of deep neural networks, like weight decay (Guo et al., 2017), label smoothing (Szegedy et al., 2016; Müller et al., 2019), and focal loss (Lin et al., 2017; Mukhoti et al., 2020). Conformal prediction based method (Lei et al., 2013) outputs the empty set as prediction in case of too high "atypicality". Top-label calibration aims to calibrate the reported probability for the predicted class label (Gupta & Ramdas, 2022). Recent work (Wang et al., 2021a) shows that these regularization methods make it harder to further improve the calibration performance with post-hoc methods. LogitNorm loss yields better calibration performance with Temperature Scaling than cross-entropy.

7. Conclusion

In this paper, we introduce Logit Normalization (LogitNorm), a simple alternative to the cross-entropy loss that enhances many existing post-hoc methods for OOD detection. By decoupling the influence of logits' norm from the training objective and its optimization, the model tends to give conservative predictions for OOD inputs, resulting in a stronger separability from ID data. Extensive experiments show that LogitNorm can improve both OOD detection and confidence calibration while maintaining the classification accuracy on ID data. This method can be easily adopted in practical settings. It is straightforward to implement with existing deep learning frameworks, and does not require sophisticated changes to the loss or training scheme. We hope that our insights inspire future research to further explore loss function design for OOD detection.

Acknowledgements

This research is supported by MOE Tier-1 project RG13/19 (S). LF is supported by the National Natural Science Foundation of China (Grant No. 62106028). YL is supported by Wisconsin Alumni Research Foundation (WARF), Facebook Research Award, and a Google-Initiated Focused Research Award.

References

Bendale, A. and Boult, T. E. Towards open set deep networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1563–1572, 2016.

Bevandić, P., Krešo, I., Oršić, M., and Šegvić, S. Discriminative out-of-distribution detection for semantic segmentation. *arXiv preprint arXiv:1808.07703*, 2018.

Chen, J., Li, Y., Wu, X., Liang, Y., and Jha, S. Atom: Robustifying out-of-distribution detection using outlier mining. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 430–445. Springer, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.

Du, X., Wang, X., Gozum, G., and Li, Y. Unknown-aware object detection: Learning what you don't know from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022a.

Du, X., Wang, Z., Cai, M., and Li, Y. Vos: Learning what you don't know by virtual outlier synthesis. In *Proceedings of the International Conference on Learning Representations*, 2022b.

Forst, W. and Hoffmann, D. *Optimization—Theory and Practice*. Springer Science & Business Media, 2010.

Geifman, Y. and El-Yaniv, R. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning*, pp. 2151–2159. PMLR, 2019.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017.

Gupta, C. and Ramdas, A. Top-label calibration and multiclass-to-binary reductions. In *International Conference on Learning Representations*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. *Proceedings of the International Conference on Learning Representations*, 2019.

Hsu, Y.-C., Shen, Y., Jin, H., and Kira, Z. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10951–10960, 2020.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

Huang, R., Geng, A., and Li, Y. On the importance of gradients for detecting distributional shifts in the wild. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Jeong, T. and Kim, H. Ood-maml: Meta-learning for few-shot out-of-distribution detection and classification. *Advances in Neural Information Processing Systems*, 33, 2020.

Katz-Samuels, J., Nakhleh, J., Nowak, R., and Li, Y. Training ood detectors in their natural habitats. In *International Conference on Machine Learning (ICML)*. PMLR, 2022.

Kornblith, S., Chen, T., Lee, H., and Norouzi, M. Why do better loss functions lead to less transferable features? In *Advances in Neural Information Processing Systems*, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31, 2018.

Lei, J., Robins, J., and Wasserman, L. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108(501):278–287, 2013.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

Lin, T.-Y., Goyal, P., Girshick, R. B., He, K., and Dollár, P. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 212–220, 2017.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Malinin, A. and Gales, M. Predictive uncertainty estimation via prior networks. *arXiv preprint arXiv:1802.10501*, 2018.

Ming, Y., Fan, Y., and Li, Y. Poem: Out-of-distribution detection with posterior sampling. In *International Conference on Machine Learning (ICML)*. PMLR, 2022a.

Ming, Y., Sun, Y., Dia, O., and Li, Y. Cider: Exploiting hyperspherical embeddings for out-of-distribution detection. *arXiv preprint arXiv:2203.04450*, 2022b.

Mohseni, S., Pitale, M., Yadawa, J., and Wang, Z. Self-supervised learning for generalizable out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5216–5223, 2020.

Morteza, P. and Li, Y. Provable guarantees for understanding out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P. H., and Dokania, P. K. Calibrating deep neural networks using focal loss. 2020.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2901–2907, 2015.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035. 2019.

Platt, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

Ranjan, R., Castillo, C. D., and Chellappa, R. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

Sastry, C. S. and Oore, S. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, pp. 8491–8501, 2020.

Sehwag, V., Chiang, M., and Mittal, P. Ssd: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2021.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, 2016.

Sun, Y., Guo, C., and Li, Y. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34, 2021.

Sun, Y., Ming, Y., Zhu, X., and Li, Y. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning (ICML)*. PMLR, 2022.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Tack, J., Mo, S., Jeong, J., and Shin, J. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, 2020.

Techapanurak, E. and Okatani, T. Hyperparameter-free out-of-distribution detection using softmax of scaled cosine similarity. *arXiv:1905.10628*, 2019.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

Wang, D.-B., Feng, L., and Zhang, M.-L. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. In *Advances in Neural Information Processing Systems*, 2021a.

Wang, F., Xiang, X., Cheng, J., and Yuille, A. L. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049, 2017.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274, 2018.

Wang, H., Liu, W., Bocchieri, A., and Li, Y. Can multi-label classification networks know what they don't know? *Advances in Neural Information Processing Systems*, 34, 2021b.

Wei, H., Tao, L., Xie, R., and An, B. Open-set label noise can improve robustness against inherent label noise. In *Advances in Neural Information Processing Systems*, 2021.

Wei, H., Tao, L., Xie, R., Feng, L., and An, B. Open-sampling: Exploring out-of-distribution data for rebalancing long-tailed datasets. In *International Conference on Machine Learning (ICML)*. PMLR, 2022.

Wu, Z., Xiong, Y., Stella, X. Y., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010.

Xu, J., Sun, X., Zhang, Z., Zhao, G., and Lin, J. Understanding and improving layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Xu, P., Ehinger, K. A., Zhang, Y., Finkelstein, A., Kulkarni, S. R., and Xiao, J. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Zadrozny, B. and Elkan, C. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning*, pp. 609–616, 2001.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, X., Zhao, R., Qiao, Y., Wang, X., and Li, H. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10823–10832, 2019.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017.

Zhu, Z., Dong, Z., and Liu, Y. Detecting corrupted labels without training a model to predict. In *International Conference on Machine Learning (ICML)*. PMLR, 2022.

A. Proof of Proposition 3.1

From Eq. (2), we have $\boldsymbol{f} = \|\boldsymbol{f}\| \cdot \hat{\boldsymbol{f}}$. Then,

$$\arg\max_i(f_i) = \arg\max_i(\|\boldsymbol{f}\| \cdot \hat{f}_i) = \arg\max_i(\hat{f}_i).$$

Similarly, for any given scalar $s > 1$, we have,

$$\arg\max_i(sf_i) = \arg\max_i(s\|\boldsymbol{f}\| \cdot \hat{f}_i) = \arg\max_i(\hat{f}_i).$$

Thus Proposition 3.1 is proved. □

B. Proof of Proposition 3.2

From Proposition 3.1, we have

$$\arg\max_i(f_i) = \arg\max_i(sf_i) = c.$$

Let $f_c = \max_i(f_i)$, and $t = s - 1$, then we have,

$$\begin{aligned}\sigma_c(s\boldsymbol{f}) &= \frac{e^{(1+t)f_c}}{\sum_{j=1}^n e^{(1+t)f_j}} \\ &= \frac{e^{f_c}}{\sum_{j=1}^n e^{f_j + t(f_j - f_c)}}.\end{aligned}$$

For any $j \in [1, 2, \ldots, n]$, we have, $f_j - f_c \leq 0$. Then

$$\sigma_c(s\boldsymbol{f}) \geq \frac{e^{f_c}}{\sum_{j=1}^n e^{f_j}} = \sigma_c(\boldsymbol{f}).$$

Thus Proposition 3.2 is proved. □

C. Proof of Proposition 3.3

Let $\widetilde{\boldsymbol{f}} = \boldsymbol{f}/(\tau\|\boldsymbol{f}\|)$, then we have $\|\widetilde{\boldsymbol{f}}\| = 1/\tau$.

That is, $\sum_{i=1}^k \widetilde{\boldsymbol{f}}_i^{\,2} = \|\widetilde{\boldsymbol{f}}\|^2 = 1/\tau^2$.

Hence,

$$-1/\tau \leq \widetilde{\boldsymbol{f}}_i \leq 1/\tau, \forall\, i \in 1, \ldots, k.$$

Let $\sigma(\widetilde{\boldsymbol{f}}) = \frac{e^{\widetilde{\boldsymbol{f}}_y}}{\sum_{i=1}^k e^{\widetilde{\boldsymbol{f}}_i}}$, then we have

$$\begin{aligned}\sigma(\widetilde{\boldsymbol{f}}) &\leq \frac{e^{1/\tau}}{e^{1/\tau} + (k-1)e^{-1/\tau}} \\ &= \frac{1}{1 + (k-1)e^{-2/\tau}}\end{aligned}$$

Hence,

$$\begin{aligned}\mathcal{L}_{\text{logit\_norm}} &= -\log(\sigma(\widetilde{\boldsymbol{f}})) \\ &\geq -\log\frac{1}{1 + (k-1)e^{-2/\tau}} \\ &= \log(1 + (k-1)e^{-2/\tau})\end{aligned}$$

Thus Proposition 3.3 is proved. □

D. Descriptions of OOD Datasets

Following the prior literature, we use six OOD test datasets: *Textures* (Cimpoi et al., 2014) is a dataset of describable textural images. *SVHN* dataset (Netzer et al., 2011) contains 32 × 32 color images of house numbers, which has ten classes comprised of the digits 0-9. *LSUN* (Yu et al., 2015) is another scene understanding dataset with fewer classes than Places365. Here we use *LSUN-C* and *LSUN-R* to denote the cropped and resized version of the LSUN dataset respectively. *iSUN* (Xu et al., 2015) is a large-scale eye tracking dataset, selected from natural scene images of the SUN database (Xiao et al., 2010). *Places365* (Zhou et al., 2017) consists in images for scene recognition rather than object recognition.

E. Future Work

In this paper, we introduce a simple fix to the cross-entropy loss that enhances existing post-hoc methods for detecting OOD instances. We expect the observations and analyses in this work could inspire the future design of loss functions for OOD detection. Some future works include:

Theoretical understanding. In this work, we empirically show that Logit Normalization can significantly improve OOD detection performance. On the theoretical side, we only present an analysis to show why the softmax cross-entropy loss encourages to produce logits with larger magnitudes, leading to the overconfidence issue that makes it challenging to distinguish ID and OOD examples. In the future work, we hope to provide a more rigorous theoretical justification to analyze how the LogitNorm loss improves OOD detection.

Hyperparameter tuning. In our experiments, we tune the hyperparameter τ with a validation set – Gaussian noises. Although the proposed method can achieve significant improvement after tuning, the tuning process is computationally expensive because it needs to train multiple models. Therefore, we expect that future work will be able to automatically adjust τ during training.

F. Detailed Experimental Results

We report the performance of OOD detectors on each OOD test dataset in Table 7, 8, and 9. In particular, Table 7 shows the detail performance of LogitPenalty and GODIN methods. Table 8 shows the detail performance of different scoring functions with CE loss and LogitNorm loss. Table 9 shows the detail performance of CE loss and LogitNorm loss with different model architectures.

Table 7. OOD detection performance comparison with Logit Penalty ($\lambda = 0.05$) and GODIN methods. We train WRN-40-2 (Zagoruyko & Komodakis, 2016) on CIFAR-10 dataset. All values are percentages.

Method	LogitPenalty			GODIN		
OOD dataset	FPR95 ↓	AUROC ↑	AUPR ↑	FPR95 ↓	AUROC ↑	AUPR ↑
Texture	62.26	72.00	87.29	34.90	92.84	98.41
SVHN	62.02	71.87	87.21	25.17	95.48	99.04
LSUN-C	54.71	66.23	82.29	15.79	96.88	99.33
LSUN-R	51.31	79.24	90.64	16.82	97.00	99.40
iSUN	51.91	79.70	91.17	23.25	95.85	99.15
Places365	63.51	70.60	87.09	35.51	93.39	98.23

Table 8. OOD detection performance comparison using cross-entropy loss and LogitNorm loss. We use WRN-40-2 (Zagoruyko & Komodakis, 2016) to train on the in-distribution datasets and use softmax confidence score as the scoring function. All values are percentages. ↑ indicates larger values are better, and ↓ indicates smaller values are better. **Bold** numbers are superior results.

ID dataset		CIFAR-10			CIFAR-100		
OOD dataset	Score	FPR95 ↓	AUROC ↑	AUPR ↑	FPR95 ↓	AUROC ↑	AUPR ↑
		Cross-entropy loss / **LogitNorm loss (ours)**					
Texture	ODIN	59.86 / **26.38**	76.35 / **94.66**	92.09 / **98.68**	80.23 / **70.64**	75.60 / **78.19**	93.49 / **93.35**
	Energy	51.02 / **35.13**	84.60 / **92.94**	94.90 / **98.29**	80.89 / **79.17**	75.59 / **76.94**	93.48 / **93.29**
	GradNorm	78.50 / **27.26**	55.15 / **93.11**	82.96 / **98.11**	86.51 / **69.07**	55.76 / **75.21**	84.28 / **91.34**
SVHN	ODIN	53.92 / **9.17**	82.98 / **98.29**	95.17 / **99.63**	83.52 / **45.41**	79.60 / **92.59**	95.34 / **98.48**
	Energy	21.27 / **10.35**	95.70 / **97.73**	99.05 / **99.55**	84.97/ **62.41**	79.41 / **89.58**	95.31 / **97.86**
	GradNorm	58.95 / **5.51**	76.75 / **98.91**	93.01 / **99.76**	97.7 / **39.79**	54.65 / **93.21**	86.6 / **98.57**
LSUN-C	ODIN	13.31 / **1.65**	97.14 / **99.59**	99.35 / **99.91**	37.45 / **13.08**	93.01 / **97.66**	98.50 / **99.50**
	Energy	9.63 / **4.15**	98.01 / **98.64**	99.56 / **99.73**	33.5 / **28.68**	93.74 / **95.33**	98.64 / **99.04**
	GradNorm	22.03 / **1.14**	93.04 / **99.70**	98.12 / **99.94**	43.43 / **8.95**	90.72 / **98.28**	97.84 / **99.63**
LSUN-R	ODIN	27.21 / **4.71**	93.52 / **98.86**	98.41 / **99.78**	69.69 / **68.3**	83.51 / **84.78**	96.1 / **96.52**
	Energy	16.5 / **13.90**	96.31 / **97.31**	99.1 / **99.49**	70.45 / **68.84**	83.57 / **85.84**	96.11 / **96.84**
	GradNorm	52.41 / **16.31**	77.70 / **97.18**	92.85 / **99.45**	98.34 / **83.51**	34.18 / **76.32**	76.53 / **94.15**
iSUN	ODIN	33.31 / **5.65**	92.03 / **98.79**	98.07 / **99.76**	74.47 / **71.11**	81.01 / **83.82**	95.43 / **96.24**
	Energy	19.74 / **16.00**	95.69 / **97.10**	98.98 / **99.45**	75.66/ **72.94**	80.99 / **84.48**	95.43 / **96.49**
	GradNorm	59.08 / **13.76**	74.64 / **97.58**	91.77 / **99.53**	99.41 / **82.51**	32.47 / **77.39**	75.58 / **94.38**
Places365	ODIN	54.32 / **30.12**	81.25 / **94.04**	94.33 / **98.63**	78.93/ **80.23**	75.55 / **76.84**	93.37 / **94.04**
	Energy	42.75 / **35.31**	88.09 / **92.84**	96.5 / **98.30**	79.72 / **80.73**	75.4 / **76.86**	93.34 / **94.08**
	GradNorm	82.86 / **42.67**	56.46 / **91.55**	85.79 / **98.04**	96.67 / **87.52**	49.29 / **68.02**	83.88 / **91.05**

Table 9. OOD detection performance comparison using cross-entropy loss and LogitNorm loss with ResNet-34 and DenseNet-BC. In-distribution dataset is CIFAR-10. All values are percentages. **Bold** numbers are superior results.

Model architecture	ResNet-34			DenseNet		
OOD dataset	FPR95 ↓	AUROC ↑	AUPR ↑	FPR95 ↓	AUROC ↑	AUPR ↑
	Cross-entropy loss / **LogitNorm loss (ours)**					
Texture	56.38 / **30.68**	87.82 / **94.05**	96.32 / **98.52**	66.14 / **39.79**	85.47 / **91.46**	96.05 / **97.78**
SVHN	52.34 / **4.72**	89.76 / **98.98**	97.34 / **99.79**	57.26 / **18.66**	91.09 / **96.81**	98.16/ **99.36**
LSUN-C	32.10 / **0.51**	95.64 / **99.77**	99.14 / **99.95**	32.30 / **2.73**	95.59 / **99.35**	99.12 / **99.87**
LSUN-R	43.31 / **14.19**	93.41 / **97.65**	98.57 / **99.54**	43.85 / **5.41**	94.04 / **98.70**	98.82 / **99.75**
iSUN	45.80 / **14.83**	92.62 / **97.47**	98.36 / **99.51**	43.08 / **5.73**	94.13 / **98.68**	98.83 / **99.74**
Places365	56.48 / **29.98**	87.57 / **94.16**	96.50 / **98.63**	60.26 / **38.70**	88.26 / **91.98**	97.19/ **98.09**