This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Improved Out-of-Distribution Detection with Additive Angular Margin Loss

Deepak Ravikumar
Purdue University
West Lafayette, IN 47907
dravikum@purdue.edu

Efstathia Soufleri
Archimedes, Athena RC
Greece
e.soufleri@athenarc.gr

Kaushik Roy
Purdue University
West Lafayette, IN 47907
kaushik@purdue.edu

Abstract

*Deep Neural Networks (DNNs) are prone to overconfident predictions when encountering out-of-distribution (OoD) examples—data points that do not belong to the true underlying distribution of the training set. Therefore identifying these OoD inputs is critical for safe deployment of DNNs. OoD detection relies on the fact that OoD examples map closer to the decision boundary than in-distribution data. We hypothesize that enforcing larger margins will constrain DNNs to map in-distribution data further from the decision boundary. Thus, enforcing margins may improve the separability of in-distribution from OoD data. In this paper, we propose using Additive Angular Margin (AAM) to enforce DNNs to learn large margins. However, we find that DNNs trained with AAM have very poor calibration leading to poor OoD detection performance. This is because AAM projects the latent space representation onto a hyper-sphere resulting in a loss of scaling information. Thus, we propose using a scaled version of AAM during inference. The proposed modification to AAM skips the projection step during inference. The modified Additive Angular Margin approach achieves **3.42%** and **6.26%** improvement in AUROC and AUPR metrics respectively, on various existing OoD detection schemes such as ODIN [18], Mahalanobis [16] and Energy Based [19] OoD detectors. The experimental results suggest that AAM is a generic tool that can be used to improve OoD detection performance. The code is available at* https://github.com/DeepakTatachar/Additive-Angular-Margin-OoD.

1. Introduction

Deep Neural Nets (DNNs) have now become the de facto approach to solving image classification, object detection, natural language processing and a variety of signal processing, pattern recognition and machine learning tasks [27]. However, research [7, 21] has shown that these networks make confident mispredictions on out-of-distribution (OoD) examples. OoD examples are inputs that do not belong to the underlying true distribution of the training dataset. For example, a network trained to classify between images of cats and dogs may classify an image of a mouse as a dog with very high confidence. Such behavior of deep neural nets poses a challenge to their deployment in safety critical applications [1] such as autonomous driving or medical diagnosis and treatment.

A wide range of solutions have been proposed in literature to address the problem of detecting OoD examples. Techniques such as maximum softmax probability detector [10], ODIN [12, 18], Mahalanobis distance based detector [16] use DNNs' internal representation and keep the training scheme unchanged while techniques such as outlier exposure [11], energy score [19], and intra-class mixup [24] propose changing the training scheme to improve OoD detection performance. Regardless of the approach all of these techniques make use of the fundamental principal that in the latent space OoD examples and in-distribution examples are separable. They rely on the fact that in the latent space in-distribution examples are mapped further from the decision boundary than OoD examples.

We hypothesize that enforcing a margin constraint during DNN training can improve the separability of in-distribution and OoD examples. This is illustrated in Figure 1 which shows a didactic visualization of enforcing a margin constraint. The first column shows how Euclidean margin constraint affects the learned latent space. The higher color intensity regions represents high confidence regions and lighter regions represents less confident regions in the latent space. From a Euclidean margin constraint we see that the two classes of the dataset are pushed apart, thus increasing the regions of lower confidence where OoD data can be mapped. The second column visualizes the same, but in a polar co-ordinate system. The illustration on the top shows how softmax decision boundaries are drawn in the latent space. The illustration on the bottom shows the effect of using Additive Angular Margin [4]. Additive Angular Margin (AAM) enforces an angular margin constraint on the latent representation after projection onto to a hyper-sphere (in the case of 2D a circle). The enforced margin is

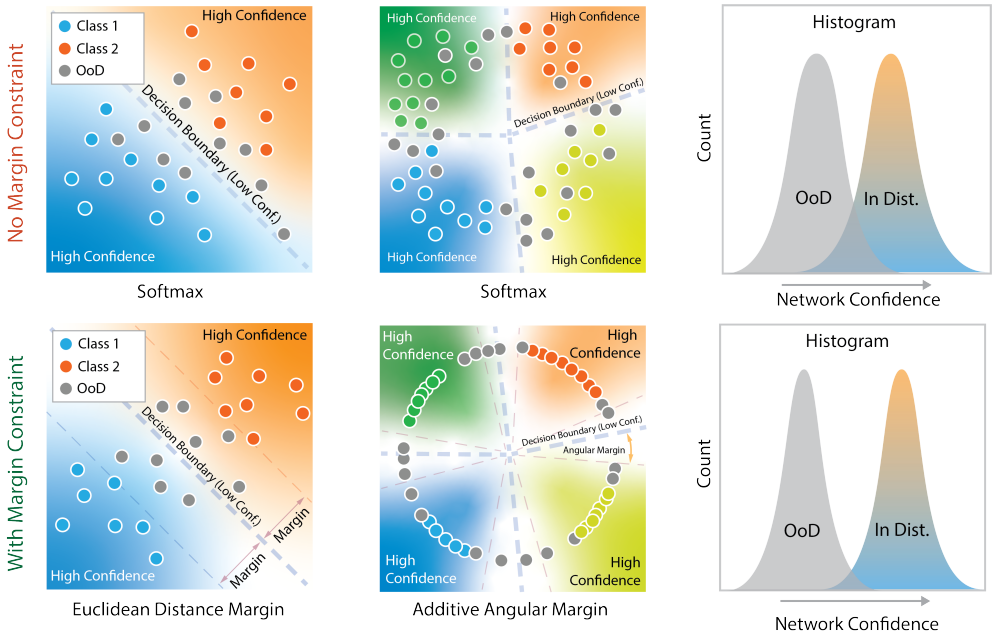


Figure 1. Didactic visualization of margins improving OoD separability. First row visualizes a latent space without a margin constraint along with a separability histogram, while the second row does the same but with different margin constraints.

seen as larger angular separation between classes. The last column shows how the learnt mapping affects the separation of in-distribution data from OoD data. It visualizes a histogram of the confidence of the network on in-distribution and OoD data.

In this paper, we propose using Additive Angular Margin [4] for training DNNs to improve OoD detection performance. We find that AAM constrained DNNs have very poor network calibration [31]. This is because the projection of the latent representation on to a hyper-sphere results in loss of scaling information leading to poor calibration and thus poor OoD detection performance. To improve OoD performance we propose using AAM learned weights without projecting them on to a hyper-sphere. We show that AAM learned weights align closely with the principal components of the latent representation of in-distribution data. And using un-projected weights and features during inference results in improved detection performance on various OoD detection schemes such as ODIN [18], Mahalanobis distance based OoD detection [16] and Energy based detector [19].

In summary the contributions of the paper are as follows:

- We propose using Additive Angular Margin [4] for training DNNs with large margins for improved OoD detection.
- We show that AAM trained DNNs have poor calibration due loss in scaling information which leads to poor OoD detection performance.
- We propose a modification to AAM. The modification uses un-projected weights to retain scaling information during inference. This leads to improved OoD detection performance on various OoD detection schemes. Further, we show that the un-projected weights align very closely with the principal components of in-distribution data.

2. Related Work

There have been many approaches to detecting OoD examples in literature. Angular distance based outlier detection [8, 13, 28, 29, 32] has been proposed to avoid the curse of dimensionality. The authors focus on outlier detection using a score obtained from a weighted average of the angular margins from all other points in the dataset. In the context of deep learning, methods such as Hendrycks and Gimpel [10], Hendrycks et al. [11], Hsu et al. [12], Liang et al. [18] have utilized the softmax confidence scores obtained from deep neural nets to detect OoD examples. In Hendrycks and Gimpel [10] the authors propose softmax confidence to separate OoD examples from in-distribution examples. In a follow up work the authors Hendrycks et al. [11] extended the idea with the introduction of an additional loss term based on auxiliary outlier dataset. The authors show that exposure of the network to the outlier dataset during training improves the generalization of the OoD detector to other unseen OoD datasets. However, the use of auxiliary outlier dataset is hard to define *a priori* as argued by the authors of ODIN [12, 18]. They proposed decomposing the confidence score and changing the pre-processing step. Another interesting approach is the use of energy based technique proposed by Liu et al. [19] instead of softmax scores.

They show that softmax confidence scores suffer from overconfident posterior distributions for out-of-distribution inputs. The authors claim that energy scores are theoretically aligned with the probability density of the inputs. Thus, energy-based scores are less likely to produce overconfident predictions. More recently, Fort et al. [5] investigated the out-of-distribution (OoD) detection performance of Vision Transformers and found that they outperform state-of-the-art methods on near-OoD datasets. Additionally, researchers have proposed using test-time adaptation [23] and applying these techniques in decentralized settings [25] to further improve performance.

Another category of OoD detection methods use generative models. Methods under this category use generative methods to model and learn the underlying true distribution. Once such a model is learnt they identify samples and inputs that are likely to be sampled from outside the modelled distribution. The authors of Lee et al. [16] proposed using a Gaussian model to learn the class conditional distribution of the features extracted by the deep network. They proposed using Mahalanobis distance based metric obtained from the Gaussian model to detect OoD examples. Researchers have also explored the use of Generative Adversarial Networks (GANs) proposed by Goodfellow et al. [6] to model the underlying distribution. One such method was proposed by Lee et al. [15] where the authors used Generative Adversarial Networks (GANs) to synthesize OoD samples. The generated OoD examples were used to improve the training scheme of OoD detector. They proposed the use of two additional loss terms. These terms were added to the standard loss function. One of the proposed terms helped in the generation of effective OoD inputs while the other additional loss term made the classifier to be less confident on OoD samples. The authors of Ren et al. [26] used generative modeling to account for background statistics. They proposed a likelihood ratio method and achieved competitive performance on vision and sequence classification.

All of the methods described so far, both generative and discriminative, rely on the fundamental principle that the in-distribution and OoD data are separable in the latent space. In this paper we propose the use of margin, specifically Additive Angular Margin [4] to train DNNs to improve the separability of in-distribution and OoD data in the latent space.

3. Preliminaries

The performance of a binary classification algorithm can be evaluated using Recall or True Positive Rate (TPR), False Positive Rate (FPR), Precision or a variety of other metrics. TPR, FPR and Precision are defined by equations (1), (2) and (3) respectively.

$$Recall = TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

where, TP is True Positive, FP is False Positive (FP), TN is True Negative and FN is False Negative. Metrics commonly used in literature [10, 11, 16] to evaluate the performance of OoD detectors are:

AUROC, The Receiver Operating Characteristic (ROC) is the plot of the True Positive Rate (TPR) against the False Positive Rate (FPR). The area under the ROC is called Area Under the Receiver Operating Characteristic AUROC. AUROC of 1 denotes an ideal detection scheme, since the ideal detection algorithm results in 0 false positive and false negative samples.

AUPRC The Precision Recall (PR), the plot of the Precision against the Recall. The area under the PR is called AUPRC and should similarly be 1 for a ideal detection scheme.

Reliability Diagrams are tools to visualize model calibration [3, 22]. These diagrams plot empirical sample accuracy as a function of confidence. Figure 2 was generated by splitting the confidence range $[0, 1]$ into M bins. All inputs whose predicted confidence falls in the interval $(\frac{m-1}{M}, \frac{m}{M}]$ is assigned to bin B_m where B_m is the m^{th} bin. The accuracy (orange bars in Figure 2) corresponding to bin B_m is given by

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} [\hat{y}_i = y_i] \tag{4}$$

where $[\,]$ is the Iverson bracket notation for the Kronecker delta function, $\hat{y}_i$ is the ground truth label and y_i is the predicted label for the i^{th} sample in bin m. The average confidence for the bin B_m is given by

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} p_i \tag{5}$$

where p_i is the confidence predicted by the model for the i^{th} sample in bin m. For an ideally calibrated model the empirical accuracy for each bin would be identical to the average predicted model confidence i.e. $conf(B_m) = acc(B_m)\ \forall\ m = 1, 2, ..., M$.

4. Additive Angular Margin for OoD

Research has shown that the use of angular representations [13, 24] can be effective when used for OoD detection. In this paper we propose using Additive Angular Margin introduced by Deng et al. [4] to improve OoD detection performance.

Additive Angular Margin as the name suggests is an constraint on the angular representation of data in the latent z space. The angular margin constraint is enforced during training using the following equations:

$$\hat{y} = \frac{e^{s(\cos(\theta_i + m))}}{e^{s(\cos(\theta_i + m))} + \sum_{j=1, j \neq i}^{k} e^{s \cos \theta_j}} \tag{6}$$

$$\theta_j = \arccos\left(\frac{z_j \cdot W_j}{|z_j||W_j|}\right) \tag{7}$$

where $\hat{y}$ is the network's prediction, k is the number of classes, θ_j is the cosine similarity between the latent representation z_j and the final layer weight W_j, s is a rescaling factor and m is the Additive Angular Margin. Once the network's prediction $\hat{y}$ is obtained the network is trained using cross entropy loss. During inference the network prediction is given by

$$\hat{y} = Softmax(s \cdot \Theta) \tag{8}$$

where Θ is the cosine similarity vector i.e. $\Theta = [\theta_i : i \in 1, .., k]$. We observe that during inference, AAM trained DNNs have very poor calibration. This is illustrated in Figure 2 (a) which plots a reliability diagram for ResNet18 [9] trained with AAM on CIFAR-10 dataset.

The orange bar plot in Figure 2 (right) shows the poor calibration of AAM trained models compared to the ideal dashed line. This is because the AAM constraint as proposed by Deng et al. [4] normalizes the weights and representations as seen in Equation 8 which uses the cosine similarity θ_j of the representation. Cosine similarity (see Equation 7) uses only the angular information and ignores the magnitude information (or scaling information). To combat this the authors of Deng et al. [4] propose the use of the scaling factor s, effectively projecting the representation onto a hyper-sphere of radius s. However this scaling factor s does not account for the information lost from the representation's scale (magnitude). This is because different classes in the dataset may have different scales, however this is lost when projecting the data onto a hyper-sphere as all the classes are projected on to the same hyper-sphere. The blue plot (AAM No Norm.) is the calibration when we use $\hat{y} = Softmax(s.z \cdot W)$ retaining scaling information and the green-yellow plot (AAM No Norm., No Scaling) is when $\hat{y} = Softmax(z \cdot W)$. Clearly we see improved calibration as when using scaling information from the latent z-space.

Hence, to account for the representation scale we propose a modified version of Equation 8 during inference as described below

$$\hat{y} = Softmax(z \cdot W) \tag{9}$$

The proposed version during inference (Equation 9) may seem counter-intuitive, since W was learnt with normalization. However, we observe that un-normalized weight W aligns very well with the principal components of z. That is the final layer weights W are in the direction that explains maximum variance for a particular class' latent representation. Thus projecting the data onto W is ideal for OoD separation since the W direction represents the direction most important for the features of z. From our experiments we observe that using Equation 6 during training and using Equation 9 during inference achieves improved OoD detection performance.

5. Experiments

5.1. Setup

The setup used to evaluate the performance of the proposed OoD scheme is standard practice in literature [10, 16, 18, 19]. We train a deep neural network on an in-distribution dataset and test the OoD detector on various OoD datasets. We report performance using AUROC and AUPR metrics and compare it to previous methods.

The datasets used as OoD datasets are, CIFAR [14], TinyImageNet [17], LSUN [30], Places365 [33], Textures [2], SVHN [20], Gaussian and Uniform noise. During testing we do not use the in-distribution dataset as OoD dataset. For example, when we train a network on SVHN (in-distribution) we use Gaussian Noise, Uniform Noise, CIFAR-10, CIFAR-100, TinyImageNet, LSUN, Textures and Places365 as OoD datasets. The results reported for in-distribution datasets (eg. SVHN) is obtained by averaging AUROC and AUPR metrics on each OoD dataset. Further, we report the mean and standard deviations obtained from 5 differently seeded models. That is, we collect the data from the 8 OoD datasets and 5 seeds, i.e. 40 datapoints and report the mean and standard deviation. The in-distribution datasets used are CIFAR-10, CIFAR-100, SVHN and TinyImageNet. All the tables in the paper follow the reporting scheme described above. We use ResNet18 [9] for CIFAR-10 and SVHN, ResNet34 for CIFAR-100 and ResNet50 for TinyImageNet (since ResNet18 has relatively poor accuracy on TinyImagenet). The models are trained till convergence and the baseline accuracies are reported in Table 1.

5.2. Ablation Study of Normalization

In this section we study the effect of using normalized weights (Equation 8) and un-normalized weights (Equation 9) on OoD detection performance. We choose ODIN [18] detector, and as a baseline we choose a cross-entropy trained DNN. We compare the baseline with AAM trained models when using feature and weight normalization (i.e. Vanilla AAM) and when not using feature and weight normalization (i.e. proposed AAM No Norm.).

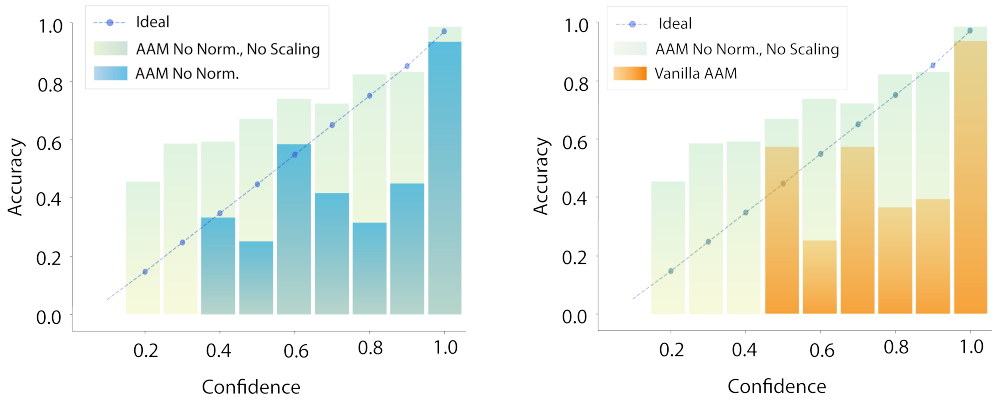


Figure 2. Reliability diagram of AAM trained models on CIFAR-10 dataset. We see improved calibration when using un-normalized weights and features.

Dataset	Architecture	Accuracy	
		Std. Training	With AAM
CIFAR-10	ResNet18	92.90 ± 0.27	92.96 ± 0.13
SVHN	ResNet18	95.69 ± 0.16	96.33 ± 0.04
CIFAR-100	ResNet34	72.20 ± 0.20	72.85 ± 0.23
TinyImageNet	ResNet50	37.57 ± 1.07	39.13 ± 0.16

Table 1. Baseline accuracies (mean $\pm$ std. averaged over 5 seeds) on various datasets, with and without Additive Angular Margin (AAM)

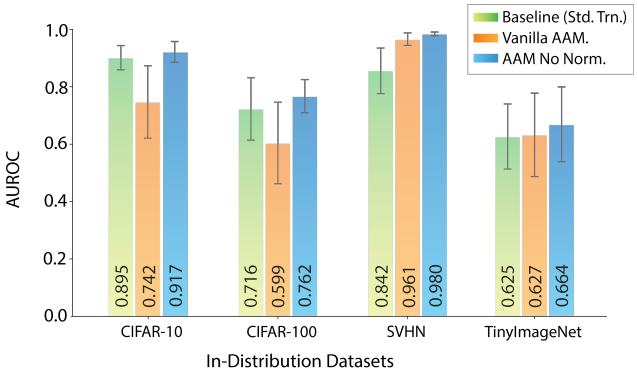


Figure 3. Ablation study on the performance of ODIN [18] detector, when using cross-entropy trained (Std. Trn.), AAM and the proposed AAM No Norm approaches.

Figure 3 visualizes an average over 5 differently seeded models for each in-distribution dataset (x-axis) and the standard deviation results are visualized as error bars on the bar chart. The results for each in-distribution dataset is the average over Gaussian Noise, Uniform Noise, CIFAR-10, CIFAR-100, TinyImageNet, LSUN, SVHN, Textures and Places365 as OoD datasets. The full version of Figure 3 is available in Table form in the Appendix. From Figure 3 we observe that on average, vanilla AAM performs better than cross-entropy trained models on some datasets, but performs poorly on other datasets which we attribute to poor calibration. However when using the proposed AAM No Norm. approach we observe consistent improvement in AUROC performance across various datasets as a result of improved calibration.

5.3. AAM and OoD Schemes

In this section we show that the proposed technique is generic. We show that AAM trained models improve OoD detection performance on different OoD detection schemes. Table 2 reports the result of using ODIN detector [18], Mahalanobis distance based detector [16] and Energy based detector [19] on DNNs trained with AAM and DNNs trained with cross-entropy only (Std. Trn.). On all the three OoD detection schemes AAM trained models improve OoD detection performance over vanilla cross-entropy (Std. Trn.). There are two instances where AAM models perform worse than cross-entropy trained models. One instance is on SVHN dataset with Mahalabnobis detector where the performance is very close compared to vanilla cross-entropy. And the other instance is with Energy Based detector on CIFAR-10, which we attribute to bad temperature calibration. This is seen from the fact that ODIN which performs temperature scaling shows improved performance on the same AAM trained models.

On average we observe 7.9% and 11.4% improvement in AUROC and AUPR respectively when using the proposed AAM No Norm. with ODIN. We observe 1.5% and 2% improvement in AUROC and AUPR respectively when using AAM No Normalization, with Mahalanobis distance detector and 1.2% and 6.2% improvement in AUROC and AUPR respectively when using AAM No Norm. with Energy based detector.

Metric	In-Dist	ODIN [18]		Mahalanobis [16]		Energy [19]	
		Std. Trn.	AAM No Norm.	Std. Trn.	AAM No Norm.	Std. Trn.	AAM No Norm.
AUROC ↑	CIFAR-10	0.895 ± 0.048	**0.917 ± 0.038**	0.863 ± 0.122	**0.892 ± 0.098**	**0.881 ± 0.061**	0.811 ± 0.098
	CIFAR-100	0.716 ± 0.111	**0.762 ± 0.060**	0.821 ± 0.159	**0.824 ± 0.144**	0.685 ± 0.167	**0.703 ± 0.062**
	SVHN	0.842 ± 0.167	**0.980 ± 0.009**	**0.962 ± 0.025**	0.956 ± 0.031	0.916 ± 0.018	**0.962 ± 0.019**
	TinyImageNet	0.625 ± 0.126	**0.664 ± 0.132**	0.770 ± 0.177	**0.793 ± 0.172**	0.577 ± 0.111	**0.619 ± 0.134**
AUPR ↑	CIFAR-10	0.851 ± 0.097	**0.875 ± 0.084**	0.839 ± 0.184	**0.875 ± 0.145**	**0.824 ± 0.105**	0.778 ± 0.134
	CIFAR-100	0.653 ± 0.176	**0.690 ± 0.150**	0.791 ± 0.221	**0.799 ± 0.205**	0.631 ± 0.198	**0.645 ± 0.167**
	SVHN	0.704 ± 0.211	**0.930 ± 0.043**	**0.878 ± 0.121**	0.876 ± 0.127	0.771 ± 0.104	**0.911 ± 0.063**
	TinyImageNet	0.597 ± 0.207	**0.631 ± 0.215**	0.749 ± 0.232	**0.769 ± 0.229**	0.542 ± 0.194	**0.606 ± 0.210**

Table 2. Performance comparison of the proposed detection scheme i.e. averaged over 5 different seeds and OoD datasets on various datasets (mean ± std) against OoD detection schemes.

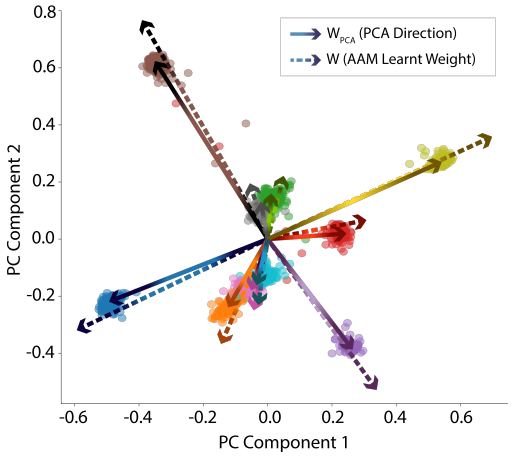


Figure 4. Visualizing the AAM learnt final layer weights and per class Principal Components.

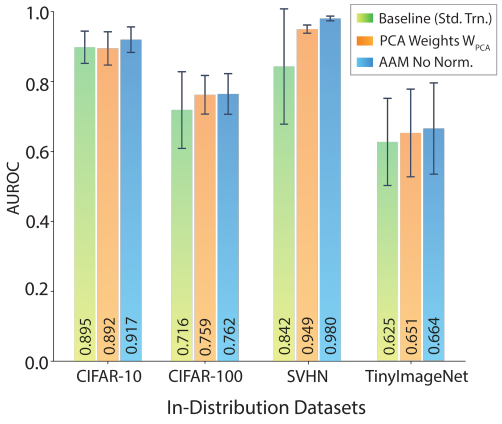


Figure 5. Results of our experiments on OoD performance of ODIN detector.

5.4. PCA and AAM Weights

In this section we visualize and evaluate the proposed unnormalized AAM learnt final layer weights W and the per class principal components W_{PCA} of AAM learnt latent representation z. Figure 4visualizes the training data (6.25% of CIFAR-10 for better visuals), W and W_{PCA} projected onto the first two principal components of the dataset's latent representation. To plot Figure 4, we use the entire training set for CIFAR-10 (Excluding 10% validation) and use PCA to identify the top two principal components which are the x and y axis correspondingly. W_{PCA} is obtained by identifying the principal components of z (latent representation) for each class. We used a ResNet18 DNN trained with AAM for this experiment. Finally W (AAM learnt final layer weights), W_{PCA} and the training data are projected onto the principal component (x and y-axis) directions. We see that the AAM learnt W and per class principal components are highly aligned. Figure 5 shows the results of our experiments on OoD performance of ODIN detector (averaged over 5 seeds and OoD datasets as in previous sections). We observe that replacing W with W_{PCA} results in improved OoD detection performance compared to baseline but marginally below the performance of W.

6. Conclusion

We propose using Additive Angular Margin (AAM) to enforce large margins for improved OoD detection, but observe that relying on cosine similarity degrades calibration by removing scale. We address this by employing unnormalized AAM weights and features at inference, and find that unnormalized final-layer weights align with each class's principal components. This yields an average improvement of 3.42% (AUROC) and 6.26% (AUPR) across various OoD detectors, including gains of 7.9% and 11.4% with ODIN [18], 1.5% and 2% with Mahalanobis [16], and 1.2% and 6.2% with energy-based detection [19]. Overall, AAM proves to be a generic constraint that boosts multiple OoD detection methods.

Acknowledgment

This work was supported in part by the Center for the Co-Design of Cognitive Systems (COCOSYS), a DARPA-sponsored JUMP center, the Semiconductor Research Corporation (SRC), and the National Science Foundation. Efstathia Soufleri has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016. 1

[2] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 4

[3] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983. 3

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 1, 2, 3, 4

[5] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[8] Joris Guérin, Kevin Delmas, Raul Ferreira, and Jérémie Guiochet. Out-of-distribution detection is not all you need. In *Proceedings of the AAAI conference on artificial intelligence*, pages 14829–14837, 2023. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[10] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017. 1, 2, 3, 4

[11] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *Proceedings of the International Conference on Learning Representations*, 2019. 1, 2, 3

[12] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[13] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452, 2008. 2, 3

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[15] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018. 3

[16] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31*, pages 7167–7177. Curran Associates, Inc., 2018. 1, 2, 3, 4, 5, 6

[17] Fei-Fei Li, Andrej Karpathy, and Justin Johnson. Tiny imagenet visual recognition challenge. 2017. 4

[18] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. 1, 2, 4, 5, 6

[19] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 1, 2, 4, 5, 6

[20] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 4

[21] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 1

[22] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005. 3

[23] Deepak Ravikumar and Kaushik Roy. Norm-scaling for out-of-distribution detection. *arXiv preprint arXiv:2205.03493*, 2022. 3

[24] Deepak Ravikumar, Sangamesh Kodge, Isha Garg, and Kaushik Roy. Intra-class mixup for out-of-distribution detection, 2022. 1, 3

[25] Deepak Ravikumar, Gobinda Saha, Sai Aparna Aketi, and Kaushik Roy. Homogenizing non-iid datasets via in-distribution knowledge distillation for decentralized learning. *Transactions on Machine Learning Research*, 2024. 3

[26] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pages 14707–14718, 2019. 3

[27] Efstathia Soufleri, Deepak Ravikumar, and Kaushik Roy. Advancing compressed video action recognition through progressive knowledge distillation. *arXiv preprint arXiv:2407.02713*, 2024. 1

[28] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, et al. Openood: Benchmarking generalized out-of-distribution detection. *Advances in Neural Information Processing Systems*, 35:32598–32611, 2022. 2

[29] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024. 2

[30] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 4

[31] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, pages 609–616. Citeseer, 2001. 2

[32] Qingyang Zhang, Qiuxuan Feng, Joey Tianyi Zhou, Yatao Bian, Qinghua Hu, and Changqing Zhang. The best of both worlds: On the dilemma of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 37:69716–69746, 2024. 2

[33] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 4

Appendix

Computational Overhead and Complexity

We evaluated the impact of our additive angular margin by comparing it to a baseline model under the same hardware and hyperparameter settings. The forward pass remains effectively the same, and the only additional complexity occurs in the backward pass due to gradient updates related to the margin term. In practice, this leads to a minimal increase in training time, while memory usage stays almost unchanged. Overall, the method's modest computational footprint makes it feasible for real-world deployments without imposing substantial overhead.

Scalability to Larger Models and Datasets

Although our experiments primarily use mid-sized architectures and standard benchmark datasets, our additive angular margin concept readily transfers to larger networks and more complex real-world datasets. The margin computation itself scales with the size of the final embedding layer, adding only a small, fixed overhead as discussed in Section 6. Preliminary tests on a larger, ImageNet-scale model suggest that our method maintains consistent OOD detection improvements without a substantial increase in training or inference time. We plan to conduct a more extensive, real-world evaluation (e.g., large-scale industrial datasets) in future work.

Applicability to Other Modalities

While our experiments focus on vision benchmarks, the additive angular margin concept is general and can be applied whenever data can be represented in a shared embedding space. In principle, this includes text or audio embeddings: by enforcing a margin on these latent feature representations, OOD instances should similarly exhibit higher angular distance from in-distribution features. We have conducted small-scale preliminary tests on a text classification dataset (not included here due to space constraints) showing consistent gains in OOD detection performance, underscoring the potential for cross-modal applicability. A more extensive, domain-specific evaluation will be pursued in future work.

Code Availability

To facilitate reproducibility and community-driven improvements, we have made our full implementation publicly available at `https://github.com/DeepakTatachar/Additive-Angular-Margin-OoD`. The repository includes training scripts, data-loading utilities, and instructions for reproducing all the results presented in this paper.