IMPROVING ADVERSARIAL ROBUSTNESS WITH HYPERSPHERE EMBEDDING AND ANGULAR-BASED REGULARIZATIONS

Olukorede Fakorede Ashutosh Nirala Modeste Atsague Jin Tian*

Department of Computer Science, Iowa State University
{fakorede, aknirala, modeste, jtian}@iastate.edu

ABSTRACT

Adversarial training (AT) methods have been found to be effective against adversarial attacks on deep neural networks. Many variants of AT have been proposed to improve its performance. Pang et al. [1] have recently shown that incorporating hypersphere embedding (HE) into the existing AT procedures enhances robustness. We observe that the existing AT procedures are not designed for the HE framework, and thus fail to adequately learn the angular discriminative information available in the HE framework. In this paper, we propose integrating HE into AT with regularization terms that exploit the rich angular information available in the HE framework. Specifically, our method, termed angular-AT, adds regularization terms to AT that explicitly enforce weight-feature compactness and inter-class separation; all expressed in terms of angular features. Experimental results show that angular-AT further improves adversarial robustness.

***Index Terms*—** Adversarial Robustness, Deep Learning, Hypersphere Embedding, Adversarial Training

1. INTRODUCTION

The application of deep neural networks (DNNs) into various domains has raised skepticism due to the observed vulnerability of DNNs to adversarial examples. Adversarial examples are produced when small, imperceptible but well-crafted perturbations are added to, e.g., natural images leading to a wrong prediction by the network. This observed flaw in DNNs has opened an active research area aimed at improving the robustness of DNNs against these malicious attacks. Of the many methods proposed to improve the robustness of DNNs against adversarial attacks, adversarial training (AT) [2, 3], which requires the introduction of adversarial examples in the training of robust models, has been found effective. The success achieved by AT has led to an array of AT variants, e.g. [4, 5, 6]. Additionally, efforts [7, 1] have been made to improve the performance of AT. One idea for improving the performance of AT is augmenting AT with hypersphere embedding (HE) [1]. HE involves enforcing discriminative constraints on a hypersphere manifold. This is done by normalizing the linear layer's weight and the penultimate layer's features, and using an additive angular penalty.

While the traditional softmax cross-entropy loss has the disadvantage of not explicitly encouraging discriminative learning of features [8, 9], various angular softmax-cross-entropy loss functions that incorporate HE have been proposed to address this limitation. Notable implementations in deep learning literature that utilize HE include CosFace[9], ArcFace [10], and SphereFace [11], among others.

A major feature of HE is that it encourages learning angularly discriminative features [9, 10, 11]. While Pang et al. [1] have demonstrated that incorporating HE improves the performance of the AT methods by directly applying the HE described in CosFace [9] into existing adversarial training methods, e.g., standard AT [3], ALP [4], and TRADES[6], we observe that these existing AT variants were not originally designed for the HE framework and were not taking advantage of the abundant angular information.

In this paper, we propose a new training objective that integrates HE into AT and exploits the rich angular information in the HE framework. Our work is the first to address adversarial robustness exclusively using angular information. Our proposed Angular-AT objective consists of an angular softmax-cross-entropy loss plus two regularization terms that: (1) encourage weight-feature compactness by explicitly minimizing the angle between an adversarial feature vector and the weight vector corresponding to the true class, (2) encourage inter-class separation by maximizing the angles among class weight vectors. Lastly, we perform extensive experimental studies to show the effectiveness of the proposed method.

2. RELATED WORKS

2.1. Hypersphere Embedding

The standard softmax cross-entropy loss has been argued to lack sufficient feature discriminative power, especially when deployed in deep face recognition models [11]. To address this limitation, variants of HE, which improve intra-class compactness and inter-class variance, were introduced [9, 10, 8]. From a geometrical perspective, HE imposes discriminative constraints on a hypersphere manifold and improves the learning of angular discriminative features.

Popular HE implementations include CosFace[9], ArcFace [10], and SphereFace [11].

2.2. Adversarial Robustness

Many methods have been proposed to defend against adversarial examples [2, 3, 6, 12, 13, 4, 5], among which *AT* [2, 3] is considered the most effective. In their seminal work, Madry et al. [3] formulated adversarial training as a min-max optimization problem as follows:

$$\min_{\boldsymbol{\omega}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \left[\max_{B_\epsilon(\mathbf{x})} L_{\boldsymbol{\omega}}(\mathbf{x}+\delta, y) \right] \tag{1}$$

where $L_\omega(.)$ is the loss function, ω are the model parameters, y is the label of $\mathbf{x}$, and δ denotes an adversarial perturbation constrained by the radius ϵ. The inner maximization typically utilizes the Projected Gradient Descent (PGD) attack to craft adversarial examples. The outer minimization minimizes the high loss induced by the adversarial examples.

Prominent variants of AT include [6, 5, 4], to cite a few. Various ideas such as [7, 14, 1] have also been explored to boost the performance of AT. Pang et al. [1] have incorporated HE into adversarial training to boost adversarial robustness.

Our work follows [1] in integrating HE into AT. While [1] directly introduces HE into existing AT variants such as PGD-AT [3], TRADES [6], and ALP[4], we introduce a novel adversarial training objective that exploits the rich angular information available on the hypersphere manifold.

3. NOTATION AND PRELIMINARIES

We denote the training set as $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^d$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^n$ represents a feature vector, n is the dimension of the feature vectors, $y_i \in \{1, \cdots, K\}$ where K represents the number of labels, and $d = |\mathcal{D}|$. For a feature vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_p = (\sum_{i=1}^n |\mathbf{x}_i|^p)^{\frac{1}{p}}$. We define the ϵ–neighborhood of $\mathbf{x}$ as $B_\epsilon(\mathbf{x}) : \{\mathbf{x}' \in \mathcal{X} : \|\mathbf{x}' - \mathbf{x}\|_p \leq \epsilon\}$. Let $f_\omega : \mathbb{R}^n \to \mathbb{R}^K$ denote a deep learning classifier with model parameters ω that produce the output

$$f_\omega(\mathbf{x}) = \mathbb{S}(\mathbf{W}^T\mathbf{z} + b) \tag{2}$$

where $\mathbf{z} = \mathbf{z}(\mathbf{x}; \omega)$ denotes the extracted features in the penultimate layer with the model parameters ω, the matrix $\mathbf{W} = (\mathbf{W}_1, ..., \mathbf{W}_K)$ is the weight matrix, and b is the bias term in the linear layer. $\mathbb{S}(.)$: $\mathbb{R}^K \to \mathbb{R}^K$ is the softmax function.

3.1. Hypersphere Embedding

HE typically involves four operations: setting the bias term in eq. (2) to zero, weight normalization (WN), feature normalization (FN), and setting an additive angular margin (AM). In eq. (2), when $b = 0$, $\mathbf{W}^T\mathbf{z} + b$ becomes $\mathbf{W}^T\mathbf{z} = (\mathbf{W}_1^T\mathbf{z}, ..., \mathbf{W}_K^T\mathbf{z})$. The inner product $\mathbf{W}_k^T\mathbf{z} = \|\mathbf{W}_k\|\|\mathbf{z}\|\cos\boldsymbol{\theta}_k$, where $\boldsymbol{\theta}_k$ is the angle between $\mathbf{z}$ and $\mathbf{W}_k$. The WN and FN operations are computed as follows:

$$\text{WN: } \widetilde{\mathbf{W}_k} = \frac{\mathbf{W}_k}{\|\mathbf{W}_k\|}, \quad \text{FN: } \widetilde{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|}. \tag{3}$$

After applying the FN and WN operations, we have that $\widetilde{\mathbf{W}}^T\widetilde{\mathbf{z}} = (\widetilde{\mathbf{W}}_1^T\widetilde{\mathbf{z}}, ..., \widetilde{\mathbf{W}}_K^T\widetilde{\mathbf{z}}) = (\cos\boldsymbol{\theta}_1, ..., \cos\boldsymbol{\theta}_K)$. $\cos\boldsymbol{\theta}_k$ represents the logit for the class k and $\boldsymbol{\theta}_k$ is the angle between feature $\mathbf{z}$ and class weight $\mathbf{W}_k$. Let $\cos\boldsymbol{\theta}$ denote $(\cos\boldsymbol{\theta}_1, ..., \cos\boldsymbol{\theta}_K)$. For a neural network with hypersphere embedding, we rewrite eq. (2) as:

$$\widetilde{f}_\omega(\mathbf{x}) = \mathbb{S}(\widetilde{\mathbf{W}}^T\widetilde{\mathbf{z}}) = \mathbb{S}(\cos\boldsymbol{\theta}), \tag{4}$$

where $\widetilde{f}_\omega(\mathbf{x})$ is the output of the neural network with hypersphere embedding, which we shall refer to as HE-DNN from now on. Wang et al [9] proposed to train HE-DNN with the following cross-entropy loss with angular margin:

$$L_{CE}(\widetilde{f}_\omega(\mathbf{x}), y) = -\mathbf{1}_y^T log\, \mathbb{S}(s \cdot (\cos\boldsymbol{\theta} - m \cdot \mathbf{1}_y)), \tag{5}$$

where the hyperparameter $s > 0$ is a scaling parameter for improving numerical stability during training [15, 1], m is the angular margin.

4. PROPOSED METHOD

It can be observed from eq. (4) that the output logits and the resulting posterior probabilities of a HE-DNN classifier depend on the angles between the normalized weight vectors of the linear layer and the normalized feature vector in the penultimate layer. Similarly, it can be argued that an adversarial attack crafted on HE-DNN attacks these angles.

Given an example x with label y, the goal of an adversarial attack is to craft adversarial example x' from x that fools the classifier to classify x' as y' such that $y' \neq y$. Consider a binary HE-DNN classifier with a single output such that the cross-entropy loss aims to maximize $\widetilde{\mathbf{W}}^T\widetilde{\mathbf{z}} = \cos\theta$ on input x with label $y = 2$. If x is correctly classified, then $\cos\theta > 0$. However, the adversarial goal of crafting x' becomes making $\cos\theta < 0$, thereby attacking angle θ between the normalized feature vector $\widetilde{\mathbf{z}}$ and the weight vector $\widetilde{\mathbf{W}}$.

Given that the angles between the feature vector and weight vectors contain abundant discriminative information [10, 16, 17] and adversarial attacks attack these angles, we propose a regularization term that directly encourages the weight-feature compactness, more specifically, by minimizing the angle between adversarial feature vector and the weight vector corresponding to the ground-truth label y. In addition, prior works [18] have argued strong connections between adversarial robustness and inter-class separability. We therefore propose an additional angular-based regularization term that improves the inter-class separability.

4.1. Weight-Feature Compactness

Generating adversarial examples involves minimizing a model's confidence on an input example w.r.t its true class. Thus, in HE-DNN, the output logit (cosine value) of an input corresponding to its true label is degraded by an adversarial attack. The lower cosine value occasioned by an adversarial attack corresponds to a larger angle between the feature embedding of the adversarial input and the weight vector of the true label, and consequently a smaller angle between the feature embedding and the weight vector of a wrong class. Hence, there exists a connection between weight-feature angular compactness and robustness. We provide a geometrical illustration of weight-feature angular compactness in Fig. 1.

To improve robustness, we utilize a regularization term that encourages minimization of the angle between the adversarial feature embedding and the weight corresponding to the ground-truth label y. We define the following regularization term to achieve this goal:

$$l_{wfc} = [arccos(\widetilde{\mathbf{W}_y}^T \cdot \widetilde{\mathbf{z}'})]^2 = (\boldsymbol{\theta}'_y)^2 \tag{6}$$

where $\boldsymbol{\theta}'_y$ is the angle between the feature embedding of the adversarial example $\mathbf{x}'$ and weight vector $\boldsymbol{W}_y$ of true class $\boldsymbol{y}$.

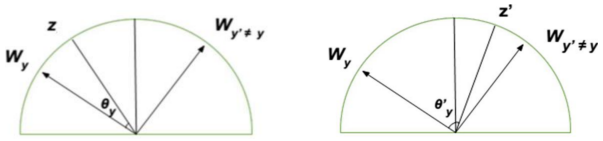

Fig. 1. The figure on the left shows nearness of a feature vector $\boldsymbol{z}$ to the weight vector of the ground truth. The figure on the right depicts how the resulting feature vector of the corresponding adversarial example, $\boldsymbol{z'}$, is pushed towards the weight vector of wrong class $\boldsymbol{y'} \neq \boldsymbol{y}$, thereby increasing the angle between the feature vector of the adversarial example and the weight vector of the true label.

4.2. Inter-Class Separation

Here we consider the inter-class separation in HE-DNN. The weight matrix $\boldsymbol{W}$, in the fully connected layer of a neural network, conceptually represents the various class centers [9, 10]. By penalizing the angles between these class centers on the hypersphere, we aim to improve angular discriminability and robustness of HE-DNN to adversarial attacks. We propose to uniformly distribute the class centers around the unit hypersphere. To achieve this goal, we utilize the popularly known Tammes problem [19, 20].

Tammes problem attempts to uniformly arrange n points on a unit sphere by maximizing the minimum pairwise distance. Using an idea similar to the Tammes problem, we penalize minimal angles between class centers by explicitly maximizing the minimum angles between class centers. Given a linear layer weight matrix $\mathbf{W} = (\mathbf{W}_1, ..., \mathbf{W}_K)$, where $\mathbf{W}_k$ corresponds to the weights for class k, we aim to maximize the angles between each close pair of weight vectors. We minimize the maximum cosine similarity between pairwise weight vectors by the following regularization term:

$$l_{sep} = \frac{1}{K}\sum_{i=1}^{K} max_{j,j\neq i}\mathbf{C}_{i,j} \tag{7}$$

where $\mathbf{C}_{i,j} = \widetilde{\boldsymbol{W}}_i^T \cdot \widetilde{\boldsymbol{W}}_j$ represents pairwise similarities, K is the number of classes, and $\widetilde{\boldsymbol{W}}_i = \frac{\boldsymbol{W}_i}{\|\boldsymbol{W}_i\|}$. Globally computing maximum cosine similarity $max_{i,j,i\neq j}\mathbf{C}_{i,j}$ is inefficient; thus, we compute the mean of each vector's maximum cosine similarity in eq. (7).

Prior work such as [21] adopted the Tammes problem in non-adversarial settings to uniformly separate prototypes which are *a priori* positioned on hyperspherical output. In this work, we uniquely apply Tammes-inspired uniform separation to improve the uniform separation of class identities in the linear layer of HE-DNNs.

4.3. Training Objective

We combine the regularization terms described in sections 4.1 and 4.2 with the loss function decribed in eq. (5) to arrive at the following Angular-AT training objective:

$$\min_{\boldsymbol{\omega}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big\{L(\widetilde{f}_\omega(\mathbf{x}'), y) + \alpha l_{wfc} + \beta l_{sep}\Big\} \tag{8}$$

where α and β are regularization hyperparameters. $\mathbf{x}'$ is an adversarial example crafted by perturbing input example $\mathbf{x}$ using PGD attack. y is the true label of $\mathbf{x}$.

5. EXPERIMENTS

We experimentally verify the effectiveness of the our method, and compare the robustness obtained with the state-of-the-art defenses on *CIFAR10/100* and *TinyImagenet*.

Baselines. We compare our proposed Angular-AT method against the two best performing defenses that are based on HE-DNN: PGD-AT-HE and TRADES-HE [1]. For completeness, we also compare against original defenses PGD-AT [3] and TRADES [6], which are based on the standard DNNs.

Defense settings. We utilize ResNet18 (*CIFAR100* and *TinyImagenet*) and Wideresnet-34-10 (*CIFAR10*) [22] for classification. The models are trained for 100 epochs, using mini-batch gradient descent with momentum 0.9, batch size 128, weight decay 3.5e-3 (ResNet-18) and 5e-4 (Wideresnet-34-10). The learning rates are set to 0.01 and 0.1 on ResNet-18 and Wideresnet-34-10 respectively. In both cases, the learning rates are decayed by a factor of 10 at 75th, and then 90th epoch. The adversarial examples used for training are obtained by perturbing each image using PGD attack, setting

the perturbation $\epsilon = 0.031$, the perturbation step size to 0.007, and the number of iterations to 10.

Hyperparameter settings. For the baselines PGD-AT-HE and TRADES-HE, m and s are respectively kept at 0.2 and 15.0, as described by the authors [1]. We set $m = 0$ and the scale $s = 15.0$. The values of regularization hyper-parameters α and β are heuristically determined, and the values yielding the best results after multiple experiments were selected. Hence, α and β are set to 0.55, 0.48 respectively.

Evaluation settings. We evaluated our defense under *White-box attack* threats including PGD-20/500 [3], CW (L_{inf} version of CW loss optimized by PGD20) [23] and Autoattack [24]. The perturbation size is set to $\epsilon = 0.031$, step size $\eta = 0.003$. In addition, we evaluated the proposed defense on the SPSA[25] attack (a strong query-based *Black-box attack*), with the perturbation size of 0.001 (for gradient estimation), sample size of 128, 80 iterations, and learning rate 0.01. Experiments were reproduced four times with different random seeds; the mean and standard deviation are subsequently computed. The results are reported as mean ± std in the tables. The results, as reported in tables 1-3, shows that our method significantly improves the baselines.

Ablation Studies. We study the contribution of each regularization to the robustness against PGD, CW, and AA attacks, using CIFAR10 on WRN-34-10. Our observations are reported in table 4. Training HE-DNN using only $L(\widetilde{f}_\omega(x'), y)$ yields good performance on PGD attacks, but weaker performance on stronger attacks like CW and AA attacks. $L(\widetilde{f}_\omega(x'), y)$ + l_{wfc} yield significantly better performance on CW and AA attacks, slightly at the expense of PGD attack robustness. The l_{sep} term improves robustness to PGD attacks, therefore l_{sep} compensates for the drop in PGD robustness caused by the l_{wfc} term.

Table 1. Robust accuracy (%) for CIFAR10 on WRN34-10.

DEFENSE	NATURAL	PGD^{20}	PGD^{500}	CW	AA
PGD-AT	86.28	55.96	51.83	54.79	51.80
PGD-AT-HE	86.19	59.36	57.49	55.12	51.65
TRADES	84.62	56.48	54.84	54.17	**53.06**
TRADES-HE	82.98	61.34	60.31	54.92	52.11
ANG-AT	**86.52**$\pm$ 0.12	**64.01** $\pm$ 0.17	**63.28**$\pm$ 0.10	**56.25**$\pm$ 0.17	52.69$\pm$ 0.14

Table 2. Robust accuracy (%) for CIFAR100 on RN18.

DEFENSE	NATURAL	PGD^{20}	PGD^{500}	CW	AA
PGD-AT	56.02	27.72	25.35	25.52	23.89
PGD-AT-HE	55.29	30.63	27.85	26.78	23.55
TRADES	53.60	28.65	25.73	25.48	24.65
TRADES-HE	**56.95**	31.40	28.62	26.96	24.10
ANG-AT	56.89 $\pm$ 0.09	**33.51** $\pm$ 0.18	**31.92** $\pm$ 0.11	**28.52** $\pm$ 0.12	**25.32** $\pm$ 0.09

6. DISCUSSION

In section 4, we argued that adversarial attacks fool the HE-DNN by attacking the angle between features and class weights. Hence, we propose a weight-feature compactness regularization term to force a minimal angle between the feature embedding of an adversarial example and the true class weight. Experimental results show that the l_{wfc} term improves robustness against stronger attacks like CW and AA, at the expense of PGD attack.

We note in section 4, the significance of encouraging inter-class separation. Input samples corresponding to classes that are semantically closer to other classes are easily misclassified by adversarial attacks. We propose a regularization term that ensures inter-class separation. The angular margin penalty used in [1] ensures inter-class separation between the true class and the closest incorrect class. In contrast, our approach encourages uniform inter-class separation among all class-centers on the hypersphere. Our experiments show that l_{sep} improves robustness against PGD attacks.

In general, incorporating hypersphere embedding into adversarial training significantly improves the robustness against adversarial attacks, especially PGD.

Table 3. Robust accuracy (%) for TinyImageNet on RN18.

DEFENSE	NATURAL	PGD^{20}	CW
PGD-AT	49.20	22.27	20.54
PGD-AT-HE	48.91	24.72	21.37
TRADES	48.17	21.17	17.43
TRADES-HE	49.05	25.91	20.32
ANG-AT	**50.21** $\pm$ **0.09**	**27.27** $\pm$ 0.18	**22.29** $\pm$ 0.09

Table 4. Ablation study acc%: WRN-34-10 on CIFAR-10.

LOSS TERMS	PGD^{500}	CW	AA
$L(\widetilde{f}_\omega(x'), y)$	62.34$\pm$ 0.22	53.91$\pm$ 0.34	50.97 $\pm$ 0.28
$L(\widetilde{f}_\omega(x'), y) + \alpha l_{wfc}$	61.52$\pm$ 0.17	**56.85**$\pm$ 0.14	**53.10**$\pm$ 0.19
$L(\widetilde{f}_\omega(x'), y) + \beta l_{sep}$	**63.37** $\pm$ 0.08	55.06$\pm$ 0.12	51.15 $\pm$ 0.19
$L(\widetilde{f}_\omega(x'), y) + \alpha l_{wfc} + \beta l_{sep}$	63.28$\pm$ 0.12	56.35$\pm$ 0.27	52.69 $\pm$ 0.24

Table 5. Robust accuracy on SPSA attack: WRN-34-10 on CIFAR-10

PGD-HE	TRADES-HE	**ANG-AT**
64.09$\pm$ 0.08	63.74$\pm$ 0.12	64.32$\pm$ 0.11

7. CONCLUSION

In this paper, we improved on an existing work in [1], which incorporates hypersphere embedding into existing adversarial training methods. We note that the existing adversarial training methods were not originally designed for the HE framework and were not taking advantage of the angular information available in the HE framework. Consequently, we intuitively justify and propose two regularization terms that exploit the rich angular information available in the HE framework to improve adversarial training. Subsequently, we extensively tested our method on various strong attacks. The experimental results show the effectiveness of our method in improving the state-of-the-art.

8. REFERENCES

[1] Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Jun Zhu, and Hang Su, "Boosting adversarial training with hypersphere embedding," *Neurips*, vol. 33, pp. 7779–7792, 2020.

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2015.

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[4] Harini Kannan, Alexey Kurakin, and Ian Goodfellow, "Adversarial logit pairing," *arXiv preprint arXiv:1803.06373*, 2018.

[5] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *ICLR*, 2019.

[6] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan, "Theoretically principled trade-off between robustness and accuracy," in *ICML*. PMLR, 2019, pp. 7472–7482.

[7] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang, "Unlabeled data improves adversarial robustness," *Neurips*, vol. 32, 2019.

[8] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang, "Large-margin softmax loss for convolutional neural networks," in *ICLR*. PMLR, 2016, pp. 507–516.

[9] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265–5274.

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019, pp. 4690–4699.

[11] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017, pp. 212–220.

[12] Hang Wang, David J Miller, and George Kesidis, "Anomaly detection of adversarial examples using class-conditional generative adversarial networks," *Computers & Security*, vol. 124, pp. 102956, 2023.

[13] Nicholas Carlini, Florian Tramer, J Zico Kolter, et al., "(certified!!) adversarial robustness for free!," *arXiv preprint arXiv:2206.10550*, 2022.

[14] Dongxian Wu, Shu-Tao Xia, and Yisen Wang, "Adversarial weight perturbation helps robust generalization," *Nerips*, vol. 33, 2020.

[15] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille, "Normface: L2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1041–1049.

[16] Zhennan Wang, Wenbin Zou, and Chen Xu, "Pr product: A substitute for inner product in neural networks," in *ICCV*, 2019, pp. 6013–6022.

[17] Zhennan Wang, Canqun Xiang, Wenbin Zou, and Chen Xu, "Dma regularization: Enhancing discriminability of neural networks by decreasing the minimal angle," *IEEE Signal Processing Letters*, vol. 27, pp. 2089–2093, 2020.

[18] Xin Li, Xiangrui Li, Deng Pan, and Dongxiao Zhu, "Improving adversarial robustness via probabilistically compact loss with logit constraints," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 8482–8490.

[19] Pieter Merkus Lambertus Tammes, "On the origin of number and arrangement of the places of exit on the surface of pollen-grains," *Recueil des travaux botaniques néerlandais*, vol. 27, no. 1, pp. 1–84, 1930.

[20] L Lovisolo and EAB Da Silva, "Uniform distribution of points on a hyper-sphere with applications to vector bitplane encoding," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 148, no. 3, pp. 187–193, 2001.

[21] Pascal Mettes, Elise van der Pol, and Cees Snoek, "Hyperspherical prototype networks," *Neurips*, 2019.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[23] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[24] Francesco Croce and Matthias Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *ICLR*. PMLR, 2020, pp. 2206–2216.

[25] Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord, "Adversarial risk and the dangers of evaluating against weak attacks," in *ICLR*. PMLR, 2018, pp. 5025–5034.