Adversarial Robustness for Face Recognition: How to Introduce Ensemble Diversity among Feature Extractors?

Takuma Amada, [1] **Kazuya Kakizaki,** [1] **Toshinori Araki,** [1]
Seng Pei Liew, [1] * **Joseph Keshet,** [2] **Jun Furukawa** [3]
[1] NEC Corporation, 7-1, Shiba, 5-chome Minato-ku, Tokyo 108-8001 Japan
[2] Bar-Ilan University, Ramat Gan, 52900, Israel
[3] NEC Israel Research Center, 2 Maskit Street, Herzliya Pituach, Israel
{t-amada, kazuya1210, toshinori_araki, jun.furukawa1971}@nec.com,
sengpei.liew@gmail.com, jkeshet@cs.biu.ac.il, jun.furukawa@necam.com

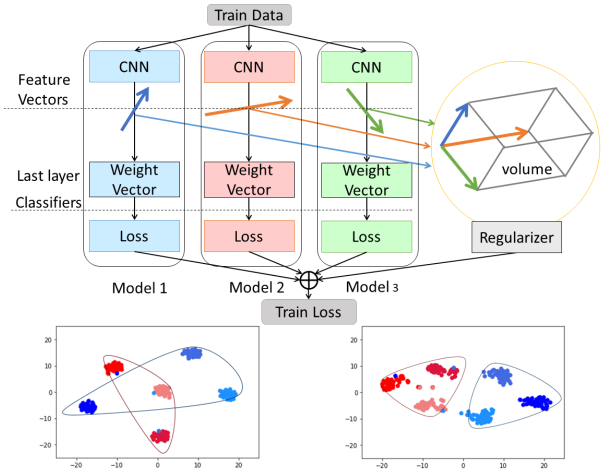
Abstract

An adversarial example (AX) is a maliciously crafted input that humans can recognize correctly, while machine learning models cannot. This paper considers how to turn deep learning-based face recognition systems to be robust against AXs. A large number of studies have proposed methods for protecting machine learning-classifiers from AXs. One of the most successful methods among them is to prepare an ensemble of classifiers and promote diversity among them. Face recognition typically relies on feature extractors instead of classifiers. We found that directly applying this successful method to feature extractors leads to failure. We show that this failure is due to a lack of *true* diversity among the feature extractors and fix it by synchronizing the direction of features among models. Our method significantly enhances the robustness against AXs under the white box and black box settings while slightly increasing the accuracy. We also compared our method with adversarial training.

Introduction

Deep neural networks (DNNs) have become core components of many essential services as their performance has gone beyond the human capability of recognition in many tasks (Parkhi, Vedaldi, and Zisserman 2015; Schroff, Kalenichenko, and Philbin 2015; Szegedy et al. 2015; He et al. 2016). Face recognition is one of the most widely used services that rely on DNNs (Sun et al. 2014), ranging from immigration inspection to smartphone authentication. However, the vulnerability of deep learning under adversarial examples has begun to threaten its promises (Szegedy et al. 2014). (Singh et al. 2020) discuss a wide variety of attacks.

An adversarial example (AX) is an inconceivably perturbed input that deceives the machine learning model into miss-classification of it, while a human can correctly classify it. Some attacking methods need an entire code of the model, a white-box setting, while some need only oracle access to the model, a black-box setting. The threat became very plausible when Sharif et al. (2016) showed physical glasses could deceive the machine learning model in the black-box setting. As a result of such an attack, a person on the blacklist may wear a slightly fancy glass to evade immigration inspection performed by machines.



t-SNE visualization: weight vectors of the last layer in models 1, 2, and 3 are independent.

t-SNE visualization: weight vectors of the last layer in models 1, 2, and 3 are the same

Figure 1: We promote the diversity of feature vectors by promoting a larger volume spanned by them. We share the same weight vectors of the last layer by all models, making the diversity concerning the same weight. Two t-SNE visualizations of features show how they are affected by the synchronizing of the weight vector. Here, the same classes are of the same color.

Many works proposed how to prevent AXs, and many of them are broken back, leading to an arms race between defenders and attackers. A model ensembling (Abbasi and Gagné 2017; Dabouei et al. 2020; Kariyappa and Qureshi 2019) is one of the most successful prevention methods among them. In particular, the adaptive diversity promoting (ADP) method (Pang et al. 2019) that promotes the diversity of models in the ensemble is the most successful. Although its defense is immature like all others when attacked adaptively (Tramèr et al. 2020), such a strategy has an ad-

[*] Currently at LINE Corporation, Japan.

vantage. It is orthogonal to other defensive approaches that focus on enhancing single-model adversarial robustness and can be used in tandem to achieve further adversarial robustness. Adversarial training (Zhong and Deng 2019), another successful AX prevention method, is an example of such a tandem method.

We are interested in applying the ADP method to enhance the robustness of face recognition against AXs. Face recognition commonly relies on a machine learning feature extractor since it enables the service to register a huge number of new faces without retraining the network. We experimented with ADP's direct application and found that it does not improve feature extractors' robustness against AXs.

We consider the cause of failure is that the directions of different models' features are not comparable in a meaningful way, and thus their diversity is insignificant. We propose letting all the models in the ensemble share weight vectors of their final layers so that features in different models can compare themselves in a common coordinate. We also propose to promote the diversity of feature vectors directly rather than the diversity of the classifier. Figure 1 illustrates our method.

We have experimented with ADP, our methods, and several variants. We trained them in various methods such as ArcFace (Deng et al. 2019) and CosFace (Wang et al. 2018), with the MS1MV2 dataset (Deng et al. 2019), the refined version of the MS-Celeb-1M, and verified by VGG2 (Cao et al. 2018). We measured the robustness by AXs adaptively generated by I-FGSM, an iterative variant of Fast Gradient Signed Method (FGSM) (Goodfellow, Shlens, and Szegedy 2015), Basic Iterative Method (BIM) (Kurakin, Goodfellow, and Bengio 2017a; Madry et al. 2018), and Carlini & Wagner (CW) attack (Carlini and Wagner 2017b). We confirmed that ADP affects neither accuracy nor robustness against AXs. On the other hand, our method significantly enhanced the robustness to AXs in both white-box and black-box settings without harming its accuracy at all.

Despite its enhancement, our model's robustness was not so distinctly high that we can still generate successful AXs for all legitimate samples in the white-box setting if with sufficiently large perturbation. Although our method does show significantly lower transferability than others in a black-box setting, we found that very well forged AXs often, with sufficiently large perturbation, can again deceive the machine recognition with very high probability.

Preliminaries

Face Recognition by DNN Feature Extractor

Face recognition typically belongs to either face identification or face verification. The former is also called closed-set face classification and assumes the probed image belongs to one of the objects enrolled in the gallery. The latter is also called open-set face classification and can reject a probed image with no corresponding object in the gallery. We focus on open-set face recognition.

Modern open-set face recognition systems commonly consist of a DNN-based feature extractor that maps an input image into a low dimensional feature space (Sun et al. 2014). Then we can measure the similarity of the two images by the distance between their two corresponding feature vectors. The commonly used distances are the Euclidean distance and the cosine distance. A feature extractor excels in face recognition service as it requires no retraining, unlike classifiers when it registers a new face. [1]

There are two major ways of training the DNN-based feature extractors. One way is to train a normal multiclass DNN classifier initially and then regard the output of the penultimate layer of the DNN as the feature vector (i.e., the DNN without the last fully connected layer is a feature extractor) (Parkhi, Vedaldi, and Zisserman 2015; Sun et al. 2014). Another approach is to train the feature extractor directly using the triplet loss (this is also called metric learning) (Schroff, Kalenichenko, and Philbin 2015). A triplet consists of two matching face thumbnails and a nonmatching face thumbnail, and the loss aims to separate the positive pair from the negative by a distance margin. As the number of triplet combinations increases exponentially, triplet loss tends to be harder to scale. We focus on the former approach in this work.

Feature extractors trained through simple softmax outputs are effective for closed-set classification tasks but are not discriminative enough for open-set verification. An angular margin penalty such as ArcFace (Deng et al. 2019), CosFace (Wang et al. 2018), and Sphereface (Liu et al. 2017), and other approaches such as (Wan et al. 2018) is a form of the loss function that has successfully improved the discriminative power of face verification by feature vectors. The angular margin penalty modifies the final prediction layer to enforce the true label's predictions to be more restrictive and discriminative than the vanilla softmax by penalty. In such a way, the loss transfers the penalty against the prediction into the distance between features, features with high inter-class variance and low intra-class variance.

We briefly review the angular margin penalty. We start from the cross-entropy loss $\mathcal{L}_{CE}(x, y)$ for a trainable parameter ϕ, an input x, and its true label y. We assume $g(x, \phi)$ is the classification over n classes and is the fully connected final layer's softmax output whose input is the output $f(x, \phi) \in \mathbb{R}^d$ of the d-dimensional penultimate layer. Then, letting $W_j \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$ for $j = 1, \ldots, n$, respectively, an final weight vector and a final bias for the j-th prediction,

$$\mathcal{L}_{CE}(x, y) := {}^T 1_y \cdot \log g(x) = \log \frac{e^{W_y \cdot f(x) + b_y}}{\sum_{\ell=1}^{n} e^{W_\ell \cdot f(x) + b_\ell}}.$$

The loss function of ArcFace $\mathcal{L}_{ARC,\sigma,\mu}(x, y)$ L_2-normalizes f and W_j, lets $b_\ell = 0$, and introduces penalty μ and smoothness hyperparameter σ. With $\cos\theta(x, \ell) = \frac{W_\ell \cdot f(x)}{\|W_\ell\| \cdot \|f(x)\|}$, it is;

$$\begin{aligned} & \mathcal{L}_{ARC,\sigma,\mu}(x, y) \\ = \quad & \log \frac{e^{\sigma \cos(\theta(x,y)+\mu)}}{e^{\sigma \cos(\theta(x,y)+\mu)} + \sum_{\ell \in \{1,\ldots,n\} \setminus y} e^{\sigma \cos\theta(x,\ell)}}. \end{aligned}$$

[1] Because of this property, we need to evaluate the accuracy of feature extractor based face recognition by datasets that do not share the labels (identities of the owners of faces) with the training dataset.

The loss function of CosFace is with a slightly different form of the penalty. We call each model of both ArcFace and CosFace as a *single model* and compare it with other models in our experiments. The feature extractor trained by this network is $\tilde{f}(\cdot, \phi) \in \mathbb{R}^d$, which is the normalization of $f(\cdot, \phi)$.

Adversarial Examples

While deep neural networks are successful to show high accuracy in their tasks, they are vulnerable to AXs (Goodfellow, Shlens, and Szegedy 2015; Carlini and Wagner 2017b). An AX is a crafted input x_{adv}, which is different from a source image x_s by δ, i.e., $x_{adv} = x_s + \delta$, so that the target classifier misclassifies it but not by the humans. Many works search for a small δ that humans cannot perceive, while some works such as (Kakizaki and Yoshida 2020) search for a large δ that humans consider natural. I-FGSM, an iterative variant of the Fast Gradient Signed Method (FGSM) (Goodfellow, Shlens, and Szegedy 2015; Madry et al. 2018), searches δ by moving in the negative gradient direction to the target label. Basic Iterative Method (BIM) (Kurakin, Goodfellow, and Bengio 2017a) is an extension of I-FGSM where the search is within a given boundary. Carlini & Wagner (CW) attack (Carlini and Wagner 2017b) searches the best δ by formalizing the problem as optimization. The above I-FGSM, BIM, and CW methods are the major strong attacks available today to generate AXs.

(Rozsa, Günther, and Boult 2017) proposed a general method, called LOTS, to generate AXs such that an internal layer representation is close to that of a target by iteratively adding perturbations to the source input. It uses a Euclidean loss defined on the internal layer representations of the origin and the target. It applies its gradient to the source input to manipulate the source input's internal layer representation. We can generate an AX for feature extractors by applying LOTS to the features layer, which is also an internal representation of the classifier that is to be a feature extractor. LOTS is sufficiently general to employ I-FGSM, BIM, and CW for the underlying perturbation method.

Defenses against Adversarial Examples

Various studies have proposed methods to make DNNs robust against AXs. They include adversarial training (Goodfellow, Shlens, and Szegedy 2015; Madry et al. 2018), feature transformation (Xu, Evans, and Qi 2018), statistical analysis (Zheng and Hong 2018), manifold learning (Samangouei, Kabkab, and Chellappa 2018), knowledge distillation (Papernot et al. 2016), selective dropout (Goswami et al. 2018, 2019), an ensemble of models, and more. Despite their partial success, none of them completely prevents AXs from deceiving DNNs with cleverer attack techniques such as (Carlini and Wagner 2017a; Athalye, Carlini, and Wagner 2018). An adversarial training, a relatively successful defense, trains DNNs by generating AXs and including them in the training data (Goodfellow, Shlens, and Szegedy 2015; Kurakin, Goodfellow, and Bengio 2017b). However, it can not sufficiently defend the model against some of the AXs that have not appeared during the training (Gilmer et al. 2019). There have also been studies on certified defenses, of which the aim is to train DNNs in a provably robust fashion (Cohen, Rosenfeld, and Kolter 2019; Hein and Andriushchenko 2017; Wong et al. 2018; Raghunathan, Steinhardt, and Liang 2018; Wong and Kolter 2018). However, this approach's successes are largely limited to simple DNN architectures and datasets with low resolution.

Some works such as (Russakovsky et al. 2015) showed that an ensemble of different models improves the generalizability in image classification tasks. It then turns out that the ensemble model is also successful defenses against AXs. (Pang et al. 2019) proposed adaptive diversity promotion (ADP), which trains an ensemble of models so that their non-maximal predictions vary largely. Since non-maximal predictions differ greatly, it is hard to align them to maximal prediction in an AX.

Direct Integration Adaptive Diversity Promoting (ADP) Method into Feature Extractors

Despite the relative success of the ensemble model, all of the works are validated only for classifications. We can directly integrate the adaptive diversity promoting (ADP) method into the angular margin penalty for feature extraction. We assume that our ensemble is composed of K models, and thus all parameters have K duplicates. Each model predicts over n labels in training. Let the output of the k-th model that represents the j-th label in the ensemble be $g_{k,j}$. The ensemble prediction $\mathcal{G}_j$ for the j-th label is the average of all the output of predictions. That is, $\mathcal{G}_j(x) = \frac{1}{K}\sum_{k=1}^{K} g_{k,j}(x)$. Then, Shannon Entropy of the distribution $\{\mathcal{G}_j\}_{j=1,\ldots,n}$ is

$$\mathcal{H}(\mathcal{G}(x)) = -\sum_{j=1}^{N} \mathcal{G}_j(x)\log(\mathcal{G}_j(x)).$$

Let $g_{k,\backslash y} \in \mathbb{R}^{n-1}$ be such an $(n-1)$- dimensional vector that is $g_k \in \mathbb{R}^n$ except for the y-th element, i.e., true prediction. Let $\tilde{g}_{k,\backslash y} \in \mathbb{R}^{n-1}$ be L^2-normalized $g_{k,\backslash y} \in \mathbb{R}^{n-1}$, and let an $(n-1) \times K$ matrix $M(x,y) = (\tilde{g}_{1,\backslash y}(x), \ldots, \tilde{g}_{K,\backslash y}(x)) \in \mathbb{R}^{(n-1)\times K}$. Then, the ensemble diversity of non-maximal (y) prediction of x is

$$\mathbb{ED}(x,y) = \det({}^T M(x,y) \cdot M(x,y)).$$

Here, the operation of ”$\cdot$” is the multiplication of $K \times (n-1)$ matrix and $(n-1) \times K$ matrix, whose result is $K \times K$ matrix. Geometrically, it is the volume of spaces that $\{\tilde{g}_{1,\backslash y}(x), \ldots, \tilde{g}_{K,\backslash y}(x)\}$ spans. With hyperparameters α and β, the regularizer for promoting adaptive diversity is

$$\mathsf{ADP}_{\alpha,\beta}(x,y) = \alpha \cdot \mathcal{H}(\mathcal{G}(x)) + \beta \cdot \log(\mathbb{ED}(x,y)).$$

With $\{\mathcal{L}^k_{M,\sigma,\mu}(x,y)\}_{k=1,\ldots,K}$ where M represents either ArcFace or CosFace by ARC or COS, the ADP method trains the model by optimizing parameters for minimum

$$\mathcal{L}_{E,M,ADP,\sigma,\mu,\alpha,\beta} := \sum_{k=1}^{K} \mathcal{L}^k_{M,\sigma,\mu}(x,y) - \mathsf{ADP}_{\alpha,\beta}(x,y).$$

The feature extractor is the normalized penultimate layer output $(\tilde{f}_k \in \mathbb{R}^d)_{k=1,\ldots,K}$. We let the ensemble features $\mathcal{F}$

be the average of all the output of features extractors as

$$\mathcal{F}(x) = \frac{1}{K}\sum_{k=1}^{N} \tilde{f}_k(x).$$

Here, we omitted ϕ. We use the Euclidean distance between the above ensemble feature $\mathcal{F}(x)$ of the input x and the registered feature $\mathcal{F}(x')$ of another input x' as a measure of similarity for our verification task.

Our Approach

Problem and Challenge

We will later see in our experiments that ADP scarcely enhances the robustness of the network against AXs. We see its cause as below. Let normalized weight vectors of the final layer by all the models be $\{\tilde{W}_\ell^k\}_{(k,\ell)\in\{1,\ldots,K\}\times\{1,\ldots,n\}}$. Here, k runs for models in the ensemble, and ℓ runs for labels. Let $\{\tilde{f}_k\}_k$ be the feature extractors. Again, k runs for models in the ensemble. The diversity promotion in the ensemble aims to enforce any perturbation δ to input x brings it to such x' that features $\{\tilde{f}_k(x')\}_k$ of different models are close to $\tilde{W}_\ell^k$ of different ℓ's. The ADP is likely to achieve such a property. However, the difference in directions of features does not imply the difference of directions from learned features, i.e., the weight vectors. For example, suppose that $\tilde{f}_1(x)$ near $\tilde{W}_1^1$ moves to a very different direction from the direction of which $\tilde{f}_2(x)$ near $\tilde{W}_1^2$ moves when we perturb x by δ. However, if these directions are such that $\tilde{f}_1(x)$ moves to $\tilde{W}_2^1$ and that $\tilde{f}_2(x)$ moves to $\tilde{W}_2^2$, the difference in directions does not prevent AXs. Since weight vectors are independent among different models, the previous approaches do not prevent such situations from occurring. Therefore, we need to promote the diversity of features among all the ensemble models to be diverse relative to respective weight vectors' positions.

Our Solution

Shared Representative Vector: We propose to share weight vectors $\{\tilde{W}_\ell^k\}_{(k,\ell)\in\{1,\ldots,K\}\times\{1,\ldots,n\}}$ of the final layer by all the models. We propose $\tilde{W}_\ell^k$ for all k are the same and let $\tilde{W}_\ell$ denote it. The change promotes all the label y features to be close to the same $\tilde{W}_y$ independent of models. Hence, the adversary needs to find the perturbation of x, which moves all features $\{\tilde{f}_k(x)\}_{k=1,\ldots,K}$ from $\tilde{W}_y$ to $\tilde{W}_{y'}$. Suppose the directions of the susceptibility of features to perturbation are different among different models. Then, it is hard for an adversary to find perturbation that moves features in the same direction. If we promote diversity of features, we can expect the chart of features around weight vector are different among models and can expect that the ensemble increases its robustness to AXs.

We call the weight vector $\tilde{W}_\ell$ that all models share as shared representative weight vectors (SRV). Let $\psi_k(x,\ell)$ be the angle, i.e., the arc, between $\tilde{W}_\ell$ and $\tilde{f}_k(x)$ in $\mathbb{R}^d$. That is, $\psi_k(x,\ell)$ is such that,

$$\cos\psi_k(x,\ell) = \tilde{W}_\ell\cdot\tilde{f}_k(x),$$

which we can compare to the original $\theta_k(x,\ell)$ such that

$$\cos\theta_k(x,\ell) = \frac{W_\ell^k\cdot f_k(x)}{\|W_\ell^k\|\cdot\|f_k(x)\|} = \tilde{W}_\ell^k\cdot\tilde{f}_k(x).$$

We have the loss function with the shared representative weight vector as

$$\begin{aligned}&\mathcal{L}_{E,ARC,SRV,\sigma,\mu}(x,y)\\ =\ &\sum_{k=1}^{K}\log\frac{e^{\sigma\cos(\psi_k(x,y)+\mu)}}{e^{\sigma\cos(\psi_k(x,y)+\mu)}+\sum_{\ell\in\{1,\ldots,n\}\setminus y}e^{\sigma\cos\psi_k(x,\ell)}}.\end{aligned}$$

We expect that feature extractors trained by the following loss function are robust to AXs.

$$\begin{aligned}&\mathcal{L}_{E,ARC,SRV,FDP,\sigma,\mu,\gamma}(x,y)\\ =\ &\mathcal{L}_{E,ARC,SRV,\sigma,\mu}(x,y)-\mathsf{FDP}_\gamma(x).\end{aligned}$$

We define $\mathcal{L}_{E,COS,SRV,FDP,\sigma,\mu,\gamma}(x,y)$ similarly.

Feature Diversity Promotion: In ADP, we promote nonmaximal predictions of models in the ensemble to be diverse. However, it is a feature that we want it hard for adversaries to manipulate rather than predictions. Hence, we choose to promote the diversity of ensemble features directly. We can measure the diversity $\mathbb{ED}_{feat}$ of ensemble features at x in the same manner as before. Let $\tilde{F}(x) = (\tilde{f}_1(x),\ldots,\tilde{f}_K(x))\in\mathbb{R}^{d\times K}$ be $d\times K$ matrix. Then,

$$\mathbb{ED}_{feat}(x) = \det({}^T\tilde{F}(x)\cdot\tilde{F}(x))$$

Here, the determinant is on the $K\times K$ matrix. We promote the ensemble feature extractors to be such that their features are diverse by the following regularizer.

$$\mathsf{FDP}_\gamma(x) = \gamma\log(\mathbb{ED}_{feat}(x))$$

The weighting coefficient γ is a hyperparameter. The regularizer has no term of Shannon entropy, unlike $\mathsf{ADP}_{\alpha,\beta}$, since we do not need to balance features with values such as maximal prediction in ADP.

Experiments

Implementation Details

We followed the same training process that is in (Deng et al. 2019). We adopt an MS1MV2 dataset (Deng et al. 2019), the refined version of the MS-Celeb-1M dataset, for the training, and VGG2 for the verification. The training dataset in the MS1MV2 dataset includes 5.8M face images and 85K identities. For data preprocessing, we crop face images to the size of 112×112 and align face images by utilizing MTCNN (Zhang et al. 2016).

For the embedding network, we employ the widely used CNN architecture, MobileFacenet (Chen et al. 2018). After the last convolutional layer, we explore the BN (Ioffe and Szegedy 2015)-Dropout (Srivastava et al. 2014)-FC-BN structure to get the final 512-dimensional embedding feature.

We follow (Wang et al. 2018) to set the feature to scale $\sigma = 64$ and choose the angular margin of ArcFace at $\mu = 0.5$. We choose the angular margin of CosFace at $\mu = 0.35$.

Method	ROC-AUC	
	ArcFace	CosFace
Single model	0.96326	0.96034
ADP	0.94269	0.95964
AdvT	**0.96437**	-
Our method	0.96417	**0.97093**

Table 1: ROC-AUCs of face verification by VGG2 among single model, ADP, and our method shows that our method does not sacrifice accuracy at all.

We set the batch size to 256 and train the ensemble consisting of three feature extractors on one NVIDIA Tesla V100 (32GB) GPU. We set the initial learning rate is 10^{-3}, and we divide it by 10 at 12, 15, and 18 epoch. The training process finishes at 20 epoch.

We have experimented with the effectiveness of regularizers $\mathsf{ADP}_{\alpha,\beta}(x, y)$ and $\mathsf{FDP}_{\gamma}(x)$ to the robustness of feature extractors trained by the ensemble model of ArcFace and CosFace. The hyperparameter of regularizers we tested are $(\alpha, \beta) = (2.0, 0.5), (2.0, 10.0)$, and $(2.0, 50.0)$ for ADP, and $\gamma = 1.0, 10.0$, and 50.0 for FDP. We also compared our method with one of the best adversarial training, which exploits margin-based triplet embedding regularization (Zhong and Deng 2019). They have not experimented with MobileFacenet that we adopted. They also observed that the robustness varies depending on the margin in the triplet embedding regularization term. Hence, we tried several hyperparameter values of $m = 0.2, 0.6, 1.4$, and 3.0 to find the best one for MobileFacenet.

We applied attacks such as I-FGSM, BIM, and CW to the following models in the LOTS framework. (1) ”single model,” which is the original model, (2) ”baseline,” which is a simple ensemble of original models, (3) ”ADP,” which is a simple ensemble model with ADP regularizer, and (4) ”FDP” which is a simple ensemble model with FDP regularizer. (5) ”SRV,” which is an ensemble with a shared representative vector, (6) ”SRV+ADP,” which is SRV with ADP regularizer, (7) ”AdvT for $m = 0.2, 0.6, 1.4, 3.0$,” which is adversarial training, and (8) ”Our method,” which is our full model that has SRV with FDP regularizer. The number of models in each ensemble is three ($K = 3$).

All the attacks are adaptive, which means we applied methods to robustified networks. We could compare our method to only those previous methods we experimented with ourselves since their reported results are in different conditions.

Performance on Legitimate Samples

We verified the accuracy of verifications on the VGG2 dataset. Table 1 shows ROC-AUCs of the single model, ADP, AdvT, and our method (SRV+FDP) for ArcFace and CosFace. The hyperparameters resulting from best ROC-AUC are $(\alpha, \beta) = (2.0, 0.5)$, $m = 0.6$, and $\gamma = 10.0$. Our proposed SRV+FDP is not only no worse than the original single model but performs best.

We also show the accuracy of verification on several legitimate datasets of different conditions in Table 2. These datasets are FW (Huang et al. 2007), AgeDB-30 (Moschoglou et al. 2017), and CFP-FP (Sengupta et al. 2016), which provide face data in an unconstrained setting. We can see that our SRV+FDP is remarkably often better slightly than the single model. On the other hand, ADP and SRV+ADP are often worse than the single model. We can use our method in tandem with other methods. The experiments are with ArcFace and CosFace.

Loss	method	LFW	CFP-FP	AgeDB-30
ArcFace	single model	99.30	89.60	94.22
	Baseline	99.12	89.71	94.15
	ADP	98.90	86.20	90.52
	FDP	**99.40**	89.94	94.13
	SRV	99.26	**90.94**	94.93
	SRV+ADP	99.38	86.62	93.98
	AdvT	99.21	90.80	94.38
	Our method	**99.40**	89.97	**95.15**
CosFace	single model	**99.40**	88.29	93.07
	SRV+ADP	99.33	88.33	92.43
	Our method	99.28	**91.14**	**94.97**

Table 2: Accuracies of verifications by different methods with various datasets show our method does not sacrifice accuracy.

Robustness against Adversarial Examples in White-Box Setting

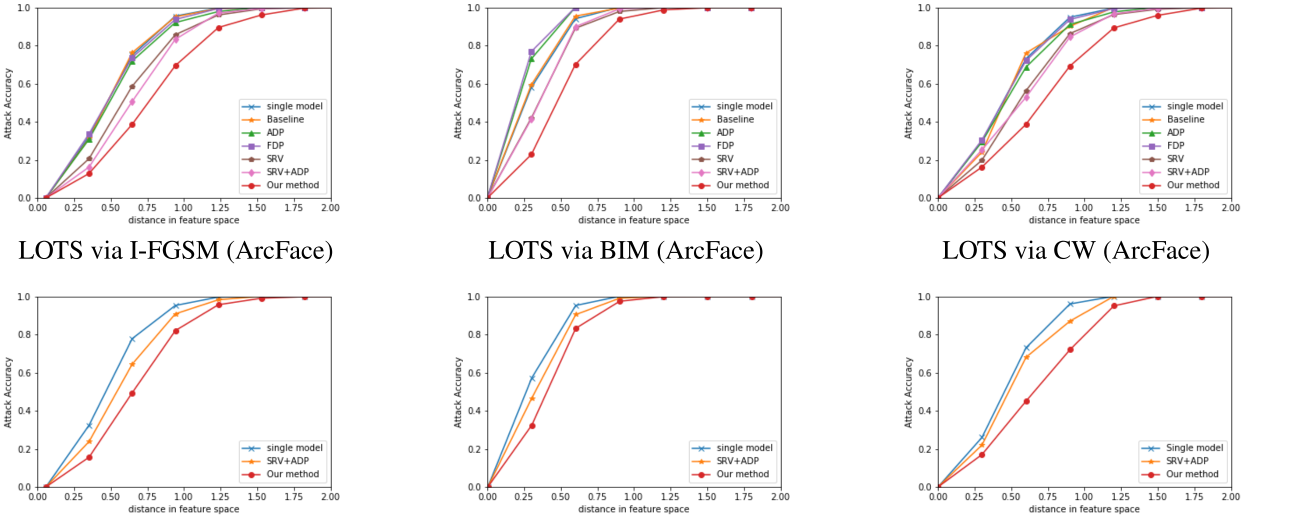
We generate AXs by LOTS with I-FGSM, BIM, and CW in a white-box setting. We did not limit the number of iterations in all attacks, which swells more than 1,000 in some cases. We chose a rather large boundary ($\epsilon = 0.1$) in BIM. We randomly sample 1000 pairs of different identity images from the VGG2 test dataset and generated 500 AXs. As we imposed less on AXs, all the attacks have been successful with their different perturbation sizes. We let the learning rate of underlying SGD to be 0.1. We note that we can still recognize images with the largest perturbation by our human eyes.

We compared the robustness of various methods through the size of perturbation. We define the τ-attack success rate as

$$\tau Acc = \frac{|\{x_{adv}|x_{adv} \in \mathcal{AX}; \|x_{adv} - x_s\|_2 < \tau\}|}{|\mathcal{AX}|}. \tag{1}$$

Here, x_s is a legitimate sample, and x_{adv} is the AX created from x_s. $\mathcal{AX}$ is a set of all the successful AXs generated in the white-box setting. This measurement represents the proportion of adversarial samples whose perturbation size is less than τ to all legitimate images. We can say that the larger perturbation we need to fool feature extractors, the more robust they are.

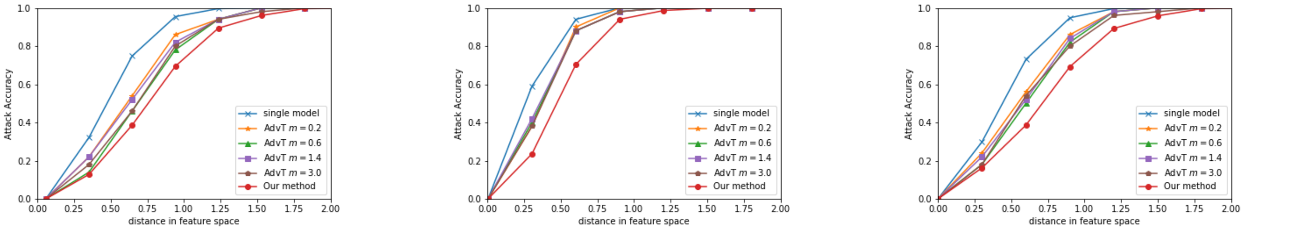
Graphs in the upper row of Figure 2 show the τAcc of LOTS via I-FGSM, BIM, and CW, respectively, for ArcFace. We observed that (1) the baseline, ADP, and the FDP do not enhance the robustness against AXs in a noticeable manner, (2) SRV alone enhances robustness, (3) the addition of ADP to SRV does not enhance robustness, (4) Our



LOTS via I-FGSM (ArcFace) LOTS via BIM (ArcFace) LOTS via CW (ArcFace)

LOTS via I-FGSM (CosFace) LOTS via BIM (CosFace) LOTS via CW (CosFace)

Figure 2: White-Box Attacks: the decrease of attack accuracy measures robustness compared to the single model.



LOTS via I-FGSM (ArcFace) LOTS via BIM (ArcFace) LOTS via CW (ArcFace)

Figure 3: White-Box Attacks: robustness of the single model, our method, and the adversarial training method with various values of hyperparameter "m." Differences of robustness among them are marginal.

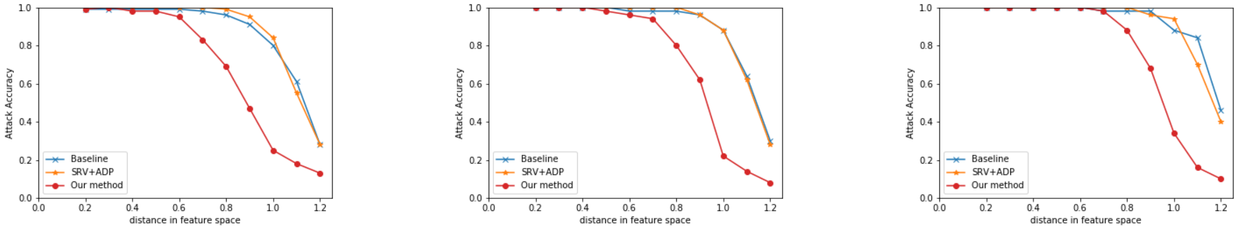
FDP+SRV has the best robustness. We see the same trends for another common loss function of CosFace in graphs in the lower row of Figure 2.

Graphs in Figure 3 compare our method with adversarial training by the τAcc of LOTS via I-FGSM, BIM, and CW, respectively, for ArcFace. The adversarial training is with several values of hyperparameter "m." Differences in robustness among them are marginal. We see our method enjoys the stronger robustness in all the cases.

Robustness against Adversarial Examples in Black-Box Setting

We first generate AXs by LOTS with I-FGSM, BIM, and CW to the single model in a white-box setting. We generated several sets of AXs with different distances between their features and the target features. All of their distances are sufficiently small that all AXs are successful attacks. Although a class with a smaller distance is hard to generate, we successfully generated AX for all input images by searching for a longer time. On average, it took 10.97 seconds to generate an AX whose distance from the target is 0.2. As in the white-box setting experiments, we did not restrict the number of iterations; we chose a rather large perturbation boundary. We applied these AX to each model for the evaluation, whose results are in graphs in Figure 4.

The results show that our method has significantly suppressed the transferability of AXs. We could also see a clear relation between the transferability of the AXs and the distance between features. We consider it because our ensemble uses the same single model. The transferability surprisingly reaches 1 when the distance between AX and the target in feature space becomes sufficiently small. However, this comes with a much larger perturbation in input images, as shown in Figure 5. The transferability of all models approaches zero as the distance between features becomes large. It is so because AX becomes no longer a successful one with such a large distance. That we eventually have complete transferability with a small distance in feature space indicates that an essential improvement within a single model is necessary, unless we can strictly forbid the small size of perturbation by some means. We consider this is a clear limitation of the ensemble model method.



LOTS via I-FGSM (ArcFace) LOTS via BIM (ArcFace) LOTS via CW (ArcFace)

Figure 4: Black-Box Attacks: the decrease of attack accuracy measures robustness compared to the baseline.

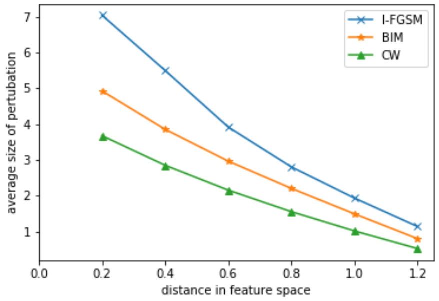


Figure 5: The trade-off between the average size of perturbation and the distance in feature space (ArcFace).

Conclusion

Introducing ensemble diversity in feature extractors for adversarial robustness was not direct. We proposed a novel method that could effectively increase the robustness of feature extractors. Our method might not completely prevent adversarial examples if the perturbation is large. However, it can potentially be used with other tandem methods to increase their robustness. Our method also showed stronger robustness than one of the adversarial training methods.

References

Abbasi, M.; and Gagné, C. 2017. Robustness to Adversarial Examples through an Ensemble of Specialists. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings.

Athalye, A.; Carlini, N.; and Wagner, D. A. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, 274–283.

Cao, Q.; Shen, L.; Xie, W.; Parkhi, O. M.; and Zisserman, A. 2018. VGGFace2: A Dataset for Recognising Faces across Pose and Age. In FG, 67–74. IEEE Computer Society.

Carlini, N.; and Wagner, D. A. 2017a. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017, 3–14.

Carlini, N.; and Wagner, D. A. 2017b. Towards Evaluating the Robustness of Neural Networks. In IEEE Symposium on Security and Privacy, 39–57. IEEE Computer Society.

Chen, S.; Liu, Y.; Gao, X.; and Han, Z. 2018. MobileFaceNets: Efficient CNNs for Accurate Real-time Face Verification on Mobile Devices. CoRR abs/1804.07573.

Cohen, J. M.; Rosenfeld, E.; and Kolter, J. Z. 2019. Certified Adversarial Robustness via Randomized Smoothing. In Chaudhuri, K.; and Salakhutdinov, R., eds., Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of Proceedings of Machine Learning Research, 1310–1320. PMLR.

Dabouei, A.; Soleymani, S.; Taherkhani, F.; Dawson, J.; and Nasrabadi, N. M. 2020. Exploiting Joint Robustness to Adversarial Perturbations. In The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In CVPR, 4690–4699. Computer Vision Foundation / IEEE.

Gilmer, J.; Ford, N.; Carlini, N.; and Cubuk, E. D. 2019. Adversarial Examples Are a Natural Consequence of Test Error in Noise. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, 2280–2289.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Goswami, G.; Agarwal, A.; Ratha, N. K.; Singh, R.; and Vatsa, M. 2019. Detecting and Mitigating Adversarial Perturbations for Robust Face Recognition. Int. J. Comput. Vis. 127(6-7): 719–742. doi:10.1007/s11263-019-01160-w. URL https://doi.org/10.1007/s11263-019-01160-w.

Goswami, G.; Ratha, N. K.; Agarwal, A.; Singh, R.; and Vatsa, M. 2018. Unravelling Robustness of Deep Learning Based Face Recognition Against Adversarial Attacks. In McIlraith, S. A.; and Weinberger, K. Q., eds., Proceedings of the Thirty-Second

AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, 6829–6836. AAAI Press. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17334.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 770–778. IEEE Computer Society.

Hein, M.; and Andriushchenko, M. 2017. Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2266–2276.

Huang, G. B.; Ramesh, M.; Berg, T.; and Learned-Miller, E. 2007. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Bach, F. R.; and Blei, D. M., eds., Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, volume 37 of JMLR Workshop and Conference Proceedings, 448–456. JMLR.org.

Kakizaki, K.; and Yoshida, K. 2020. Adversarial Image Translation: Unrestricted Adversarial Examples in Face Recognition Systems. In SafeAI@AAAI, volume 2560 of CEUR Workshop Proceedings, 6–13. CEUR-WS.org.

Kariyappa, S.; and Qureshi, M. K. 2019. Improving Adversarial Robustness of Ensembles with Diversity Training. CoRR abs/1901.09981.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017a. Adversarial examples in the physical world. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017b. Adversarial Machine Learning at Scale. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. SphereFace: Deep Hypersphere Embedding for Face Recognition. In CVPR, 6738–6746. IEEE Computer Society.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.

Moschoglou, S.; Papaioannou, A.; Sagonas, C.; Deng, J.; Kotsia, I.; and Zafeiriou, S. 2017. AgeDB: The First Manually Collected, In-the-Wild Age Database. In CVPR Workshops, 1997–2005. IEEE Computer Society.

Pang, T.; Xu, K.; Du, C.; Chen, N.; and Zhu, J. 2019. Improving Adversarial Robustness via Promoting Ensemble Diversity. In ICML, volume 97 of Proceedings of Machine Learning Research, 4970–4979. PMLR.

Papernot, N.; McDaniel, P. D.; Wu, X.; Jha, S.; and Swami, A. 2016. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016, 582–597.

Parkhi, O. M.; Vedaldi, A.; and Zisserman, A. 2015. Deep Face Recognition. In Xie, X.; Jones, M. W.; and Tam, G. K. L., eds., Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015, 41.1–41.12. BMVA Press.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018. Semidefinite relaxations for certifying robustness to adversarial examples. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada, 10900–10910.

Rozsa, A.; Günther, M.; and Boult, T. E. 2017. LOTS about attacking deep features. In IJCB, 168–176. IEEE.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; Berg, A. C.; and Li, F. 2015. ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vis. 115(3): 211–252.

Samangouei, P.; Kabkab, M.; and Chellappa, R. 2018. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In CVPR, 815–823. IEEE Computer Society.

Sengupta, S.; Chen, J.; Castillo, C. D.; Patel, V. M.; Chellappa, R.; and Jacobs, D. W. 2016. Frontal to profile face verification in the wild. In WACV, 1–9. IEEE Computer Society.

Sharif, M.; Bhagavatula, S.; Bauer, L.; and Reiter, M. K. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In Weippl, E. R.; Katzenbeisser, S.; Kruegel, C.; Myers, A. C.; and Halevi, S., eds., Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016, 1528–1540. ACM.

Singh, R.; Agarwal, A.; Singh, M.; Nagpal, S.; and Vatsa, M. 2020. On the Robustness of Face Recognition Algorithms Against Attacks and Bias. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, 13583–13589. AAAI Press. URL https://aaai.org/ojs/index.php/AAAI/article/view/7085.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15(1): 1929–1958.

Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep Learning Face Representation by Joint Identification-Verification. In NIPS, 1988–1996.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, 1–9. IEEE Computer Society.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In Bengio, Y.; and LeCun, Y., eds., 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings.

Tramèr, F.; Carlini, N.; Brendel, W.; and Madry, A. 2020. On Adaptive Attacks to Adversarial Example Defenses. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. URL https://proceedings.neurips.cc/paper/2020/hash/11f38f8ecd71867b42433548d1078e38-Abstract.html.

Wan, W.; Zhong, Y.; Li, T.; and Chen, J. 2018. Rethinking Feature Distribution for Loss Functions in Image Classification. In CVPR, 9117–9126. IEEE Computer Society.

Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W. 2018. CosFace: Large Margin Cosine Loss for Deep Face Recognition. In CVPR, 5265–5274. IEEE Computer Society.

Wong, E.; and Kolter, J. Z. 2018. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In Dy, J. G.; and Krause, A., eds., Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of Proceedings of Machine Learning Research, 5283–5292. PMLR.

Wong, E.; Schmidt, F. R.; Metzen, J. H.; and Kolter, J. Z. 2018. Scaling provable adversarial defenses. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada, 8410–8419.

Xu, W.; Evans, D.; and Qi, Y. 2018. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018.

Zhang, K.; Zhang, Z.; Li, Z.; and Qiao, Y. 2016. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. CoRR abs/1604.02878.

Zheng, Z.; and Hong, P. 2018. Robust Detection of Adversarial Attacks by Modeling the Intrinsic Properties of Deep Neural Networks. In Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada, 7924–7933.

Zhong, Y.; and Deng, W. 2019. Adversarial Learning With Margin-Based Triplet Embedding Regularization. In ICCV, 6548–6557. IEEE.