Metric Learning for Adversarial Robustness

Chengzhi Mao
Department of Computer Science
Columbia University
New York, NY 10027
cm3797@columbia.edu

Ziyuan Zhong
Department of Computer Science
Columbia University
New York, NY 10027
zz2521@columbia.edu

Junfeng Yang
Department of Computer Science
Columbia University
New York, NY 10027
junfeng@cs.columbia.edu

Carl Vondrick
Department of Computer Science
Columbia University
New York, NY 10027
vondrick@cs.columbia.edu

Baishakhi Ray
Department of Computer Science
Columbia University
New York, NY 10027
rayb@cs.columbia.edu

Abstract

Deep networks are well-known to be fragile to adversarial attacks. Using several standard image datasets and established attack mechanisms, we conduct an empirical analysis of deep representations under attack, and find that the attack causes the internal representation to shift closer to the "false" class. Motivated by this observation, we propose to regularize the representation space under attack with metric learning in order to produce more robust classifiers. By carefully sampling examples for metric learning, our learned representation not only increases robustness, but also can detect previously unseen adversarial samples. Quantitative experiments show improvement of robustness accuracy by up to 4% and detection efficiency by up to 6% according to Area Under Curve (AUC) score over baselines.

1 Introduction

Despite their impressive accuracy and wide adoption, deep networks remain fragile to adversarial attacks where natural images are perturbed with human-imperceptible, carefully crafted noises [24, 19, 11, 15]. Extensive effort has been devoted to explaining and enhancing the robustness of machine learning models against adversarial attacks [25, 21, 32, 6, 17, 13, 20, 11, 24].

To better understand adversarial attacks, we first conduct an empirical analysis of the latent representations under attack for undefended and robustly trained [17, 13] models. In particular, we investigate what happens to the input representations as they undergo attack. Our results show that the attack shifts the latent representations of adversarial samples away from their *true* class and closer to the *false* class. The adversarial representations often spread across the false class distribution in such a way that the natural images of the false class become indistinguishable from the adversarial images.

Motivated by this empirical observation, we propose to add an additional constraint to the model using metric learning [12, 22, 30] to produce more robust classifiers. We add a triplet loss term on

Preprint. Under review.

the latent representations of adversarial samples to the original loss function. However, the naïve implementation of triplet loss is not effective because the pairwise distances of a natural sample a_1, its adversarial sample a_1', and a randomly selected natural sample of the false class b are hugely uneven. Specifically, given considerable data variance in the false class, b is often far from the decision boundary where a_1' resides, therefore b is too easy an negative sample. To address this, we sample the negative example for each triplet with the closest example in a mini-batch of training data. In addition, we randomly select another sample a_2 in the correct class as the positive example.

Our main contributions are (1) the analysis of latent representations under attack that reveals that the attack shifts the representation closer to the false class even with state-of-the-art robust training and (2) a simple yet effective metric learning method, TLA, that leverages triplet loss to produce more robust classifiers. TLA brings near both the natural and adversarial samples of the same class while enlarging the margins between different classes (Sec. 3). It requires no change to the model architecture and thus can improve the robustness of most of the off-the-shelf deep networks without additional overhead during inference. Evaluation on popular datasets, model architectures, and untargeted, state-of-the-art attacks, including projected gradient descent (PGD), shows that our method classifies adversarial samples more accurately by 1%–4% than prior robust training methods; and makes adversarial attack detection more effectively by up to 6% according to the AUC score.

2 Preliminary Knowledge and Related Work

This work is built upon the prior work on Adversarial Attacks and Robust Training models. We briefly describe them below. Please refer to [11, 15, 17, 5, 7, 33, 13] for more details. We first introduce some notations that we will follow in the rest of the paper.

Notations. For an image classification task, let M be the number of classes to predict, and N be the number of training examples. We formulate the deep network classifier as $F_{\boldsymbol{\theta}}(\mathbf{x}) \in \mathbb{R}^M$ as a probability distribution, where $\mathbf{x}$ is the input variable and $\boldsymbol{\theta}$ denotes the network's parameters to learn (we simply use $F(\mathbf{x})$ most of time); $\mathcal{L}(F(\mathbf{x}), y)$ is the standard loss function.

Adversarial Attacks. In this work, we assume that an adversary is capable of launching adversarial attacks bounded by L_∞ norm, i.e., the adversary can perturb the input pixel by ϵ bounded by L_∞, where $\mathbf{I}(\mathbf{x}, \epsilon)$ is the L_∞ ball centered at $\mathbf{x}$ with radius ϵ. We also assume that the adversary is capable of *untargeted* attack, i.e., the objective is to generate $\mathbf{x}' \in \mathbf{I}(\mathbf{x}, \epsilon)$ such that $F(\mathbf{x}') \neq F(\mathbf{x})$. We call $F(\mathbf{x})$, i.e., the original class of an adversarial image as *true* class, while the mis-predicted class $F(\mathbf{x}')$ is called *false* class. Under the generic threat model of L_∞ and un-targeted attacks, we consider the following white-box (1-5) and black-box attacks (6) which are commonly used in literature:

1. *Fast Gradient Sign Method (FGSM)* [11] generates adversarial examples $\mathbf{x}'$ by $\mathbf{x}' = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(F(\mathbf{x}), y))$ with a single step.
2. *Basic Iterative Method (BIM)* [15] is an extension of FGSM by applying it multiple times with small steps, where the update formula at the i-th step is: $\mathbf{x}_i' = \text{clip}_{\mathbf{x},\epsilon}(\mathbf{x}_{i-1}' + \alpha \cdot \text{sign}(\nabla_{\mathbf{x}_{i-1}'} \mathcal{L}(F(\mathbf{x}_{i-1}'), y)))$, where α is the step size.
3. *Projected Gradient Descent (PGD)* [17] is a variant of the BIM method, where the starting point $\mathbf{x}_0$ is in $\mathbf{I}(\mathbf{x}, \epsilon)$. It can be made more powerful by randomly starting several times and calculating the union of the mispredictions.
4. *Carlini & Wagner (C&W)* [5] We reformulate the C&W method to L_∞ attack as in [17]. We generate $\mathbf{x}'$ by minimizing the loss $g(x) := \max_{\mathbf{x}} (\max_i \{z(\mathbf{x})_i : i \neq t\} - z(\mathbf{x})_t, -\kappa)$, where $z(\mathbf{x}) = \text{logit}(h_{n-1}(x))$ and κ controls the confidence on adversarial examples.
5. *Momentum Iterative Method (MIM)* [7] is a BIM with momentum which won the NeurIPS 2017 Adversarial Competition. MIM updates the gradient $g_i = \text{momentum} \cdot g_{i-1} + \nabla_{\mathbf{x}_{i-1}'} \mathcal{L}(F(\mathbf{x}_{i-1}'), y))$ with momentum and generates the attack iteratively with $\mathbf{x}_i' = \text{clip}_{\mathbf{x},\epsilon}(\mathbf{x}_{i-1}' + \alpha \cdot \text{sign}(g_i))$.
6. *Black-box Attacks (BB)* [26] Black-box attack is to generate adversarial examples on one model and transfer them to the target models. In this paper, we generate the adversarial attacks with PGD on the substitute model and conduct black-box attack to the evaluated models.

Robust Training Methods. Researchers proposed different adversarial training methods to defend neural networks against adversarial perturbations [24, 17]. We use the following state-of-the-art methods as our baseline:

1. *Adversarial training (AT)* [17] achieves adversarial robustness by training the DNN on the created adversarial examples, which goes through densely scrutiny and its robustness is en-

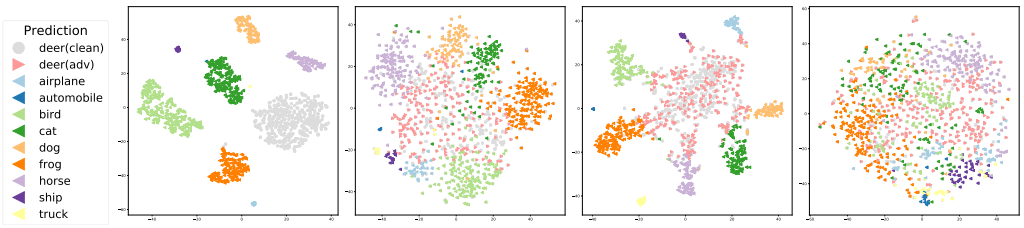
dorsed by Athalye et al [4]. The formulation is proposed as: $\min_{\boldsymbol{\theta}} \rho(\boldsymbol{\theta})$, where $\rho(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}}[\max_{\delta\in S} \mathcal{L}(F_{\boldsymbol{\theta}}(\mathbf{x}+\delta), y)]$, where $S \in \mathbf{I}(\mathbf{x}, \epsilon)$.

2. *Adversarial Logit Pairing (ALP)* [13] builds on top of [17] and introduces an additional loss term that matches the logits from a clean image $\mathbf{x}$ and its corresponding adversarial image $\mathbf{x}'$. However, this method has been known to have a distorted loss function and is not scalable to untargeted attack [9, 18].

Several other robust training methods do not focus on the L_∞ adversarial attack or only have limited empirical robustness, so we do not study them under our threat model. For example, stability training regularizes the natural loss with $L_{stability}(\mathbf{x}, \mathbf{x}') = ||h(\mathbf{x}) - h(\mathbf{x}')||_2$, where $\mathbf{x}'$ is the result of some natural transformation such as rotation, not adversarial attack. DeepDefense [32] considers integrating an adversarial attack directly in the loss function without generating adversarial examples. Model ensemble is another way to improve robustness [25, 20].

3 Qualitative Analysis of Latent Representations under Adversarial Attack

We begin our investigation by analyzing how the adversarial images are represented by different models. Figure 1 shows the visualization of the high dimensional latent representation of sampled CIFAR-10 images with t-SNE [28, 3]. Here, we see the penultimate fully connected (FC) layer of three existing models: standard undefended model (UM), model after adversarial training (AT) [17], model after adversarial logit pairing (ALP) [13], and our proposed model that we will discuss later. Though all the adversarial images belong to the same *true* class, UM separates them into different *false* classes with large margins. This shows UM is highly non-robust against adversarial attacks because with such latent representations it is very easy to craft an adversarial image that will be mistakenly classified to a different category. With AT and ALP methods, the representations are getting closer together, but one could still be able to discriminate. Note that, a good robust model will bring the representations of the adversarial images closer to their original *true* classes so that it will be difficult to discriminate the adversarial images from the original images. We will leverage this observation to design our approach.

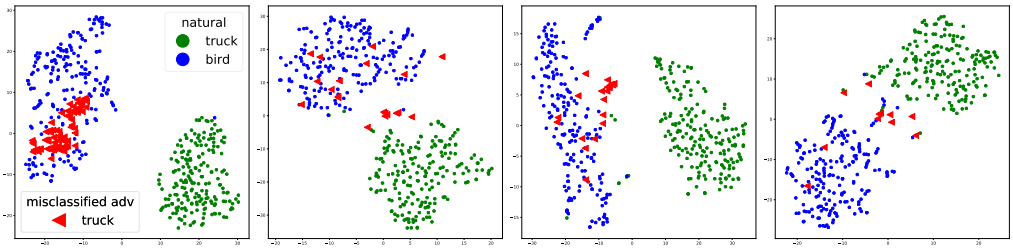


(a) UM (b) AT (c) ALP (d) TLA

Figure 1: **t-SNE Visualization of adversarial images from the same *true* class which are mistakenly classified to different *false* classes.** These are representations of second to last layer of 1000 adversarial examples crafted from 1000 natural (clean) test examples from CIFAR-10 dataset, where the *true* class is "deer". The different colors represent different *false* classes. The gray dots further show 500 randomly sampled natural deer images. Notice that for (a) undefended model (UM), the adversarial attacks clearly separate the images from the same "deer" category into different classes. (b) adversarial training (AT) and (c) adversarial logit pairing (ALP) method still suffer from this problem at a reduced level. In contrast, our proposed ATL (see (d)) clusters together all the examples from the same *true* class, which improves overall robustness.

In Figure 2, we further analyze how the representation of images of one class is attacked into the neighborhood of another class. The green and blue dots are the natural images of trucks and birds respectively. The red triangles are the adversarial images of truck mispredicted as birds. For UM model (Figure 2a), all the adversarial attacks successfully get into the center of the false class. The AT and ALP models achieve some robustness by separating some adversarial images from natural images, but most adversarial images are still inside the false class. A good robust model should promote the representations of adversarial examples away from the false class, as shown in Figure 2d. Such separation not only improves the adversarial classification accuracy but also helps to reject the mispredicted adversarial attacks, because the mispredicted adversaries tend to lie on the edge.

Based on these two observations, we build a new approach that ensures adversarial representations will be (i) closer to the natural image representations of their true classes, and (ii) farther from the natural image representations of corresponding false classes.



(a) UM (b) AT (c) ALP (d) TLA

Figure 2: **Illustration of the separation margin of adversarial examples from the natural images of the corresponding false class.** We show t-SNE visualization of the second to last layer representation of test data from two different classes across four models. The blue and green dots are 200 randomly sampled natural images from "bird" and "truck" classes respectively. The red triangles denote adversarial (adv) perturbed "truck" images but mispredicted as "bird". Notice that for (a) UM, the adversarial examples are moved to the center of the false class which is hard to separate from the natural images of the false class. (b) AT and (c) ALP achieve some robustness by separating adversarial and false natural images, but they are still too close to each other. Plot (d) shows proposed TLA promotes the mispredicted adversarial examples to lie on edge and can still be separated from natural images of the false class, which improves the robustness.

4 Approach

Inspired by the adversarial feature space analysis, we add an additional constraint to the model using metric learning. Our motivation is that the triplet loss function will pull all the images of one class, both natural and adversarial, closer while pushing the images of other classes far apart. Thus, an image and its adversarial counterpart should be on the same manifold, while all the members of the 'false' class should be forced to be separated by a large margin.

Triplet Loss. Triplet loss is a widely used strategy for metric learning. It trains on a triplet input $\left\{\left(\mathbf{x}_a^{(i)}, \mathbf{x}_p^{(i)}, \mathbf{x}_n^{(i)}\right)\right\}$, where the elements in the positive pair $\left(\mathbf{x}_a^{(i)}, \mathbf{x}_p^{(i)}\right)$ are from the same class and the negative pair $\left(\mathbf{x}_a^{(i)}, \mathbf{x}_n^{(i)}\right)$ are from different classes [22, 12]. $\mathbf{x}_p^{(i)}$, $\mathbf{x}_a^{(i)}$, and $\mathbf{x}_n^{(i)}$ are referred as *positive*, *anchor*, and *negative* examples of the triplet loss. The embeddings are optimized such that examples of the same class are pushed together and the examples of different classes are pulled apart by some margin [23]. The standard triplet loss for clean images is as follows:

$$\sum_i^N \mathcal{L}_{trip}(\mathbf{x}_a^{(i)}, \mathbf{x}_p^{(i)}, \mathbf{x}_n^{(i)}) = \sum_i^N [D(h(\mathbf{x}_a^{(i)}), h(\mathbf{x}_p^{(i)})) - D(h(\mathbf{x}_a^{(i)}), h(\mathbf{x}_n^{(i)})) + \alpha]_+$$

where, $h(\mathbf{x})$ maps from the input $\mathbf{x}$ to the embedded layer, $\alpha \in \mathbb{R}^+$ is a hyper-parameter for margin and $D(h(\mathbf{x}_i), h(\mathbf{x}_j))$ denotes the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the embedded representation space. There are many different distance measurements for constructing the triplet loss. In this paper, we define the embedding distance between two examples as the angular distance [29]: $D(h(\mathbf{x}_a^{(i)}), h(\mathbf{x}_{p,n}^{(j)})) = 1 - \frac{|h(\mathbf{x}_a^{(i)}) \cdot h(\mathbf{x}_{p,n}^{(j)}))|}{||h(\mathbf{x}_a^{(i)})||_2 ||h(\mathbf{x}_{p,n}^{(j)}))||_2}$, such that we encode the information in the angular space. Since the absolute norm of the embedding does not change the prediction of the classifier, use of angular loss excludes the influence of the norm on the loss value.

Metric Learning for Adversarial Robustness. In this case, we use triplet loss such that an image $\mathbf{x}_p^{(i)}(positive)$ and its adversarial counterpart $\mathbf{x}_a^{(i)}(anchor)$ come closer while another image from the false class $\mathbf{x}_n^{(i)}(negative)$ goes far in the embedded representation space. We add triplet loss directly to the representation of the second to last layer. Different from standard triplet loss where all the elements in the triplet loss term are clean images [22, 33], at least one element in the triplet loss under our setting will be adversarially perturbed image. Note that generating adversarial examples is more computational intensive compared with just taking the clean images. For efficiency, we only generate one adversarial perturbed image for each triplet data, using the same method introduced by Madry et al [17]. Specifically, we generate the adversarial image $\mathbf{x}'^{(i)}$ based on $\nabla_{\mathbf{x}} \mathcal{L}(F(\mathbf{x}), y)$ (standard loss without the triplet loss) with PGD method given a clean image $\mathbf{x}^{(i)}$. We do not add the triplet loss term into the loss of adversarial example generation because it causes non-convergence.

Figure 3: Illustration of the triplet loss for adversarial robustness (TLA). The anchor is moved from the true class to the false class by adversarial perturbation. TLA learns to pull the *anchor* and *positive* from the true class closer, and push the *negative* of false classes apart.

The other elements in the triplet data are clean images, where the positive is a clean image from the anchor's class and the negative is from a different class. We forward the triplet data in parallel through the model and jointly optimize the cross-entropy loss and the triplet loss, which enables the model to capture the invariant representation with semantic meaning. The total loss function is formulated as follows:

$$\mathcal{L}_{all} = \sum_{i}^{N} \mathcal{L}_{ce}(f(\mathbf{x}_a'^{(i)}), y^{(i)}) + \lambda_1 \mathcal{L}_{trip}(h(\mathbf{x}_a'^{(i)})), h(\mathbf{x}_p^{(i)}), h(\mathbf{x}_n^{(i)})) + \lambda_2 \mathcal{L}_{norm} \tag{1}$$

$$\mathcal{L}_{norm} = ||h(\mathbf{x}_a'^{(i)})||_2 + ||h(\mathbf{x}_p^{(i)})||_2 + ||h(\mathbf{x}_n^{(i)})||_2$$

where λ_1 is a positive coefficient trading off the two losses, $\mathbf{x}_a'^{(i)}$ is an adversarially perturbed image based on $\mathbf{x}^{(i)}$, λ_2 is the weight for the feature norm decay term, which prevents the L_2 norm of the feature from getting too large.

Notice that the adversarial perturbed images can either be anchor or positive example. We choose the adversarial example as the anchor according to the experimental result (refer to the TLA-SA in Sec 5). Intuitively, the adversarial image is picked as anchor because it tends to be closer to the decision boundary between the "true" class and the "false" class. As an anchor, it is able to be considered in both the positive pair and the negative pair which gives more useful gradient for the optimization. An illustration of the modified triplet loss for adversarial robustness is shown in Figure 3.

Negative Sample Selection. In addition to the anchor selection, the selection of the negative example is crucial for the training process, because most of the negative examples are easy examples which already satisfied the margin constraint of pairwise distance and thus contribute little to a useful gradient [22, 8]. In this paper, we select negative samples which is the nearest to the anchor based on the representation angular distance we predefined from a false class. As a result, our model is able to learn to enlarge the boundary between the adversarial perturbed samples and their closest negative samples from the other classes.

Unfortunately, finding the closest negative samples from the entire training set is computationally intensive. Besides, using very hard negative examples have been found to significantly decrease the network's convergence speed [22]. Instead, we use a semi-hard negative example, where we select the closest sample in a mini-batch. We demonstrate the advantage of this sampling strategy by comparing it with the random sampling (TLA-RN). The results are shown in Sec 5. A similar strategy of sampling negative sample with adversarial generation process mentioned in DAML [8] could also be applied here, which uses adversarial generator to exploit more unseen hard negative examples.

Implementation Details. We study the embedding of the second to last layer of the neural network for classification task, because it will be followed by a linear classifier, which is beneficial because small fluctuation to this layer only brings monotonous adjustment to the output controlled by some Lipschitz constant. We don't use the same logit layer as ALP [13] but the hidden layer ahead of it. Compared with the logit layer, the penultimate layer tends to have more information because it usually has much higher dimensions. The details of the algorithm are introduced in the supplementary.

5 Experiments

Experimental Setting. We validate our method on different model architectures across three popular datasets: MNIST [16], CIFAR-10 [14] and Tiny-ImageNet [1]. We train our models from scratch and compare the performance of the models with the following baselines: **Undefended Model (UM)** refers to standard training without adversarial samples, **Adversarial Training (AT)** refers to min-max optimization method proposed in [17], **ALP** refers to the adversarial logit pairing method which is currently the state-of-the-art [13]. We use **TLA** to denote the triplet loss adversarial training

Table 1: Classification accuracy under various L_∞ bounded *untargeted* attacks on MNIST (L_∞ =0.3), CIFAR-10 (L_∞ =8/255), and Tiny-ImageNet (L_∞ =8/255). TLA improves the adversarial accuracy by 1.86%, 4.12% , and 1.05% respectively. The best results of each column are in **bold** and underline shows the empirical lower bound for each method (the lowest accuracy of each row if any).

	MNIST									
	Attacks	Clean	FGSM	BIM	C&W	PGD	PGD	20PGD	MIM	BB
	(Steps)	-	(1)	(40)	(40)	(40)	(100)	(100)	(200)	(100)
Methods	UM	99.20%	34.48%	0%	0%	0%	0%	0%	0%	81.81%
	AT	99.24%	97.31%	95.95%	96.66%	96.58%	94.82%	93.87%	95.47%	96.67%
	ALP	98.91%	97.34%	96.00%	96.50%	96.62%	95.06%	94.93%	95.41%	96.95%
	TLA-RN	99.50%	98.12%	97.17%	97.17%	97.64%	**97.07%**	96.73%	**96.84%**	97.69%
	TLA-SA	99.44%	98.14%	97.08%	**97.45%**	97.50%	96.78%	95.64%	96.45%	97.65%
	TLA	**99.52%**	**98.17%**	**97.32%**	97.25%	**97.72%**	96.96%	**96.79%**	96.64%	**97.73%**
	CIFAR-10									
	Attacks	Clean	FGSM	BIM	C&W	PGD	PGD	20PGD	MIM	BB
	(Steps)	-	(1)	(7)	(30)	(7)	(20)	(20)	(40)	(7)
Methods	UM	**95.01%**	13.35%	0%	0%	0%	0%	0%	0%	7.60%
	AT	87.14%	55.63%	48.29%	46.97%	49.79%	45.72%	45.21%	45.16%	62.83%
	ALP	89.79%	**60.29%**	50.62%	47.59%	51.89%	48.50%	45.98%	45.97%	67.27%
	TLA-RN	81.02%	55.41%	51.44%	49.66%	52.50%	49.94%	45.55%	49.63%	65.96%
	TLA-SA	86.19%	58.80%	52.19%	49.64%	53.53%	49.70%	49.15%	49.29%	61.67%
	TLA	86.21%	58.88%	**52.60%**	**50.69%**	**53.87%**	**51.59%**	**50.03%**	**50.09%**	**70.63%**
	Tiny ImageNet									
	Attacks	Clean	FGSM	BIM	C&W	PGD	PGD	20PGD	MIM	BB
	(Steps)	-	(1)	(10)	(10)	(20)	(20)	(20)	(40)	(10)
Methods	UM	**53.85%**	8.07%	0.69%	0.04%	0.84%	0.39%	0%	0%	**27.08%**
	AT	34.60%	19.34%	13.66%	11.42%	13.79%	13.46%	13.33%	13.57%	19.62%
	ALP	43.20%	19.85%	11.24%	9.70%	11.37%	10.87%	10.59%	10.75%	22.08%
	TLA-RN	35.88%	19.93%	14.08%	11.42%	14.23%	13.89%	13.65%	11.29%	20.27%
	TLA-SA	34.91%	19.42%	13.57%	11.34%	13.75%	13.33%	13.19%	13.22%	19.90%
	TLA	36.79%	**20.95%**	**14.89%**	**12.47%**	**15.05%**	**14.65%**	**14.43%**	**14.74%**	21.09%

mentioned in Section 4. To further evaluate our design choice, we study two variants of TLA: **Random Negative (TLA-RN)**, which refers to our proposed triplet loss training method with a randomly sampled negative example, and **Switch Anchor (TLA-SA)**, which sets the anchor to be natural example and the positive to be adversarial example (i.e., switching the anchor and the positive of our proposed method).

We conduct all of our experiments using TensorFlow v1.13 [2]. We adopt the untargeted adversarial attacks during all of our training process, and evaluate the models with both white and black-box *untargeted* attacks instead of the targeted attacks following the suggestions in [10] (a defense which is only robust to targeted adversarial attacks is weaker than one which is robust to untargeted adversarial attacks). The PGD and 20PGD in our table refer to the PGD attacks with random start of 1 and 20 times, respectively. For ALP, we set $\lambda = 0.5$ as mentioned in the original paper. All the other implementation details are discussed in the supplementary material.

5.1 Effect of TLA on Robust Accuracy

MNIST consists of a training set of 55,000 images (excluding the 5000 images for validation as in [17]) and a testing set of 10,000 images. We use a variant of LeNet CNN architecture which has a larger model capacity. The details of network architectures and hyper-parameters are summarized in the supplementary material. We adopt the $L_\infty = 0.3$ bounded attack during the training and evaluation. We generate adversarial examples using PGD with 0.01 step size for 40 steps during the training. In addition, we conduct different types of $L_\infty = 0.3$ bounded attacks to achieve good evaluations. The adversarial classification accuracy of different models under various adversarial attacks are shown in Table 1. As shown, we improve the empirical state-of-the-art adversarial accuracy by up to **1.86%** on 20PGD attacks (100 steps PGD attacks with 20 times of random restart), along with **0.28%** improvement on clean set.

CIFAR-10 consists of 32×32×3 color images in 10 classes, with 50k images for training and 10k images for testing. We follow the same wide residual network architecture and the same hyper-parameters settings as AT [17]. As shown in Table 1, our method achieves up to **4.12%** adversarial

accuracy improvement over the baseline methods under the strongest 20PGD attacks (20 steps PGD attack with 20 times of restart). Note that our method results in a minor decrease of standard accuracy but such loss of generic accuracy is observed in all the existing robust training models [27]. The comparison with TLA-RN illustrates the effectiveness of the negative sampling strategy. According to the result of the TLA-SA, our selection of the adversarial example as the anchor also achieves better performance than the method which chooses the clean image as the anchor.

Tiny Imagenet is a tiny version of ImageNet consisting of color images with size $64\times64\times3$ belonging to 200 classes. Each class has 500 training images and 50 validation images. We adopt $L_\infty = 8/255$ for both training and validation. During training, we use 7 step PGD attack with step size $2/255$ to generate the adversarial samples. As shown in Table 1, our proposed model achieves higher adversarial accuracy under white box adversarial attacks by up to **1.10%** on 20PGD (20 steps of PGD with 20 random restarts) attacks along with a minor decrease of standard accuracy.

5.2 Effect of TLA on Adversarial vs. Natural Image Separation

As shown in Figure 2a, in the 'Undefended Mode', the representations of adversarial images are shifted toward the false class. A good robust training model should separate them apart. To quantitatively evaluate how well TLA can separate the adversarial examples from the natural images of the corresponding 'false' classes, we define the following metric.

Let $\{c_k^i\}$ denote the *embedded representations* of all the natural images from class c_k, where $i = 1, \ldots, |c_k|$, and $|c_k|$ is the total number of images in class c_k. Then, the average pairwise within-class distance of these embedded images is: $\sigma_{c_k}^{ntrl} = \frac{2}{|c_k|(|c_k|-1)} \sum_{i=1}^{|c_k|-1} \sum_{j=i+1}^{|c_k|} D(c_k^i, c_k^j)$. Let $\{c_k'^q\}$ further denote embedded representations of all the adversarial examples that are misclassified to class c_k, where $q = 1, \ldots, |c_k'|$, and $|c_k'|$ is the total number of such examples. Note that, class c_k is the 'false' class to those adversarial images. Then, the distance between an adversarial images $c_k'^{\,i}$ and a natural image c_k^j is: $D(c_k'^{\,i}, c_k^j)$, and the average pair-wise distance between adversary image and natural images is: $\sigma_{c'_k}^{adv} = \frac{1}{|c_k'||c_k|} \sum_{i=1}^{|c_k'|} \sum_{j=1}^{|c_k|} D(c_k'^{\,i}, c_k^j)$. We then define the ratio $r_{c_k} = \frac{\sigma_{c_k}^{adv}}{\sigma_{c_k}^{ntrl}}$ as a metric to evaluate how close the adversarial images are w.r.t. the natural images of the 'false' class while compared with the average pairwise within-class distance of all the natural images of that class. Finally, for all classes we compute the average ratio as $r = \frac{1}{M} \sum_{k=1}^{M} (r_{c_k})$. Note that, any good robust method should increase the value of r, indicating σ^{adv} is far from σ^{ntrl}, i.e., they are better separated than the natural cluster, as shown in Figure 2d.

Table 2: Average Ratio (r) of mean distance between adversary points and natural points over the mean intra-class distance. The results illustrate that TLA increases the relative distance of adversarial images w.r.t. the natural images of the respective false classes. The best results of each column are in **bold**.

	Dataset	MNIST		CIFAR-10		Tiny-ImageNet	
	Perturbation Level	$L_\infty = 0.03$	$L_\infty = 0.3$	$L_\infty = \frac{8}{255}$	$L_\infty = \frac{25}{255}$	$L_\infty = \frac{8}{255}$	$L_\infty = \frac{25}{255}$
Methods	UM	1.485	1.194	0.8082	0.9208	0.9236	0.9293
	AT [17]	1.288	1.308	1.053	1.007	0.9774	0.9504
	ALP [13]	1.398	1.394	1.038	1.210	0.9840	0.9360
	TLA	**1.810**	**1.847**	**1.093**	**1.390**	**0.9979**	**0.9995**

For every dataset, we estimate the ratios under two perturbation levels for all the models. The results are shown in Table 2. As we can see, the adversarial attacks tend to shift their latent representation toward the false class. Note that for CIFAR-10 and Tiny-ImageNet, the adversarial examples are even closer ($r < 1$) to the false class's manifold than the corresponding natural images to itself. In all the settings, r values of TLA are highest as compared to the other training methods. This indicates TLA is most effective to pull apart the misclassified adversary examples from their false class under both small and large perturbations attacks.

We also define a similar metric to show TLA tends to pull closer the adversary images to their true class, as shown in Figure 1d. Please refer to the supplementary material for more details.

We further conduct the nearest neighbor analysis on the latent representations across all the models. The results illustrate the advantage of our learned representations in nearest neighbor retrieval (See Figure 4)—while querying with an adversarial image TLA performs better in retrieving true class members than others. Table 3 shows that the latent representations of TLA achieves higher accuracy using K-Nearest Neighbors classifier than its competitors.

Table 3: Accuracy of K-Nearest Neighbors classifier with $K = 50$ illustrating TLA has better similarity measures in embedding space even with adversarial samples. The best results of each column are in **bold**.

	Dataset	MNIST		CIFAR-10		Tiny-ImageNet	
	Type	Adv	Natural	Adv	Natural	Adv	Natural
Method	AT	93.01%	98.68%	47.46%	87.06%	14.18%	30.27%
	ALP	95.20%	98.43%	48.85%	**89.63%**	14.56%	**38.14%**
	TLA	**96.98%**	**99.47%**	**51.74%**	86.29%	**15.90%**	35.88%

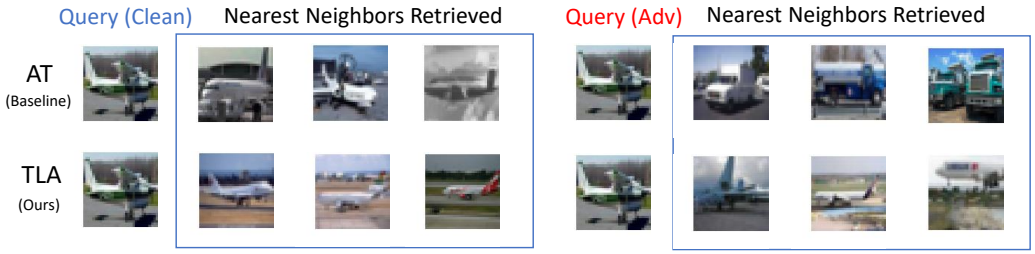
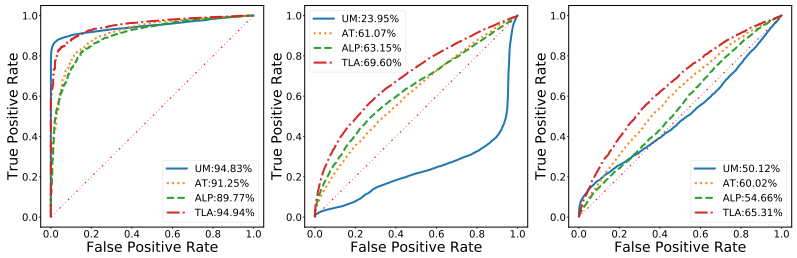


Figure 4: Visualization of nearest neighbor images while querying about a “plane" on AT and TLA trained models separately. For a natural query image, both methods retrieve correct images (left column). However, given a maliciously perturbed query image (right column), the AT retrieved false “truck" images indicate the perturbation moves the representation of the “plane" into the neighbors of “truck," while TLA still retrieves images from the true "plane" class.

5.3 Effect of TLA on Adversarial Image Detection



(a) MNIST (b) CIFAR-10 (c) Tiny-ImageNet

Figure 5: The ROC curve and AUC scores of applying GMM on trained model (with perturbation level $\epsilon = 0.03/1$(40 steps) for MNIST, $\epsilon = 8/255$(7 steps) for CIFAR-10, and $\epsilon = 8/255$(7 steps) for Tiny-ImageNet) for misclassified example detection on 10000 natural test images and 10000 adversary test images (with perturbation level $\epsilon = 0.3/1$(100 steps) for MNIST, and $\epsilon = 25/255$(20 steps) for CIFAR-10, and $\epsilon = 25/255$(30 steps) for Tiny-ImageNet). The numerical results for AUC score are shown in the legend, which shows TLA (our method) achieves higher detection efficiency for adversarial examples compared with other baseline methods.

Efficiently detecting the adversarial inputs is another dimension to improve model's robustness, which is orthogonal to enhancing the model's overall prediction accuracy. Given TLA can better separate the adversarial examples from the natural examples of the false class, we should be able to detect the mis-classified adversarial examples more efficiently by filtering out the outliers. To evaluate how efficiently TLA can detect mis-classified images in a test setting, we conduct the following experiments.

Following the adversarial detection method proposed in [34], we train a Gaussian Mixture Model for 10 classes where the density function of each class is modeled by one Gaussian distribution. For each test image, we assign a confidence score of a class based on the Gaussian distribution density of the class at that image, as shown in [31]. We assign such confidence score for all the 10 classes for each test image. Then, we pick the class with the largest confidence value as the potential class of the image. Then, we rank all the test images based on the confidence value of their respective assigned class. We further reject those with lower confidence scores below a certain threshold indicating they are potential misclassified examples. Note that, such a method can be used as an additional confidence metric to detect adversarial examples in a real-world setting.

As shown in Figure 5, the results of the ROC-curves and AUC score demonstrate that our learned representations are superior in adversarial detection task and induce better confidence estimation. The detection results here are consistent with the visual results shown in Figure 2.

6 Conclusion

In this work, we take the first step to analyze the property of high dimensional latent representation of adversarial examples and find that the attack causes the embedding move closer to the false class such that the adversarial and natural images are almost indistinguishable. Inspired by this observation, we propose a simple but effective metric learning based approach for adversarial robustness. We empirically validate our approach using three popular datasets across various deep network architectures. Our results show that along with increasing robust accuracy, our latent representation is also important in detecting adversarial examples.

References

[1] Tiny imagenet visual recognition challenge.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] Sanjeev Arora, Wei Hu, and Pravesh K. Kothari. An analysis of the t-sne algorithm for data visualization. In *COLT 2018*, 03 2018.

[4] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pages 274–283, 2018.

[5] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57, 2017.

[6] Moustapha Cissé, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 854–863, 2017.

[7] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, pages 9185–9193. IEEE Computer Society, 2018.

[8] Yueqi Duan, Wan qing Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2780–2789, 2018.

[9] Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *CoRR*, abs/1807.10272, 2018.

[10] Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *CoRR*, abs/1807.10272, 2018.

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[12] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *ICLR*, 2015.

[13] Harini Kannan, Alexey Kurakin, and Ian J. Goodfellow. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018.

[14] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[15] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2017.

[16] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

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

[18] Marius Mosbach, Maksym Andriushchenko, Thomas Alexander Trost, Matthias Hein, and Dietrich Klakow. Logit pairing methods can fool gradient-based attacks. *CoRR*, abs/1810.12042, 2018.

[19] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436, 2015.

[20] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. *CoRR*, abs/1901.08846, 2019.

[21] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *CoRR*, abs/1805.06605, 2018.

[22] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015.

[23] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012. IEEE Computer Society, 2016.

[24] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[25] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. *CoRR*, abs/1705.07204, 2017.

[26] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[27] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *stat*, 1050:11, 2018.

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[29] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *ICCV*, pages 2612–2620. IEEE Computer Society, 2017.

[30] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[31] Xi Wu, Uyeong Jang, Jiefeng Chen, Lingjiao Chen, and Somesh Jha. Reinforcing adversarial robustness using model confidence induced by adversarial training. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 5330–5338. PMLR, 2018.

[32] Ziang Yan, Yiwen Guo, and Changshui Zhang. Deep defense: Training dnns with improved adversarial robustness. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 417–426, USA, 2018. Curran Associates Inc.

[33] Stephan Zheng, Yang Song, Thomas Leung, and Ian J. Goodfellow. Improving the robustness of deep neural networks via stability training. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4480–4488, 2016.

[34] Zhihao Zheng and Pengyu Hong. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7913–7922. Curran Associates, Inc., 2018.