Learning Discriminative Features for Adversarial Robustness

Ryan Hosler, Tyler Phillips, Xiaoyuan Yu, Agnideven Sundar, Xukai Zou, Feng Li
Department of Computer Science
Indiana University-Purdue University Indianapolis
Indianapolis, Indiana 46202, USA
{rjhosler, phillity, xyu1, agpalan, xzou, fengli}@iupui.edu

***Abstract*—Deep Learning models have shown incredible image classification capabilities that extend beyond humans. However, they remain susceptible to image perturbations that a human could not perceive. A slightly modified input, known as an Adversarial Example, will result in drastically different model behavior. The use of Adversarial Machine Learning to generate Adversarial Examples remains a security threat in the field of Deep Learning. Hence, defending against such attacks is a studied field of Deep Learning Security. In this paper, we present the Adversarial Robustness of discriminative loss functions. Such loss functions specialize in either inter-class or intra-class compactness. Therefore, generating an Adversarial Example should be more difficult since the decision barrier between different classes will be more significant. We conducted White-Box and Black-Box attacks on Deep Learning models trained with different discriminative loss functions to test this.**

Moreover, each discriminative loss function will be optimized with and without Adversarial Robustness in mind. From our experimentation, we found White-Box attacks to be effective against all models, even those trained for Adversarial Robustness, with varying degrees of effectiveness. However, state-of-the-art Deep Learning models, such as Arcface, will show significant Adversarial Robustness against Black-Box attacks while paired with adversarial defense methods. Moreover, by exploring Black-Box attacks, we demonstrate the transferability of Adversarial Examples while using surrogate models optimized with different discriminative loss functions.

***Index Terms*—Adversarial Machine Learning, Deep Learning, Metric Learning, Discriminative Loss Function**

I. Introduction

Since their discovery in 2014, Adversarial Examples [1] have posed a significant security threat to Machine Learning and Deep Learning models. Generating Adversarial Examples typically use the back-propagation gradient information derived from a target model [2] (White-Box attack) or a supervised attack model similar to the target model [3] (Black-Box attack). Using this gradient information and gradient ascent, an attacker can alter "clean" (un-altered) data instances in such a way that is imperceptible to human observers. The resulting altered data instances, known as Adversarial Examples, can then be used to cause a target model to make targeted or untargeted mistakes with great ease. Forming Adversarial Examples can be done through a variety of adversarial attacks [2], [4]–[6]. Moreover, Adversarial Examples possess an ability known as transferability: Adversarial Examples that are effective against one model are typically effective against another. This property will make even Black-Box attacks effective and difficult to thwart.

As several types of adversarial attacks have been proposed, the research community has investigated how to make models robust to such attacks. The most successful and widely accepted method, Adversarial Training (AT) [6], was formally proposed in 2017. AT involves generating Adversarial Examples during model training and using the Adversarial Examples to train models rather than an original unmodified dataset. Later in 2019, Adversarial Logit Pairing (ALP) [7] was proposed. ALP involves training models with a mix of unmodified Clean Examples and their corresponding Adversarial Examples. In addition to learning from both the clean and adversarial versions of training examples, ALP adds a regression term to loss functions to force models to make similar predictions for both clean and adversarial versions of training examples. Through this additional loss function term, ALP has achieved state-of-the-art Adversarial Robustness results.

Besides the general AT and ALP approach, researchers have investigated additional modifications that can be applied to model training to offer increased Adversarial Robustness. In 2019, researchers proposed using a metric learning-based loss function [8]. More specifically, they proposed using Triplet Loss [9] (along with AT or ALP) for training adversarially robust models. Triplet Loss's use is quite intriguing as it aims to combat Adversarial Example attacks using "discriminative" features.

Triplet Loss was first proposed in 2015 for the task of face recognition. In many face recognition tasks, facial features extracted from facial images must be "discriminative". Features are considered "discriminative" if, given two feature vectors, it is easy to discern if the feature vectors belong to a class or two different classes. This discriminative property is typically achieved through training a Deep Learning model using a metric-based loss function, such as Triplet Loss. Such metric learning-based loss functions directly guide models in learning intra-class compactness and inter-class distinguishability in the feature space. Furthermore, these loss functions often even ensure a margin of separation between classes in the feature space. As a result, models trained using metric learning-based loss functions can extract discriminative features.

This discriminative property is advantageous when combating Adversarial Examples. If the feature-space has robust intra-

class compactness and inter-class variability, it is difficult for an attacker to trick the discriminative model through the use of imperceptible adversarial attacks. This was shown in [8] since their Triplet Loss models were more robust to adversarial attacks than their Softmax models. Therefore, we further investigate the use of metric learning-based loss functions to provide Adversarial Robustness. The contributions of this paper are as follows:

- We provide a comprehensive study of discriminative loss functions versus adversarial attacks.
- We compare AT and ALP with each discriminative loss function to test their effectiveness against adversarial attacks.
- We compare each discriminative loss function's transferability when used as surrogate models for adversarial attacks.
- We demonstrate that White-Box attacks are typically effective in every scenario while adversarial defense techniques thwart Black-Box attacks.

Specifically, we evaluate advanced Euclidean distance-based methods, such as Contrastive Loss (ConL) [10] and Center Loss (CenL) [11], and a recently proposed angular margin-based method, the Additive Angular Margin Loss (AAML) [12]. The code used for this research can be found on our github (https://github.com/rjhosler/arcface-pytorch).

The paper is organized as follows: Section 2 discusses related works, section 3 explains the attacks used in our experiments, section 4 describes the defense methods used in our experiments, section 5 details the discriminative loss functions that are tested for Adversarial Robustness, section 6 explains the experiments and their results, and section 7 provides concluding remarks.

II. Related Works

Deep Learning security threats and their implications have been a recently researched topic since the discovery of Adversarial Examples in [1]. However, current literature in Deep Learning Security is not limited to attacks that affect model performance. Another related topic, inference attacks, will attempt to extrapolate a model's training dataset. An example of research in this area involves Nasr et al. investigating the effectiveness of Membership Attacks [13].

A Membership Attack is a binary classification problem: given a data point, determine if it belongs in the training data set. This attack is easy to evaluate since it has a 50% baseline (random guessing). Black-Box versions only have access to the model's probability vector; therefore, they exploit the statistical difference between a model predicting unseen data versus training data. Nasr et al. only managed 54.5% inference accuracy when attacking DenseNet trained on CIFAR-100 [13]. However, a more potent White-Box attack utilizing model gradients results in 74.3% inference accuracy.

Given the interest in Membership Attacks, there exists current literature on privacy-preserving methods. For example, Yu et al. utilize Differential Privacy [14]. Differential Privacy implies that any input change should not alter an algorithm's output. Yu et al. implemented a general approach, Concentrated Differential Privacy, that constrains cumulative data points rather than each data point individually [14]. They implemented data batching methods for this technique to allow for easy implementation for Neural Network training.

While Privacy is a concern for Deep Learning Security, this research focuses on Adversarial Examples. There exist various methods for Adversarial Example generation motivated by computational efficiency and attack effectiveness. Section III describes the methods implemented for testing the robustness of discriminative loss functions. Moreover, we implement adversarial defense methods known for increasing Adversarial Robustness for any image classification model. Specifically, the baseline defenses for comparing discriminative loss functions are AT and ALP. Those defense methods are detailed in Section IV.

Regarding ArcFace (AAML) itself, there exist adversarial attacks that are effective against the facial recognition model. For example, Pautov et al. demonstrate an adversarial patching method against ArcFace-100 [15]. Such an attack shows potential real-world vulnerabilities with AAML as a facial recognition method. However, such research does not compare the Adversarial Robustness ArcFace to other discriminative loss functions.

The focus of this research is comparing the Adversarial Robustness of differing discriminative loss functions. Previously, Moa et al. used Triplet Loss, AT, and ALP to demonstrate the effectiveness of Metric Learning for Adversarial Robustness [8]. They determined that the discriminative features learned by Triplet Loss reduced the effectiveness of Adversarial Example Attacks. Since AAML [12] has been a more recent metric loss function than Triplet Loss for facial recognition, it ought to reduce the effectiveness of Adversarial Example Attacks further. Moreover, we compare Triplet Loss and AAML to other discriminative loss functions described in section V.

III. Attacks

In this section, we will review the Adversarial Example attacks used for our experimentation. Specifically, we detail the iterative and non-iterative methods used for conduction White-Box attacks. Moreover, we explain how we conduct Black-Box attacks.

A. Fast Gradient Sign Method (FGSM)

Non-iterative attacks only require one step for generating an Adversarial Example. FGSM was the first method used for creating an Adversarial Example [1] and act as a base attack for other methods. Thus, it is the non-iterative method we utilize here. FGSM creates an Adversarial Example by finding the gradients of the neural network and using the gradients to maximizing the loss, which is usually the cross-entropy loss [2]. The objective function is summarised using the following expression:

$$x^{adv} = x + \epsilon * \text{sign}(\Delta_x L(x, y, \theta)) \tag{1}$$

Here the gradients are taken concerning the original image, so the gradients' signs show the direction to maximize the loss. Hence, the image is perturbed by a factor of θ to be classified as an unspecified incorrect label. Since this method only requires one iteration, it is the most computationally efficient attack used in our experiments.

B. Basic Iterative Method (BIM)

In order to improve the effectiveness of an adversarial attack, applying perturbations iteratively within given constraints has been a typical solution. By performing multiple steps, the Adversarial Example could cross a decision boundary that FGSM could not. Since BIM is the iterative version of the fast gradient sign method, it takes the gradients of the loss and applies the gradients to the image several times [5]. Here is the objective function:

$$\mathrm{x}_0^{adv} = x,\ x_{i+1}^{adv} = \mathrm{clip}_{x,\epsilon}\{x_i^{adv} + \epsilon * \mathrm{sign}(\Delta_{x_i^{adv}} L(x_i^{adv}, y, \theta))\} \quad (2)$$

Since Adversarial Examples should not deviate from original images such that they are perceptible by a human, x_i^{adv} is either clipped within the boundary of ϵ, and the size of steps is set to be α/T where T is the number of iterations. The iterative method is notably more potent than the one-step FGSM; however, it is relatively costly in computational power. Furthermore, according to the findings of Kurakin et al. [5], BIM has less transferability (ability to generate an Adversarial Example without accessing the underlying model) than FGSM. This drawback could indicate that methods effective for White-Box attacks may perform worse for Black-Box attacks.

C. Momentum Iterative Method (MIM)

Optimization methods typically benefit from utilizing momentum for faster convergence. MIM extends BIM by following this idea. Hence, in order to have a more optimal convergence, MIM takes advantage of the momentum of previous steps with a factor of μ [4]:

$$\begin{aligned} &\alpha = \epsilon/T,\ g_{i+1} = \mu * g_i + \frac{\Delta_{x_i^{adv}} L(x_i^{adv}, y, \theta)}{||\Delta_{x_i^{adv}} L(x_i^{adv}, y, \theta)||_1} \\ &\mathrm{x}_{i+1}^{adv} = x_i^{adv} + \alpha * \mathrm{sign}(g_{i+1}) \end{aligned} \quad (3)$$

Here the step size is set to $\alpha = \epsilon/T$ to meet the L_∞ norm bound $||x^{adv} - x||_\infty < \epsilon$. With the momentum, the gradients have a better chance to pass through poor local optimal and converge faster. Compared with the basic iterative method, MIM overcomes the weakness of overfitting.

D. Projected Gradient Descent (PGD)

Projected Gradient Descent(PGD) is a more general first-order adversarial method for maximizing loss within some constraints [6]. Due to its generality, Madery et al. [6] concluded that any model robust to this adversarial attack ought to be robust against all other first-order adversarial attacks.

$$\begin{aligned} &\alpha = \epsilon/T,\ g_{i+1} = g_i + \alpha * \mathrm{sign}(\Delta_{g_i} L((x + g_i), y, \theta)) \\ &\mathrm{g}_{i+1} = max(min(g_{i+1}, \epsilon), -\epsilon) \\ &\mathrm{x}^{abv} = x + g_T \end{aligned} \quad (4)$$

In PGD, the Adversarial Example is generated after every iteration, though the gradients move along the corresponding directions for each step. Moreover, the gradients are clipped within the specific boundary of ϵ to meet the request of L_∞ norm bound.

We use three variants of this attack for experimentation: seven iterations, twenty iterations (PGD_20), and twenty iterations with twenty restarts (20PGD). The latter performs the twenty iteration attack twenty times rather than increase the number of iterations by a factor of twenty. Ideally, conducting the attack in this manner will be more effective since, due to convergence, running more than twenty iterations will have diminishing returns.

The Adversarial Examples shown here assume the attacker can use the back-propagation gradient information derived from a target model. This assumption does not always hold since a Deep Learning model could be an API that only returns classification labels. Therefore, the attacker must generate an Adversarial Example while treating the target model as a Black-Box API.

E. Black-Box Attack (BB)

The BB attack used in this paper will use surrogate models to create an Adversarial Example. Meaning, the attacker will create their own model and use a White-Box attack on this model to attack the target model. Since it may not be possible for an attacker to access the model (for example, an API that only returns predictions), surrogate models may be the next best option.

A surrogate model is built for each discriminative loss function. Moreover, FGSM and 20 restarts PGD 20 are the conducted attacks. It is possible that an attack on certain loss functions may have better transferability than other loss functions. Also, it may be essential to match the loss function to that of the target model. Furthermore, attacks more potent in the White-Box setting may be more potent in the Black-Box setting. Each of these points is discussed in section VI.

The Adversarial Example attacks mentioned here are able to generate an image that forces an incorrect classification while being imperceptible to a human. Therefore, an image processing method to remove these subtle perturbations is a non-trivial task. The following section will discuss defense methods that take a different approach by altering a model's training process.

IV. Defenses

This section will explain the defenses used for improving a model's Adversarial Robustness. The idea is to make a model exposed to Adversarial Examples during training to learn how to classify Adversarial Examples correctly. The three methods

implemented for our experiments are as follows: Unmodified training, Adversarial Training, and Adversarial Logit Pairing.

A. Unmodified Training (UM)

UM is the general training method where the model is trained only using Clean Examples [6]. For a given underlying data distribution $\mathcal{D}$, over Clean Examples $x \in \mathbb{R}$, which corresponds to the label $y \in [k]$, the aim is to minimize the risk given by:

$$\arg\min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathcal{D}}[L(x, y, \theta)] \tag{5}$$

where L is some suitable loss function such as cross-entropy loss. $\theta \in \mathbb{R}^{\prime}$ represents all the parameters of the model. The goal is to correctly classify the input x to its corresponding label y with minimum loss. The model is not robust against adversarial attacks but tends to have better accuracy on Clean Examples than adversarially robust models.

B. Adversarial Training (AT)

In [16], Athalye et al. conducted an intensive test on the majority of the previously existing Adversarial Robustness algorithms in an attempt to break the underlying method. They found [6] to be the only method that withstood their severe scrutiny. Madry et al. [6] suggest that PGD is a universal first-order adversary; any robust model against a PGD adversary will also be robust against all other first-order adversaries. Their method works by treating the Adversarial Robustness problem as an optimization problem, specifically a saddle point problem, which specifies a quantitative measure of robustness:

$$\arg\min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathcal{D}}[\arg\max_{\delta\in\epsilon} L(x + \delta, y, \theta)] \tag{6}$$

Where $\mathcal{D}$ is the underlying training data distribution, L(θ, x, y) is a loss function, x is a data point with a true class y, θ is the set of model parameters, and δ is the permitted perturbation allowed on an input image x, which is governed by ϵ.

The saddle point problem is a combination of maximization and minimization problems. The inner maximization attempts to find strong Adversarial Examples, and the outer minimization aims to enhance the model's robustness.

C. Adversarial Logit Pairing (ALP)

ALP is built with AT [6] as the underlying basis. Instead of solving the min-max problem in AT, they train their models on a mixture of both Adversarial Examples and Clean Examples to maintain the clean accuracy of the model. They are trained in mini-batches of size N. ALP functions by matching the logits of a clean image and its corresponding adversarial image to be similar to each other:

$$L_{ALP} = \frac{1}{N}\sum_{i}^{N} ||f(x_i) - f(x_i^{adv})||_2 \tag{7}$$

here, $f(x_i)$ and $f(x_i^{adv})$ are functions representing the vector logits of clean image x_i and its adversarial counterpart x_i^{adv}. L^2 loss is the loss function used in the equation. Such logit pairing encourages similar embeddings of the clean and adversarial version of the same example, guiding the model towards better internal data representation.

V. Discriminative Loss Functions

Loss functions are essential for estimating how well a Machine Learning model fits its training data set. For image classification, a loss function needs to estimate how a model classifies multiple labels. This section covers the loss functions tested for Adversarial Robustness. Specifically, we focus on loss functions meant for creating discriminate features in a latent space.

A. Loss Functions

1) Softmax Loss (SM): SM is a multi-label logistic regression cost function used for training Deep Learning Neural Networks. Each label will have a probability value between 0 and 1 that sum to one for a given data point.

$$L_{SM} = -\frac{1}{n}\sum_{i=1}^{n} \log(\frac{e^{W_{y_i}^T f(x_i)+b_{y_i}}}{\sum_{j=1}^{c} e^{W_j^T f(x_i)+b_j}}) \tag{8}$$

For standard classification and regression, this cost function is satisfactory for discriminating vastly different objects. However, the cost function does not enforce any significant decision boundary between labels.

2) Contrastive Loss (ConL): Hadsell et al. formulated this loss for dimensionality reduction via an invariant mapping [10]. In the equation, there are separate partial loss functions given the data label. The equation computes the first if the pair of points are similar, the latter if they are not. Hence, similar points in high-dimensional space are nearby points in the lower dimensional space, while dissimilar points are distant.

$$L_{ConL} = \frac{1}{P}\sum_{i=1}^{P} \begin{cases} ||f(x_i) - f(x_j)||_2, & \text{if } y_i = y_j \\ \max(0, m - ||f(x_i) - f(x_j)||_2), & \text{otherwise} \end{cases} \tag{9}$$

These cost functions are customarily used for unsupervised Deep Learning tasks such as dimensionality reduction or feature embedding. However, effective feature embedding can improve the accuracy of supervised classification methods. Unlike Softmax, this cost function will impose a noticeable distance between differently classified points. Hence, this and the following discriminative loss functions show significantly more Adversarial Robustness.

3) Triplet Loss (TL): Schroff et al. invented this cost function to improve Deep Learning Facial Recognition and achieved superhuman results [9]. The triplets in the cost function are the anchor, positive, and negative. The anchor and the positive are the same labels, while the negative is a different label. Hence, the loss minimization achieves the following: the anchor is near the positive, far from the negative, and the margin between the positive and the negative is at least m.

$$L_{TL} = \frac{1}{T}\sum_{i=1}^{T}[||f(x_i^a) - f(x_i^p)||_2 - ||f(x_i^a) - f(x_i^n)||_2 + m] \tag{10}$$

This discriminative loss function has demonstrable Adversarial Robustness. As evidenced by Moa et al., the discriminative features learned by Triplet Loss showed improved accuracy under the duress of adversarial attacks [8]. Therefore, other discriminative loss functions ought to obtain similar Adversarial Robustness results.

4) Center Loss (CenL): Like Triplet Loss, Wen et al. developed this discriminative loss function for Deep Learning Facial Recognition by minimizing intra-class variance while features of different classes remain separate [11]. The idea is that each c_{y_i} represents the center of deep features for that class; therefore, center loss captures intra-class variations.

$$L_{CenL} = \frac{1}{N}\sum_{i=1}^{N}||f(x_i) - c_{y_i}||_2 \tag{11}$$

Unlike Contrastive Loss and Triplet Loss, Center Loss focuses on intra-class variability and does require comparative data samples and training data recommendations [11]. Hence, the Center Loss is significantly more efficient.

5) Additive Angular Margin Loss (AAML): For Deep Learning Facial Recognition, this discriminative loss function is designed to minimize intra-class compactness and maximize inter-class discrepancy within a feature embedding hypersphere [12]. In a way, this method is a reformulated Softmax function such that predictions only rely on the angle between features and their corresponding weights [12].

$$L_{AAML} = -\frac{1}{N}\sum_{i=1}^{N}\log(\frac{e^{s*\cos(\theta_{y_i}+m)}}{e^{s*\cos(\theta_{y_i}+m)} + \sum_{j\neq y_i}^{C} e^{s*\cos(\theta_j)}}) \tag{12}$$

This discriminative loss function is formulated such that s is the hypersphere's radius and m in the additive angular margin penalty [12]. Hence, this method is scalable for large facial recognition datasets and has achieved state-of-the-art performance. Since this is the highest performing discriminative loss function, it is imperative to test its Adversarial Example robustness.

B. Latent Space Analysis of Adversarial Robustness

Softmax loss only works as a multi-label logistic regression function. Other discriminative loss functions, such as AAML, enforce a strong barrier between the feature embedding of different labels. This property is known as inter-class variability. Moreover, they decrease intra-class variability for feature embeddings of the same label, i.e., feature embeddings of similar points will be compact.

A T-Distributed Stochastic Neighbor Embedding (T-SNE) of the feature embedding latent space will visualize the inter-class variability and intra-class compactness of discriminative loss functions. T-SNE is specifically useful for visualizing high dimensional data, such as feature embeddings, into a lower dimension such that similar data will be modeled by points close to each other while dissimilar data will be modeled by points distant from each other. [17].

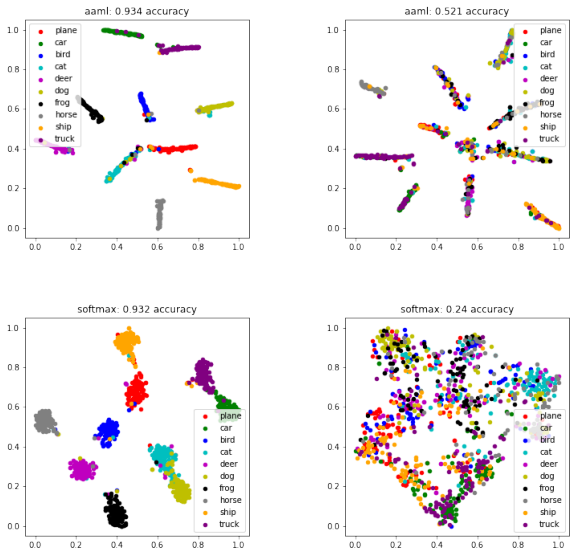

Fig. 1: T-SNE visualization of the second to last 512-dimension feature embedding layer. 1000 samples were randomly selected. Left images are for Clean Examples. Right images are for FGSM Adversarial Examples.

From Figure 1, it is clear that the feature embeddings are more compact and separate between labels for AAML than Softmax. Therefore, it is significantly more challenging for a one-step adversarial attack to succeed due to the more considerable barrier between classification labels. Under an FGSM attack, AAML feature embeddings are still compact, whereas Softmax loses noticeable clusters.

For a visual comparison of the other discriminative loss functions, Figure 2 shows how they react to an FGSM attack. Interestingly enough, noticeable clusters indicated more Adversarial Robustness. This property could indicate that, for an Adversarial Example to be effective, it must be perturbed enough to be within the cluster of another label within the feature embedding space.

VI. Experiment and Results

Results for testing Adversarial Robustness are shown in Table I. An immediate conclusion from these results reinforces the findings from [8]: discriminative loss functions will have significantly more resistance to Adversarial Examples than Softmax while unprotected. Moreover, in the Black-Box setting, Contrastive loss had the most effective attack. This section will cover CIFAR10 Black-Box results, CIFAR10 White-Box results, and the effect of hyperparameters of AAML on Adversarial Robustness. Before that, we will briefly explain the environment, Deep Neural Network, and data used for each experiment.

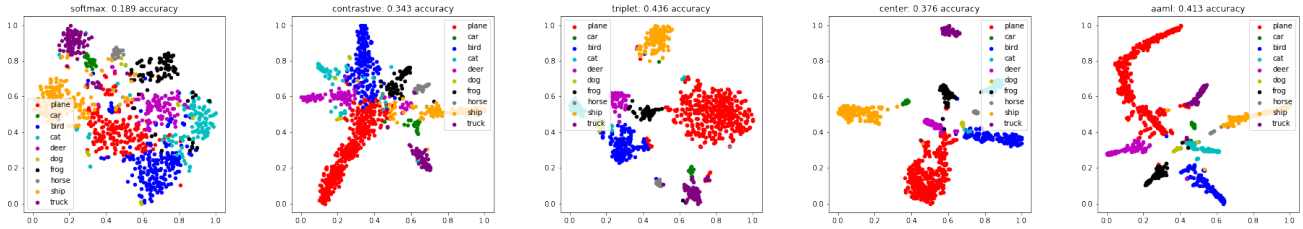

Fig. 2: T-SNE visualization for features whose true label is plane. Each other label is a false negative. Here is how each discriminative loss function reacts to an FGSM attack.

Attacks (Steps)	Clean -	FGSM (1)	BIM (7)	PGD (7)	PGD (20)	20PGD (20)	MIM (40)
SM-UM	93.15	23.95	0.39	0.36	0.04	0.01	0.01
SM-AT	73.90	42.96	36.03	37.31	34.36	33.83	33.96
SM-ALP	72.81	45.55	38.72	40.34	37.14	36.67	36.83
ConL-UM	92.41	44.83	4.03	4.84	1.03	0.52	0.75
ConL-AT	75.44	45.68	38.02	39.20	35.76	35.24	35.38
ConL-ALP	74.60	45.62	38.98	40.33	37.16	36.52	36.84
TL-UM	93.26	48.80	13.48	12.69	5.64	2.11	3.64
TL-AT	74.23	44.77	38.05	39.61	36.28	35.67	35.72
TL-ALP	74.00	45.57	39.31	40.55	**37.57**	**36.99**	**37.11**
CenL-UM	93.25	44.27	10.71	10.09	4.80	1.90	3.13
CenL-AT	80.85	47.50	36.91	38.71	33.32	32.78	32.63
CenL-ALP	78.03	47.69	39.22	40.59	36.26	35.60	35.75
AAML-UM	**93.37**	**52.04**	9.58	9.36	2.46	0.76	1.51
AAML-AT	79.95	51.78	**41.35**	**43.03**	36.68	35.01	35.46
AAML-ALP	76.64	47.07	38.75	40.14	35.33	34.42	34.72

TABLE I: CIFAR10 White-Box attack results. Bold numbers indicate the highest accuracy in its column.

Model	SM	ConL	TL	CenL	AAML
SM-UM	48.13	54.02	56.72	57.10	65.95
SM-AT	72.27	72.15	72.77	72.62	73.11
SM-ALP	72.60	72.74	72.85	72.97	73.60
ConL-UM	49.43	52.72	57.67	58.14	66.98
ConL-AT	73.77	74.04	74.24	74.19	74.76
ConL-ALP	72.94	73.04	73.57	73.41	73.86
TL-UM	51.18	55.09	56.92	57.29	66.86
TL-AT	72.51	72.73	73.12	72.95	73.5
TL-ALP	72.60	72.88	72.94	72.80	73.25
CenL-UM	50.18	55.66	57.09	57.20	67.40
CenL-AT	78.16	78.01	78.34	78.54	79.25
CenL-ALP	75.69	75.85	76.11	76.43	76.60
AAML-UM	48.52	53.76	56.05	56.76	65.88
AAML-AT	76.69	76.59	77.08	77.06	77.82
AAML-ALP	73.85	74.16	74.31	74.14	74.83

TABLE II: FGSM CIFAR 10 Black-Box attack results.

A. environment

The code for our experiments is written in Python 3 with PyTorch as the Machine Learning optimization library. *PyTorch* is an automatic differentiation library designed for rapid research on Machine Learning [18]. This library allowed for implementing our models in a high-level interface that was optimized utilizing a GPU.

The GPU used for training our models was an NVIDIA TESLA V100 16GB. Moreover, the server contained an Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz processor with 125G of RAM. The Neural Network and model parameters used for our experiments are covered in the next section.

Model	SM	ConL	TL	CenL	AAML
SM-UM	0	13.87	22.92	24.16	29.90
SM-AT	71.54	71.28	71.53	71.56	71.64
SM-ALP	71.75	71.64	71.75	71.83	71.91
ConL-UM	28.35	11.56	23.13	27.40	28.75
ConL-AT	72.75	72.54	72.59	72.77	72.93
ConL-ALP	71.47	71.63	71.76	71.69	71.84
TL-UM	27.91	15.45	18.31	21.63	28.20
TL-AT	71.30	71.09	71.42	71.36	71.33
TL-ALP	71.42	71.25	71.40	71.28	71.45
CenL-UM	25.97	16.19	18.07	23.12	27.06
CenL-AT	77.27	77.05	77.40	77.40	77.62
CenL-ALP	74.77	74.79	74.80	74.88	75.05
AAML-UM	24.05	13.19	16.50	21.93	24.19
AAML-AT	75.52	75.65	75.78	75.72	75.98
AAML-ALP	72.75	72.59	72.78	72.83	72.99

TABLE III: 20PGD CIFAR 10 Black-Box attack results.

Margin	S=2	S=4	S=6	S=8
M=0.05	41.46	38.18	38.34	38.42
M=0.10	40.78	39.21	38.38	38.21
M=0.15	40.83	38.95	30.33	30.73
M=0.20	43.03	-	-	-
M=0.25	42.18	-	-	-

TABLE IV: CIFAR10 AAML AT PGD7 results. Each "-" represents an untrainable model.

B. ResNet

Deep Neural Networks typically do not increase in performance with an increase of layers. Issues such as gradient vanishing and overfitting prevent a strict correlation between the number of layers and accuracy. This problem motivated He at. al. to develop an architecture known as ResNet (Residual Neural Network) to alleviate these problems [19]. ResNet introduces skip-connections, i.e., residuals, and has been shown to gain increased accuracy with depth, outperforming the previous VGG architecture [19].

Here, ResNet is used as a feature extractor in which high-level features are embedded in the final layer. From those features, the discriminative loss function is used to train the entire model. Since a CNN architecture requires significantly fewer parameters than fully connected dense layers, it is ideal for extracting features from high dimensional input.

Every model in this experiment uses an 18-layered Resnet, which is optimized by SGD (Stochastic Gradient Descent). Here are the following hyper-parameters used for each model: initial LR (learning rate) of 0.1, reduce LR on plateau scheduler (factor of 0.1, patience of 10 epochs), and an SGD weight decay of 0.0002 with a 0.9 momentum, and a batch size of 256 with 1000 epochs.

C. Data

The dataset used for model training is CIFAR10. Krizhevsky et al. within their research, funded by the Canadian Institute for Advanced Research (CIFAR), created human-labeled datasets of low resolution (32x32) images [20]. CIFAR10 is a balanced 60,000 images dataset with ten labels.

The MNIST hand-written digit dataset was not included due to its overwhelming simplicity. However, the CIFAR dataset includes more unique labels (such as bird, horse, ship, etc.) and is more challenging to train models with high Adversarial Robustness. For example, Mao et al. demonstrated that adversarial attacks on MNIST AT and ALP Softmax models fail to cause sub 93% accuracy [8]. While this dataset shows the Adversarial Robustness of TL over Softmax, it would not suffice in highlighting the differences between other discriminative loss functions.

D. CIFAR10 White-Box Attack Results

There are some expected results from Table I. For example, SM-UM was the most susceptible to every adversarial attack. BIM to MIM resulted in less than 1% accuracy. While some attack methods were slightly more effective, such as 20 restarts PGD at 0.01 and BIM at 0.39, the difference is minimal as the attack success rate is nearly perfect.

The unprotected models for each other discriminative loss function shared some characteristics regarding Adversarial Robustness. Each was relatively robust against FGSM, with AAML the best at 52.04% accuracy. However, BIM to MIM were still effective attacks that resulted in at least sub-14% accuracy. Moreover, 20 restarts PGD had near-perfect results, with TL having the best robustness at 2.11%. Therefore, the only attack that the unprotected models have reasonable robustness to is FGSM.

In most cases, ALP will have better Adversarial Robustness than AT at the cost of worse accuracy on clean images. However, AAML-AT outperforms AAML-ALP in each category. Unlike other discriminative loss functions, AAML fails to improve robustness with ALP. Furthermore, AT and ALP do not significantly affect robustness against FGSM for each model except for SM. For TL and AAML, FGSM robustness was better in the UM case, while CenL and ConL only slightly improved.

Overall, robustness against iterative adversarial attacks will improve with AT and ALP at the cost of model accuracy. Losing over 13% model accuracy for Adversarial Robustness may not be a trade-off worth considering. Moreover, the robustness gained is less than stellar since model accuracy will significantly reduce against White-Box attacks.

Since White-Box attacks are effective regardless of the discriminative loss function, AT, or ALP, it may be worth considering to keep a Deep Learning model within an API that only returns classifications. Here, the attacker will be limited and must resort to using a Black-Box attack. In the next section, we explore the effectiveness of such attacks and how adversarial defense methods mitigate them.

E. CIFAR10 Black-Box Attack Results

While AT and ALP may not be effective against White-Box attacks, they sufficiently mitigate Black-Box attacks. Tables II and III show the transferability of surrogate discriminative loss function models while highlighting the Black-Box adversarial vulnerability of unprotected models.

Without access to the underlying model, to a lesser extent, 20PGD is still effective against unprotected models. Rather than achieving near 0% accuracy, the attack results range from 13.19% to 27.40% accuracy. Although the attack is less effective, it still drastically reduces the accuracy of the target model. Conversely, FGSM did not see a significant performance degradation except for unmodified Softmax, which went from 23.95% accuracy to 48.13% accuracy. Therefore, single-step Adversarial Examples have relatively better transferability than their multi-step counterparts. Nevertheless, the resulting attack is still less effective.

Regarding the transferability of discriminative loss functions, it is clear the ConL outperforms each other loss while AAML performs the worst. What is unexpected about these results is that matching the loss functions to their surrogate models did not increase the effectiveness of the adversarial attack. Intuitively, matching loss functions should have "narrowed the gap" between White-Box and Black-Box attacks. Instead, the effectiveness of each Black-Box attack remained consistent when targeting different models.

Lastly, both adversarial defense techniques, AT and ALP, had remarkable resistance to every Black-Box attack. The resulting accuracy of each attack was similar to that of normal classification accuracy. While there were variations among performance, such as CenL having the highest accuracy, classification models were resistant to Black-Box surrogate model attacks. However, the cost of this adversarial resistance is lower model performance on clean images and only moderate resistance to White-Box attacks.

F. AAML Hyper Parameter Testing

To maximize the Adversarial Robustness of AAML-AT, we ran s and m hyper-parameter tests as shown in Table IV. Each model was trained with the same train/test/validation split, and other hyper-parameters remained constant. Since AAML requires large s and m values for datasets with thousands of labels, such as facial recognition, they had to be scaled down for CIFAR-10. Moreover, increasing s and m beyond the values shown in Table IV resulted in numerical instability during training. Hence, each "-" represents an untrainable model.

Intuitively, increasing the margin parameter ought to increase the model's Adversarial Robustness. Since the margin

is responsible for inter-class variance, the margin should also increase the boundary that an Adversarial Example must cross in order to succeed. However, the margin cannot be too large since it decreases performance and numerical stability. Regarding the radius of the hypersphere, a small *s* was consistently optimal due to the dataset only containing ten labels.

From the results in Table IV, it is clear that s=2 and m=0.2 had the best robustness to PGD7. Thus, those values were used during each other AAML experiment. As shown in Table I, those parameters worked well since AAML-UM performed well and had the highest accuracy on clean images and FGSM Adversarial Examples.

VII. Conclusion

It is clear that discriminative loss functions, including inter-class variability and intra-class compactness, will have more Adversarial Robustness than Softmax. However, even with defense methods such as AT and ALP, White-Box attacks will be effective, just to a lesser extent. Even with the extra robustness, there is decreased model performance for images that are not Adversarial Examples.

Black-Box surrogate model attacks are effective against unprotected models. However, AT and ALP models are effective against these forms of Black-Box attacks. Hence, while the performance trade-off may not be worth it against White-Box attacks, it is worth noting that Black-Box attacks will fail to reduce accuracy beyond the baseline of clean image classification significantly.

The work done here has some clear paths for an extension. For example, using other image datasets, such as Imagenet [21], could be used to confirm that the results in this paper hold for a wider variety of data. Moreover, other Neural Network architectures could be subjected to adversarial attacks to demonstrate that the experiments in this paper are not exclusively applicable to Resnet-18.

For future work, alternative methods for adversarial defenses ought to be explored. Ideally, there should not be a trade-off between model effectiveness and Adversarial Robustness. Possibly, unsupervised learning methods could utilize anomaly detection to prevent Adversarial Examples from reaching an unprotected model. Such a method would not require a model to lose accuracy in order to gain Adversarial Robustness.

Acknowledgement

This work is partially supported by U.S. National Science Foundation grants DGE-2011117 and OAC-1839746 and also the NSF Jetstream [22]/XSEDE [23] project (ACI-1445604 & OCI-1053575).

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "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*, 2015.

[3] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[4] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "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*, 2018.

[7] S. Wu, J. Sang, K. Xu, G. Zheng, and C. Xu, "Adaptive adversarial logits pairing," *CoRR*, 2020.

[8] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray, "Metric learning for adversarial robustness," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 478–489, 2019.

[9] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[10] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1735–1742, 2006.

[11] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, vol. 9911, pp. 499–515, Springer, 2016.

[12] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[13] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks," *ArXiv*.

[14] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 332–349, IEEE, 2019.

[15] M. Pautov, G. Melnikov, E. Kaziakhmedov, K. Kireev, and A. Petiushko, "On adversarial patches: real-world attack on arcface-100 face recognition system," in *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pp. 0391–0396, IEEE, 2019.

[16] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *arXiv preprint arXiv:1802.00420*, 2018.

[17] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[20] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[22] C. Stewart, T. Cockerill, I. Foster, and D. H. et al., "Jetstream: a self-provisioned, scalable sci. and eng. cloud environment," *XSEDE'15 Conf.: Sci. Adv. Enabled by Enhanced Cyberinfra.*, pp. 1–8, 2015.

[23] J. Towns and T. C. et al., "XSEDE: Accelerating scientific discovery," *Computing in Science & Engineering*, vol. 16, no. 5, pp. 62–74.