Improving Adversarial Robustness with Self-Paced Hard-Class Pair Reweighting

Pengyue Hou, Jie Han, Xingyu Li
University of Alberta
pengyue@ualberta.ca, jhan8@ualberta.ca, xingyu@ualberta.ca

Abstract

Deep Neural Networks are vulnerable to adversarial attacks. Among many defense strategies, adversarial training with untargeted attacks is one of the most effective methods. Theoretically, adversarial perturbation in untargeted attacks can be added along arbitrary directions and the predicted labels of untargeted attacks should be unpredictable. However, we find that the naturally imbalanced inter-class semantic similarity makes those hard-class pairs become virtual targets of each other. This study investigates the impact of such closely-coupled classes on adversarial attacks and develops a self-paced reweighting strategy in adversarial training accordingly. Specifically, we propose to upweight hard-class pair losses in model optimization, which prompts learning discriminative features from hard classes. We further incorporate a term to quantify hard-class pair consistency in adversarial training, which greatly boosts model robustness. Extensive experiments show that the proposed adversarial training method achieves superior robustness performance over state-of-the-art defenses against a wide range of adversarial attacks. The code of the proposed SPAT is published at *https://github.com/puerrrr/Self-Paced-Adversarial-Training*.

Introduction

In recent years, DNNs are found to be vulnerable to adversarial attacks, and extensive work has been carried out on how to defend or reject the threat of adversarial samples (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014; Nguyen, Yosinski, and Clune 2015). Adversarial samples are carefully generated with human-imperceptible noises, yet they can lead to large performance degradation of well-trained models.

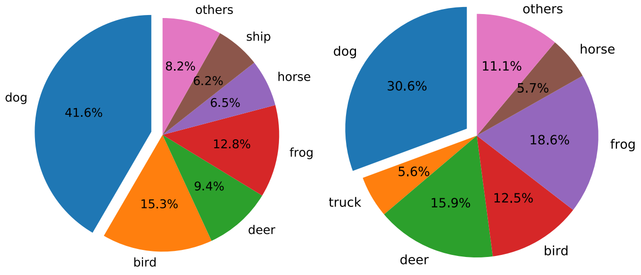
While numerous defenses have been proposed, adversarial training (AT) is a widely recognized strategy (Madry et al. 2017) and achieves promising performance against a variety of attacks. AT treats adversarial attacks as an augmentation method and aims to train models that can correctly classify both adversarial and clean data. Based on the AT framework, further robustness improvements can be achieved by exploiting unlabeled, miss-classified data, pre-training, etc (Alayrac et al. 2019; Carmon et al. 2019; Hendrycks, Lee, and Mazeika 2019; Zhai et al. 2019; Wang et al. 2019; Jiang et al. 2020; Fan et al. 2021; Hou et al. 2022).

In existing adversarial training, untargeted attacks are widely used in model optimization and evaluation (Moosavi-Dezfooli, Fawzi, and Frossard 2016; Madry et al. 2017; Zhang et al. 2019; Wang et al. 2019; Kannan, Kurakin, and Goodfellow 2018; Shafahi et al. 2019; Wong, Rice, and Kolter 2020). Unlike targeted attacks that aim to misguide a model to a particular class other than the true one, untargeted adversaries do not specify the targeted category and perturb the clean data so that its prediction is away from its true label. In theory, adversarial perturbation in untargeted attacks can be added along arbitrary directions and classification of untargeted attacks should be unpredictable. However, the study by Carlini *et al* argues that an untargeted attack is simply a more efficient method of running a targeted attack for each target and taking the *closest* (Carlini and Wagner 2017b). Figure 1 (a) presents the misclassification statistics of PDG-attacked dog images, where almost half of dog images are misclassified as cats, and over 40% of the cat images are misclassified as dogs. Considering that cat and dog images share many common features in vision, we raise the following questions:

"Does the unbalanced inter-class semantic similarity lead to the non-uniformly distributed misclassification statistics? If ***yes****, are classification predictions of untargeted adversaries predictable?"*

To answer these questions, this paper revisits the recipe for generating gradient-based first-order adversaries and surprisingly discovers that untargeted attacks may be targeted. In theory, we prove that adversarial perturbation directions in untargeted attacks are actually biased toward the hard-class pairs of the clean data under attack. Intuitively, semantically-similar classes constitute **hard-class pairs (HCPs)** and semantically-different classes form **easy-class pairs (ECPs)**.

Accordingly, we propose explicitly taking the inter-class semantic similarity into account in AT algorithm design and develop a self-paced adversarial training (SPAT) strategy to upweight hard/easy-class pair losses and downweight easy-class pair losses, encouraging the training procedure to neglect redundant information from easy class pairs. Since HCPs and ECPs may change during model training (depending on the current optimization status), their scaling factors



(a) PDG attacks on vanilla-trained model (b) PDG attacks on SPAT-trained model

Figure 1: Predictions of untargeted adversarial attacks (PGD-20) by CIFAR-10 vanilla-trained and SPAT-trained classifiers. (a) For the vanilla-trained model, over 40% of the dog images are misclassified as cats and (b) it is reduced to 30.6% with the SPAT-trained model.

are adaptively updated at their own pace. Such self-paced reweighting offers SPAT more optimization flexibility. In addition, we further incorporate an HCP-ECP consistency term in SPAT and show its effectiveness in boosting model adversarial robustness. Our main contributions are:

- We investigate the cause of the unevenly distributed misclassification statistics in untargeted attacks. We find that adversarial perturbations are actually biased by targeted sample's hard-class pairs.
- We introduce a SPAT strategy that takes inter-class semantic similarity into account. Adaptively upweighting hard-class pair loss encourages discriminative feature learning.
- We propose incorporating an HCP-ECP consistency regularization term in adversarial training, which boosts model adversarial robustness by a large margin.

Related Work

Adversarial Attack and Defense

The objective of adversarial attacks is to search for human-imperceptible perturbation $\boldsymbol{\delta}$ so that the adversarial sample

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta} \qquad (1)$$

can fool a model $f(\boldsymbol{x};\boldsymbol{\phi})$ well-trained on clean data $\boldsymbol{x}$. Here ϕ represents the trainable parameters in a model. For notation simplification, we use $f(\boldsymbol{x})$ to denote $f(\boldsymbol{x};\boldsymbol{\phi})$ in the rest of the paper. One main branch of adversarial noise generation is the gradient-based method, such as the Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014), and its variants (Kurakin, Goodfellow, and Bengio 2016; Madry et al. 2017). Another popular strategy is optimization based, such as the CW attack (Carlini and Wagner 2017b).

Several pre/post-processing-based methods have shown outstanding performance in adversarial detection and classification tasks (Grosse et al. 2017; Metzen et al. 2017; Xie et al. 2017; Feinman et al. 2017; Li and Li 2017). They aim to use either a secondary neural network or random augmentation methods, such as cropping, compression and blurring to strengthen model robustness. However, Carlini *et al.* showed that they all can be defeated by a tailored attack (Carlini and Wagner 2017a). Adversarial Training, on the other hand, uses regulation methods to directly enhance the robustness of classifiers. Such optimization scheme is often referred to as the ”min-max game”:

$$\underset{\boldsymbol{\phi}}{\text{argmin}}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D}[\max_{\delta\in S}\mathscr{L}(f(\boldsymbol{x}'),\boldsymbol{y})], \qquad (2)$$

where the inner max function aims to generate efficient and strong adversarial perturbation based on a specific loss function $\mathscr{L}$, and the outer min function optimizes the network parameters $\boldsymbol{\phi}$ for model robustness. Another branch of AT aims to achieve *logit level robustness*, where the objective function not only requires correct classification of the adversarial samples, but also encourages the logits of clean and adversarial sample pairs to be similar (Kannan, Kurakin, and Goodfellow 2018; Zhang et al. 2019; Wang et al. 2019). Their AT objective functions usually can be formulated as a compound loss:

$$\mathscr{L}(\boldsymbol{\theta}) = \mathscr{L}_{acc} + \lambda\mathscr{L}_{rob} \qquad (3)$$

where $\mathscr{L}_{acc}$ is usually the cross entropy (CE) loss on clean or adversarial data, $\mathscr{L}_{rob}$ quantifies clean-adversarial logit pairing, and λ is a hyper-parameter to control the relative weights for these two terms. The proposed SPAT in this paper introduces self-paced reweighting mechanisms upon the compound loss and soft-differentiates hard/easy-class pair loss in model optimization for model robustness boost.

Re-weighting in Adversarial Training

Re-weighting is a simple yet effective strategy for addressing biases in machine learning, for instance, class imbalance. When class imbalance exists in the datasets, the training procedure is very likely over-fit to those categories with a larger amount of samples, leading to unsatisfactory performance regarding minority groups. With the re-weighting technique, one can down-weight the loss from majority classes and obtain a balanced learning solution for minority groups.

Re-weighting is also a common technique for hard example mining. Generally, hard examples are those data that have similar representations but belong to different classes. Hard sample mining is a crucial component in deep metric learning (Hoffer and Ailon 2015; Hermans, Beyer, and Leibe 2017) and Contrastive learning (Chen et al. 2020; Khosla et al. 2020). With re-weighting, we can directly utilize the loss information during training and characterize those samples that contribute large losses as hard examples. For example, OHEM (Shrivastava, Gupta, and Girshick 2016) and Focal Loss (Lin et al. 2017) put more weight on the loss of misclassified samples to effectively minimize the impact of easy examples.

Previous studies show that utilizing hard adversarial samples promotes stronger adversarial robustness (Madry et al. 2017; Wang et al. 2019; Mao et al. 2019; Pang et al. 2020). For instance, MART (Wang et al. 2019) explicitly applies a re-weighting factor for misclassified samples by a soft decision scheme. Recently, several re-weighting-based algorithms have also been proposed to address fairness-related issues in AT. (Wang et al. 2021) adopt a re-weighting strategy to address the data imbalance problem in AT and showed that adversarially trained models can suffer much worse performance degradation in under-represented classes. Xu *et al.* (Xu et al. 2021) empirically showed that even in balanced datasets, AT still suffers from the fairness problem, where some classes have much higher performance than others. They propose to combine re-weighting and re-margin for different classes to achieve robust fairness. Zhang *et al.* (Zhang et al. 2020) propose to assign weights based on how difficult to change the prediction of a natural data point to a different class. However, existing AT re-weighting strategies only considered intra-class or inter-sample relationships, but ignored the inter-class biases in model optimization. We propose to explicitly take the inter-class semantic similarity into account in the proposed SPAT strategy and up-weights the loss from hard-class pairs in AT.

Untargeted Adversaries are Targeted

Untargeted adversarial attacks are usually adopt in adversarial training. In theory, adversarial perturbation in untargeted attacks can be added along arbitrary directions, leading to unpredictable false classification. However, our observations on many adversarial attacks contradict this. For example, when untargeted adversaries attack images of cats, the resulting images are likely to be classified as dogs empirically. We visualize image embeddings from the penultimate layer of the vanilla-trained model via t-SNE in Figure 2. In the figure, the embeddings of dog and cat images are close to each other, which suggests the semantic similarity in their representations. With this observation, we hypothesize that the unbalanced inter-class semantic similarity leads to the non-uniformly distributed misclassification statistics.

In this section, we investigate this interesting yet overlooked aspect of adversarial attacks and find that untargeted adversarial examples may be highly biased by their hard-class pairs. The insight in this section directly motivates the proposed self-paced adversarial training for model robustness improvement.

Notations

Given a dataset with labeled pairs $\{\mathscr{X}, \mathscr{Y}\} = \{(x, y) | x \in \mathbb{R}^{c \times m \times n}, y \in [1, C]\}$, a classifier can be formulated as a mapping function $f : \mathscr{X} \to \mathscr{Y}$:

$$f(x) = \mathbb{S}(\boldsymbol{W}^T \boldsymbol{z_x}), \tag{4}$$

where C is the number of categories, and $\mathbb{S}$ represents the softmax function in the classification layer. We use $\boldsymbol{z_x}$ to denote the representation of an input sample x in the penultimate layer of the model and $\boldsymbol{W} = (\boldsymbol{w_i}, \boldsymbol{w_2}, ..., \boldsymbol{w_C})$ for the trainable parameters (including weights and bias) of the softmax layer. Note that $\boldsymbol{w_i}$ can be considered as the prototype of class i and the production $\boldsymbol{W}^T \boldsymbol{z_x}$ in (4) calculates the similarity between $\boldsymbol{z_x}$ and different class-prototype $\boldsymbol{w_i}$. During training, the model f is optimized to minimize a specific loss $\mathscr{L}(f(x), y)$.

In literature, the most commonly used adversarial attacks, such as PGD and its variants, generate adversaries based on first-order derivative information about the network (Madry et al. 2017). Such adversarial perturbations can be generally formulated as follows:

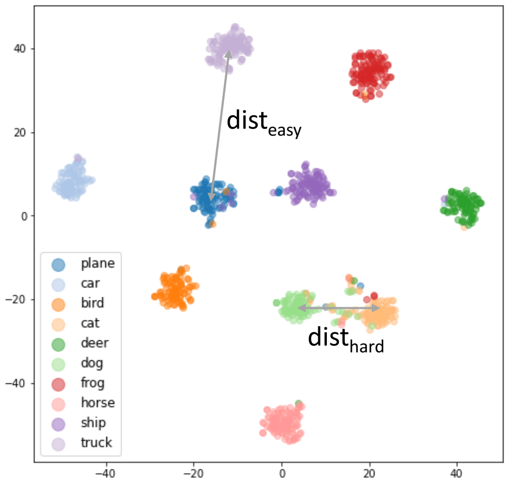


Figure 2: t-SNE visualization of 1000 randomly sampled image embeddings from CIFAR-10. Due to the naturally imbalanced semantic similarity, inter-class distance is much smaller for hard-class pairs.

$$x' = x + \epsilon g(\nabla_{\boldsymbol{x}} \mathscr{L}(f(x), y)), \tag{5}$$

where ϵ is the step size to modify the data and $\nabla_{\boldsymbol{x}}$ is the gradient with respect to the input x. We take g to denote any function on the gradient, for example, $g(x) = \|x\|_p$ is the ℓ_p norm.

Bias in Untargeted Adversarial Attacks

The first-order adversarial attacks usually deploy the CE loss between the prediction $f(x)$ and the target y to calculate adversarial perturbations. The CE loss can be formulated as

$$\mathscr{L}(f(x), y) = -log \frac{e^{\boldsymbol{w_i}^T \boldsymbol{z_x}}}{\sum_{j=1}^{C} e^{\boldsymbol{w_j}^T \boldsymbol{z_x}}} \tag{6}$$

For notation simplification in the rest of this paper, we have $\sigma(\boldsymbol{w_i}^T \boldsymbol{z_x}) = \frac{e^{\boldsymbol{w_i}^T \boldsymbol{z_x}}}{\sum_{j=1}^{C} e^{\boldsymbol{w_j}^T \boldsymbol{z_x}}}$.

Lemma 1 (proof in Appendix): *For an oracle model that predicts the labels perfectly on clean data, the gradient of the CE loss with respect to sample x from the i^{th} category is:*

$$\nabla_{\boldsymbol{x}} \mathscr{L}(f(x), y) = [\sum_{j \neq i}^{C} \sigma(\boldsymbol{w_j}^T \boldsymbol{z_x}) \boldsymbol{w_j}] \nabla_{\boldsymbol{x}} \boldsymbol{z_x}. \tag{7}$$

Lemma 1 indicates that for a clean data x from the i^{th} category, its first-order adversarial update follows the direction of the superposition of all false-class prototypes $\boldsymbol{w_j}$ for $j \in [1, C], j \neq i$. The weight of the j^{th} prototype $\boldsymbol{w_j}$ in the superposition is $\sigma(\boldsymbol{w_j}^T \boldsymbol{z_x})$. The greater the value of the dot product $\sigma(\boldsymbol{w_j}^T\boldsymbol{z_x})$, the more bias in adversarial perturbations toward the i^{th} category. In an extreme case where only one $\sigma(\boldsymbol{w_k}^T\boldsymbol{z_x})$ is non-zero, the untargeted attack becomes a targeted attack.

To investigate if the values of $\sigma(\boldsymbol{w_j}^T\boldsymbol{z_x})$ is equal or not, we let $v_j = \|\boldsymbol{w_j}\|_2$ and $s = \|\boldsymbol{z_x}\|_2$ be the Euclidean norm of the weight and data embedding. Then (7) in Lemma 1 can be rewritten as $\nabla_{\boldsymbol{x}}\mathscr{L}(f(x),y) = [\sum_{j\neq i}^{C}\sigma(v_j s\cos(\boldsymbol{\theta}_j))\boldsymbol{w_j}]\nabla_{\boldsymbol{x}}\boldsymbol{z_x}$, where $\cos(\boldsymbol{\theta}_j)$ measures the angle between the two vectors $\boldsymbol{w_j}$ and $\boldsymbol{x_z}$. Here, we discussed two conditions.

Condition 1 . We regulate $v_j = 1$ and thus convert the CE loss to the normalized cross entropy (NCE) loss in Lemma 1. Recently, many studies show that NCE loss encourages a model to learn more discriminative features (Wang et al. 2018; Liu et al. 2017; Schroff, Kalenichenko, and Philbin 2015). Furthermore, such hypersphere embedding boosts adversarial robustness (Pang et al. 2020). When we follow NCE's regularization and enforce $v_j = 1$, (7) in Lemma 1 is further simplified to

$$\nabla_{\boldsymbol{x}}\mathscr{L}(f(x),y) = [\sum_{j\neq i}^{C}\sigma(s\cos(\boldsymbol{\theta}_j))\boldsymbol{w_j}]\nabla_{\boldsymbol{x}}\boldsymbol{z_x}, \qquad (8)$$

Since $\sigma()$ is a monotonically increasing function, the adversarial update direction is significantly biased by large $\cos(\boldsymbol{\theta}_j)$. It is noteworthy that $s\cos(\boldsymbol{\theta}_j)$ quantifies the projection of a data representation $\boldsymbol{x_z}$ onto the j^{th} class prototype $\boldsymbol{w_j}$, which reflects the inter-class similarity between z_x and a specific false-class prototype. Therefore, this paper defines the false classes associated with a higher $\cos(\boldsymbol{\theta}_j)$ as the **hard-class pairs** of data x; contrastively, the false classes with large $\boldsymbol{\theta}_j$ as the **easy-class pairs**. With this context, we conclude that the adversarial perturbations introduced by the NCE loss are dominated by those **hard** classes with smaller inter-class distances from the true data category.

Condition 2. We relax the condition $v_j = 1$ and extend our discovery to a generic CE loss. Though v_j can be any value in theory, we empirically find that their values are quite stable and even for all j (as shown in Appendix). With these observations, we conclude that untargeted adversaries are actually targeted; Furthermore, the virtual targeted categories are its hard-class pairs.

Figure 3 illustrates a geometric interpretation of our discovery in a simple triplet classification setting, with $y = \{-1, 0, 1\}$. We assume the latent representation of class -1 is closer to class 0 (a hard class pair) and class 1 is farther from class 0 (an easy class pair). Since $cos(\boldsymbol{\theta}_{-1}) > cos(\boldsymbol{\theta}_{+1})$, The attack direction of samples from class 0 is dominated by class -1. Therefore, the data from class 0 is adversarially modified towards class -1.

Self-Paced Adversarial Training

Our discovery in Section motivates the innovation of our re-weighting strategy in the proposed SPAT in twofold.

- From the perspective of learning robust, discriminative features. Compared to adversaries from hard-class pairs having similar semantic representations, easy-class pairs contribute less to model optimization. Encouraging a model to learn HCP samples facilitates the model to extract good features.
- From the perspective of adversarial defense of untargeted attacks. Thanks to the discovered targeted property of untargeted attacks, we know that many clean data are adversarially modified toward their hard-class pairs. With this prior knowledge of untargeted attacks, one can improve models' robustness by learning HCP adversaries in AT.

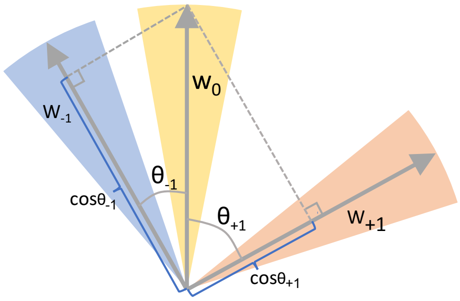


Figure 3: A geometric interpretation of our discovery about untargeted attacks. Different colors represent different classes and W_i is the prototype vector for class i. According to Lemma 1, the overall attack direction for class 0 will be dominated by class -1.

With the above considerations, our self-pace strategy proposes to up-weights training sample's hard-class pair loss in adversarial training.

Specifically, following prior arts in adversarial training, the proposed SPAT strategy adopts a compound loss:

$$\mathscr{L}^{SPAT} = \mathscr{L}^{sp}_{acc} + \lambda\mathscr{L}^{sp}_{rob} \qquad (9)$$

where λ is the trade-off parameter for the accuracy and robustness terms. Notably, we introduces distinct up-weighting policies in $\mathscr{L}^{sp}_{acc}$ and $\lambda\mathscr{L}^{sp}_{rob}$, which encourages the model learning from hard-class pairs.

Self-Paced Accuracy Loss

According to our empirical observations and theoretical analysis in Section 3, untargeted attacks are prone to generate adversaries from hard-class pairs. We argue that a model with stronger HCP discrimination capability would be more robust against adversarial attacks. To this end, we propose up-weighting HCP loss and down-weighting ECP loss in model training to facilitate discriminative representation learning.

As shown in the analysis in Section 3.2, $\cos(\boldsymbol{\theta}_j)$ evaluates the representation similarity between $\boldsymbol{z_x}$ and the prototype vector $\boldsymbol{w_j}$ of the j^{th} class. Ideally, for data from the i^{th} category, we target $\cos(\boldsymbol{\theta}_j) = \delta(i-j)$, where $\delta(x)$ is the Dirichlet identity function. Toward this goal, we monitor the values of $\cos(\boldsymbol{\theta}_j)$ and take them as metrics to adaptively re-weight training samples in adversarial training.

Formally, we propose to reshape the NCE loss by the self-paced modulating factors g^t and g_j^f:

$$\mathscr{L}_{acc}^{sp} = -\log(\frac{e^{g^t \boldsymbol{w_i}^T \boldsymbol{z_x}}}{\sum_{j\neq i}^{C} e^{g_j^f \boldsymbol{w_j}^T \boldsymbol{z_x}} + e^{g^t \boldsymbol{w_i}^T \boldsymbol{z_x}}}), \qquad (10)$$

where $\|\boldsymbol{w_i}\|_2 = 1$ and $\|\boldsymbol{z_x}\|_2 = s$ (Wang et al. 2018). For a sample with true label i, the true-class modulating gain g^t and false-class weights g_j^f are defined as

$$\begin{cases} g^t = 1 - \cos(\boldsymbol{\theta}_i) + \beta \\ g_j^f = \cos(\boldsymbol{\theta}_j) + \beta \end{cases} . \qquad (11)$$

β is a smoothing hyper-parameter to avoid $g^t = 0$ and $g_j^f = 0$. This study adopts the NCE loss, rather than the CE loss, in $\mathscr{L}_{acc}^{sp}$ for the following reasons. NCE is a hypersphere embedding. Compared to the CE loss, the directional embedding encourages a model to learn more discriminative features (Wang et al. 2018; Liu et al. 2017; Schroff, Kalenichenko, and Philbin 2015). Recent study in (Pang et al. 2020) further shows that deploying NCE in adversarial training boosts model robustness against various attacks. It is noteworthy that our ablation study shows that the proposed self-paced modulating mechanism does not only boost model robustness with the NCE loss but also improves model performance with the CE loss.

Intuitively, the introduced self-paced modulating factors amplify the loss contribution from hard-class pairs, and meanwhile down-weight easy-class pair loss. Specifically, according to (11), data from the i^{th} category are associated with large g^t and g_j^f when its representation z_x is far away from its true-class prototype vector $\boldsymbol{w_i}$ while close to a false-class prototype $\boldsymbol{w_i}$. In this scenario, z_x and a false-class prototype vector $\boldsymbol{w_j}$ constitutes a hard-class pair and both g^t and g_j^f amplify the loss in (10), encouraging the model to learn a better representation. On the other hand, when z_x and a false-class prototype vector $\boldsymbol{w_j}$ constitutes an easy-class pair with small $\cos(\boldsymbol{\theta}_j)$, g_j^f is small and thus reduces the ECP contributions to model optimization.

Self-Paced Robustness Loss

The robustness loss term in AT encourages a model to generate the same label to both clean data x and their adversarial samples x'. Intuitively, given a robust representation model, x and x' should share the same hard-class pairs and easy-class pairs. From our analysis in Section 3.2, such an HCP-ECP consistency constraint on x and x' can be formulated as:

$$\cos(\boldsymbol{\theta}_j) \approx \cos(\boldsymbol{\theta'}_j), \forall j. \qquad (12)$$

$\boldsymbol{\theta'}_j$ is the angle between $z_{x'}$ and a prototype vector $\boldsymbol{w_j}$ in the softmax layer of a model.

In prior arts, KL divergence is a widely used as a surrogate robust loss in AT (Wang et al. 2019; Zhang et al. 2019). It quantifies the difference between predicted logits on clean data and its adversarial version:

$$KL(f(x)\|f(x')) = \sum_{i=1}^{C} f_i(x) log \frac{f_i(x)}{f_i(x')}. \qquad (13)$$

Though the KL divergence measures the logit similarity from the point of view of statistics, it doesn't impose the aforementioned HCP-ECP consistency constant in (12) on model optimization.

In this study, we propose a new regularization factor, $L_{inc}(x, x')$, to penalize HCP-ECP inconsistency in model robustness training. With simple math, (12) can be converted into a more intuitive expression: $f_j(x) \approx f_j(x')$ for all j. To accommodate the two inconsistency conditions, $f_j(x) \gg f_j(x')$ and $f_j(x) \ll f_j(x')$), within one formula, we propose the use of $[log \frac{f_j(x)}{f_j(x')}]^2$ to quantify the HCP-ECP inconsistency between x and x with respect to a specific class j. Another benefit of the square operation is its amplification effect on large values, which encourages the model to satisfy the HCP-ECP consistency constraint. Instead of accumulating all inconsistency penalties direction, we follow the statistic perspective of computing KL divergence and the new regularization factor is formulated as

$$L_{inc}^{sp}(x, x') = \sum_{j}^{C} [f_j(x) log \frac{f_j(x)}{f_j(x')}]^2. \qquad (14)$$

Therefore, our new robustness loss is

$$\mathscr{L}_{rob}^{sp} = \alpha KL(f(x)\|f(x')) + L_{inc}^{sp}(x, x'), \qquad (15)$$

where α is a hyper-parameter to balance the two robustness terms.

Experiments

In this section, we first conduct a comprehensive empirical study on the proposed SPAT, providing an in-depth analysis of the method. Then we evaluate SPAT on two popular benchmark datasets, MNIST and CIFAR10, in both white-box and black-box settings. A comparison study with state-of-the-art AT methods is presented.

Breaking Down SPAT

To gain a comprehensive understanding of SPAT, three sets of ablation experiments are conducted: (1) Sensitivity to hyper-parameters, (2)Removing the SP factors in the SPAT loss, and (3) Replacing NCE with CE in SPAT.

Experimental Setup . We use ResNet-18 (He et al. 2016) as our classifier for the CIFAR-10 dataset. Our experimental settings follow prior arts in (Zhang et al. 2019; Wang et al. 2019). All models in this ablation study are trained 100 epochs with SGD and the batch size is 128. The initial learning rate is set as 0.1 and decays by 10 times at 75^{th} and 90^{th} epoch. At the training stage, we use 10-step PGD to generate adversarial samples, with $\epsilon = 8/255$, step size = $\epsilon/4$, and $\lambda = 6$. For evaluation, we apply 20-step PGD to generate attack data, with $\epsilon = 8/255$, step size = $\epsilon/10$. The default hyper-parameter in all experiments are $s = 5$ and $\alpha = \beta = 0.2$, unless otherwise specified.

Sensitivity of Hyper-parameters SPAT has three newly introduced hyper-parameters, s and α in $\mathscr{L}_{acc}^{sp}$ and β in $\mathscr{L}_{rob}^{sp}$. Table 1 presents the sensitivity of these hyper-parameters on CIFAR-10 dataset and shows their impacts

s	Clean	PGD-20
1	**87.57**	49.52
3	86.16	55.77
5	84.26	59.56
8	82.54	60.24
10	81.24	**61.02**

(a) Varying s in $\mathscr{L}^{sp}_{acc}$

α	Clean	PGD-20
0.0	**84.66**	58.32
0.2	84.26	59.56
0.4	83.60	60.11
0.6	83.01	**60.57**

(b) Varying α in $\mathscr{L}^{sp}_{rob}$

β	Clean	PGD-20
0.0	**85.03**	57.88
0.2	84.26	59.56
0.4	83.81	**59.64**
0.6	82.66	57.62

(c) Varying β in $\mathscr{L}^{sp}_{acc}$

Table 1: Hyper-parameter sensitivity in SPAT. If unspecified, the default values are: $s = 5, \alpha = \beta = 0.2$.

loss functions	Clean	PGD-20
$L^{sp}_{nce} + \lambda L^{sp}_{rob}$	**84.26**	**59.56**
$L_{nce} + \lambda L^{sp}_{rob}$	82.49	58.74
$L^{sp}_{nce} + \lambda L_{rob}$	84.01	56.14
$L_{nce} + \lambda L_{rob}$	83.33	54.58

Table 2: Removing SP factors from SPAT.

loss functions	Clean	PGD-20
$L^{sp}_{nce} + \lambda L^{sp}_{rob}$	**84.26**	**59.56**
$L^{sp}_{ce} + \lambda L^{sp}_{rob}$	82.86	53.55
$L_{ce} + \lambda L_{rob}$	82.12	51.82

Table 3: Replacing NCE with CE in SPAT.

defense	Clean	FGSM	PGD-20	C&W
Madry's	99.15	97.22	95.51	95.66
ALP	98.79	97.31	95.85	95.50
TRADES	99.10	97.42	96.22	96.01
MART	98.89	97.70	96.24	96.33
SPAT	**99.21**	**98.12**	**96.64**	**96.57**

Table 4: White box robustness accuracy(%) on MNIST

on model accuracy and robustness. The best performance metrics are highlighted in bold. Similar to NCE(Wang et al. 2018; Pang et al. 2020), the scale factor s in SPAT regulates the length of embeddings. From Table 1a, a larger s leads to higher robustness but lower accuracy. This is because a larger s indicates a larger spherical embedding space and thus samples from different classes can be distributed more discretely. However, the relatively-sparse sample distribution in the large embedding space increases the difficulty of classification. α and β are parameters up-weighting hard-class pair loss in SPAT. As shown in Table 1b and 1c, appropriately choosing α and β can boost model robustness with little accuracy degradation.

Analysis of SP: Table 2 records the performance when removing the proposed self-paced factors in the SPAT loss function. Note, when removing SP weights in the accuracy loss, we let $g^t = g^f_j = 0$ and the proposed self-paced NCE loss becomes the original NCE loss. As indicated in Table 2, removing the SP mechanism from either robustness loss or accuracy loss leads to substantial performance degradation. In particular, the introduced self-paced robustness term encourages the model to follow the HCP/ECP consistency constraint, which contributes to a larger margin of robustness improvement.

Analysis of NCE in SPAT: This study introduces the self-paced modulation factors upon the NCE loss. Table 3 compares model performance when we replace NCE with either the CE loss or a self-paced CE loss (by relaxing normalization $v_j = 1$). The normalization regularization in NCE boosts both model robustness and standard accuracy. In addition, incorporating the self-paced factors into the CE loss also improves model performance. This observation validates our innovation of up-weighting hard-class pair loss in model optimization.

Robustness Evaluation under Different Attacks

In this section, we evaluate the robustness of SPAT on two benchmarks, MNIST and CIFAR10, under various attacks.

Experimental settings: For MNIST, we use a simple 4-layer-CNN followed by three fully connected layers as the classifier. We apply 40-step PGD to generate adversaries in training, with $\epsilon = 0.3$ and step size of 0.01. We train the models for 80 epochs with the learning rate of 0.01. Since MNIST is a simple dataset, three classical attacks, FGSM (Goodfellow, Shlens, and Szegedy 2014), PGD-20 (Madry et al. 2017), and C&W with l_∞ (Carlini and Wagner 2017a), are deployed in our white-box and black-box settings.

On CIFAR-10, adversarial samples used in ATs are generated by 10-step PGD, with $\epsilon = 8/255$ and step size of $\epsilon/4$. The rest training setup is the same as in section . Since CIFAR-10 is a more complex dataset, we further include four stronger attacks in this experiment, which are PGD-100, MIM (Dong et al. 2018), FAB (Croce and Hein 2020a),

defense	Clean	FGSM	PGD-20	C&W
Madry's	99.15	97.06	96.00	96.88
ALP	98.79	97.23	96.13	97.32
TRADES	99.10	97.27	96.88	97.03
MART	98.89	97.68	96.73	97.20
SPAT	**99.21**	**97.80**	**97.27**	**97.40**

Table 5: Black box robustness accuracy(%) on MNIST.

defense	Clean	FGSM	PGD-20	PGD-100	MIM-20	FAB	C&W	AA
Madry's	**84.35**	54.23	46.70	45.73	47.03	47.67	48.62	46.90
TRADES	82.12	56.49	51.82	50.21	51.25	48.21	49.96	47.32
MART	83.08	60.19	54.87	52.97	53.91	48.62	**51.23**	47.87
GAIRAT	83.14	60.03	54.85	52.68	53.44	37.11	40.73	35.90
MAIL-AT	83.80	61.33	55.06	53.26	54.57	45.55	48.67	44.32
SEAT	83.20	61.54	55.86	55.53	57.01	45.70	49.03	47.43
SPAT	84.08	**61.71**	**58.33**	**58.11**	**58.93**	**48.54**	50.60	**48.09**

Table 6: White box robustness accuracy(%) on CIFAR-10 with ResNet-18.

defense	Clean	FGSM	PGD-20	PGD-100	MIM-20	FAB	C&W	AA
Madry's	**84.35**	79.84	80.35	80.91	80.12	81.93	79.98	82.02
TRADES	82.12	79.98	80.69	80.80	80.24	81.71	80.55	81.91
MART	83.08	81.50	82.31	82.89	82.04	83.02	82.97	83.06
GAIRAT	83.14	79.92	80.40	80.61	80.22	82.49	82.43	82.69
MAIL-AT	83.80	81.22	82.16	82.37	81.96	83.10	82.38	83.36
SEAT	83.20	80.44	81.60	82.15	82.33	83.05	81.90	83.10
SPAT	84.08	**82.39**	**83.20**	**83.41**	**82.91**	**83.98**	**84.05**	**84.07**

Table 7: Black box robustness accuracy(%) on CIFAR-10 with ResNet-18.

and AutoAttack (AA) (Croce and Hein 2020b). All attacks are bounded by the l_∞ box with the same maximum perturbation $\epsilon = 8/255$.

Baselines: SOTA defense methods including Madry's (Madry et al. 2017), TRADES (Zhang et al. 2019), MART (Wang et al. 2019), GAIRAT (Zhang et al. 2020), MAIL-AT (Liu et al. 2021) and SEAT (Wang and Wang 2022) are evaluated in this comparison study. We follow the default hyperparameter settings presented in the original papers. For instance, $\lambda = 6$ in TRADES and 5 in MART. For ALP, we set the weight for logit paring as 0.5.

White-Box Robustness: Table. 4 and Table 6 report the white-box robustness performance on MNIST and CIFAR-10, respectively. We omit the standard deviations of 4 runs as they are typically small (< 0.50%), which hardly affects the results. SPAT achieves the highest robustness in all 4 attacks on MNIST and 6 out of 7 on CIFAR-10. The only exception is the l_∞ C&W attack which directly optimizes the difference between correct and incorrect logits (Madry et al. 2017). Notice that the optimization function of the C&W attack (l_∞ version) is the same as the objective function (boosted cross entropy) for MART which makes the rest defense strategies in an unfair position. Even so, SPAT is only 0.67% less robust than MART under the C&W attack. We shown in Appendix that the proposed SPAT also works well with larger models such as WideResNet-34.

Black-Box Robustness: In the black-box attack setting, since adversaries do not access the model architecture and parameters, adversarial samples are crafted on a naturally trained model and transferred to the evaluated models. Here we use a naturally trained LENET-5 (LeCun et al. 1998) and ResNet101 for adversarial sample generation, whose natural accuracy is 98.94% and 95.53% on MNIST and CIFAR-10 respectively.

Table. 5 and Table 7 report the white-box robustness performance on MNIST and CIFAR-10, respectively. Since the features for MNSIT is simple and linear, we notice for certain cases the black box attacks are even stronger than the white box attacks. For example, white box FGSM attacks are weaker than their black box counterpart on all defenses. On the CIFAR10 dataset, while all models reach much higher robustness accuracy compared to white box attacks, SPAT again achieves the top performance. It is worth noting that the weakest attack (FGSM) has the highest black box transferability, while the strongest attack method, AutoAttack, has almost no effect on the SPAT trained model (from 84.08% to 84.07%).

In addition, our experimental results on CIFAR-10C in Appendix suggest that the model trained by SPAT is also robust to natural image corruptions.

Conclusion

In this paper, we studied an intriguing property of untargeted adversarial attacks and concluded that the direction of a first-order gradient-based attack is largely influenced by its hard-class pairs. With this insight, we introduced a self-paced adversarial training strategy and proposed up-weighting hard-class pair loss and down-weighting easy-class pair loss in model optimization. Such an online re-weighting strategy on hard/easy-class pairs encouraged the model to learn more useful knowledge and disregard redundant, easy information.Extensive experiment results show that SPAT can significantly improve the robustness of the model compared to state-of-the-art AT strategies.

References

Alayrac, J.-B.; Uesato, J.; Huang, P.-S.; Fawzi, A.; Stanforth, R.; and Kohli, P. 2019. Are labels required for improving adversarial robustness? *Advances in Neural Information Processing Systems*, 32.

Carlini, N.; and Wagner, D. 2017a. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 3–14.

Carlini, N.; and Wagner, D. 2017b. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. IEEE.

Carmon, Y.; Raghunathan, A.; Schmidt, L.; Duchi, J. C.; and Liang, P. S. 2019. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Croce, F.; and Hein, M. 2020a. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, 2196–2205. PMLR.

Croce, F.; and Hein, M. 2020b. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, 2206–2216. PMLR.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9185–9193.

Fan, L.; Liu, S.; Chen, P.-Y.; Zhang, G.; and Gan, C. 2021. When Does Contrastive Learning Preserve Adversarial Robustness from Pretraining to Finetuning? *Advances in Neural Information Processing Systems*, 34.

Feinman, R.; Curtin, R. R.; Shintre, S.; and Gardner, A. B. 2017. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Grosse, K.; Manoharan, P.; Papernot, N.; Backes, M.; and McDaniel, P. 2017. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; Lee, K.; and Mazeika, M. 2019. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, 2712–2721. PMLR.

Hermans, A.; Beyer, L.; and Leibe, B. 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

Hoffer, E.; and Ailon, N. 2015. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, 84–92. Springer.

Hou, P.; Zhou, M.; Han, J.; Musilk, P.; and Li, X. 2022. Adversarial Fine-tune with Dynamically Regulated Adversary. In *Proceedings of the EEE International Joint Conference on Neural Networks*.

Jiang, Z.; Chen, T.; Chen, T.; and Wang, Z. 2020. Robust pre-training by adversarial contrastive learning. *Advances in Neural Information Processing Systems*, 33: 16199–16210.

Kannan, H.; Kurakin, A.; and Goodfellow, I. 2018. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33: 18661–18673.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, X.; and Li, F. 2017. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE international conference on computer vision*, 5764–5772.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, F.; Han, B.; Liu, T.; Gong, C.; Niu, G.; Zhou, M.; Sugiyama, M.; et al. 2021. Probabilistic margins for instance reweighting in adversarial training. *Advances in Neural Information Processing Systems*, 34: 23258–23269.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212–220.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mao, C.; Zhong, Z.; Yang, J.; Vondrick, C.; and Ray, B. 2019. Metric learning for adversarial robustness. *Advances in Neural Information Processing Systems*, 32.

Metzen, J. H.; Genewein, T.; Fischer, V.; and Bischoff, B. 2017. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 427–436.

Pang, T.; Yang, X.; Dong, Y.; Xu, K.; Zhu, J.; and Su, H. 2020. Boosting adversarial training with hypersphere embedding. *Advances in Neural Information Processing Systems*, 33: 7779–7792.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Shafahi, A.; Najibi, M.; Ghiasi, M. A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 761–769.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Wang, H.; and Wang, Y. 2022. Self-Ensemble Adversarial Training for Improved Robustness. *arXiv preprint arXiv:2203.09678*.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5265–5274.

Wang, W.; Xu, H.; Liu, X.; Li, Y.; Thuraisingham, B.; and Tang, J. 2021. Imbalanced Adversarial Training with Reweighting. *arXiv preprint arXiv:2107.13639*.

Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; and Gu, Q. 2019. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*.

Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*.

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. 2017. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*.

Xu, H.; Liu, X.; Li, Y.; Jain, A.; and Tang, J. 2021. To be robust or to be fair: Towards fairness in adversarial training. In *International Conference on Machine Learning*, 11492–11501. PMLR.

Zhai, R.; Cai, T.; He, D.; Dan, C.; He, K.; Hopcroft, J.; and Wang, L. 2019. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, 7472–7482. PMLR.

Zhang, J.; Zhu, J.; Niu, G.; Han, B.; Sugiyama, M.; and Kankanhalli, M. 2020. Geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2010.01736*.