Adversarial Robustness: From Self-Supervised Pre-Training to Fine-Tuning

Tianlong Chen[1], Sijia Liu[2], Shiyu Chang[2], Yu Cheng[3], Lisa Amini[2], Zhangyang Wang[1]
[1]Texas A&M University, [2]MIT-IBM Watson AI Lab, IBM Research [3]Microsoft Dynamics 365 AI Research
{*wiwjp619,atlaswang*}*@tamu.edu*, {*sijia.liu,shiyu.chang,lisa.amini*}*@ibm.com*, *yu.cheng@microsoft.com*

Abstract

Pretrained models from self-supervision are prevalently used in fine-tuning downstream tasks faster or for better accuracy. However, gaining robustness from pretraining is left unexplored. We introduce adversarial training into self-supervision, to provide general-purpose robust pretrained models for the first time. We find these robust pretrained models can benefit the subsequent fine-tuning in two ways: ***i) boosting final model robustness; ii) saving the computation cost, if proceeding towards adversarial fine-tuning****. We conduct extensive experiments to demonstrate that the proposed framework achieves large performance margins (e.g.,* 3.83% *on robust accuracy and* 1.3% *on standard accuracy, on the CIFAR-10 dataset), compared with the conventional end-to-end adversarial training baseline. Moreover, we find that different self-supervised pretrained models have diverse adversarial vulnerability. It inspires us to ensemble several pretraining tasks, which boosts robustness more. Our ensemble strategy contributes to a further improvement of* 3.59% *on robust accuracy, while maintaining a slightly higher standard accuracy on CIFAR-10. Our codes are available at* `https://github.com/TAMU-VITA/Adv-SS-Pretraining`.

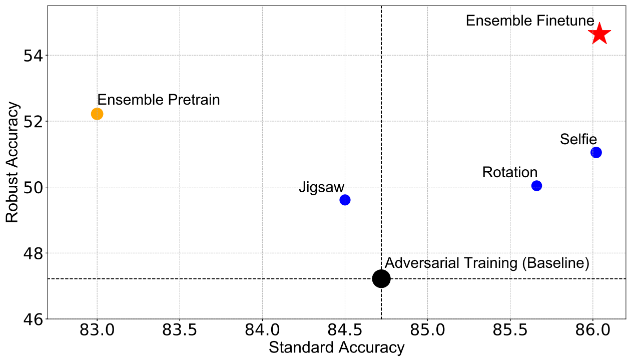

Figure 1: Summary of our achieved performance (CIFAR-10). **The upper right corner indicates the best performance in terms of both standard and robust accuracy.** The *size of markers* represents the number of training epochs to achieve the best robust accuracy. *Black circle* (●) is the baseline method: *end-to-end adversarial training*. *Blue circles* (●) are fine-tuned models that inherit robust models from different self-supervised pretraining tasks. *Orange circle* (●) is the ensemble of three self-supervised pretraining tasks. *Red Star* (★) is the ensemble of three fine-tuned models. The correspondence between the marker size and # epochs is given by, *Ensemble Fine-tune* (★, 144 epochs) > *Baseline* (●, 99 epochs) > *Ensemble Pretrain* (●, 56 epochs) > *Selfie* (●, 50 epochs) > *Jigsaw* (●, 48 epochs) > *Rotation* (●, 46 epochs)

1. Introduction

Supervised training of deep neural networks requires massive, labeled datasets, which may be unavailable and costly to assemble [15, 2, 28, 36]. Self-supervised and unsupervised training techniques attempt to address this challenge by eliminating the need for manually labeled data. Representations pretrained through self-supervised techniques enable fast fine-tuning to multiple downstream tasks, and lead to better generalization and calibration [20, 23]. Examples of tasks proven to attain high accuracy through self-supervised pretraining include position predicting tasks (*Selfie* [35], *Jigsaw* [25, 3]), rotation predicting tasks (*Rotation* [9]), and a variety of other perception tasks [6, 41, 8].

The labeling and sample efficiency challenges of deep learning are further exacerbated by vulnerability to adversarial attacks. For example, Convolutional Neural Networks (CNNs), are widely leveraged for perception tasks, due to high predictive accuracy. However, even a well-trained CNN suffers from high misclassification rates when imperceivable perturbations are applied the input [18, 24]. As suggested by [30], the sample complexity of learning an adversarially robust model with current methods is significantly higher than that of standard learning. Adversarial training (AT) [21], the state-of-the-art model defense approach, is also known to be computationally more expensive than standard training (ST). The above facts make it especially meaningful to explore:

> *Can appropriately pretrained models play* ***a similar role*** *for adversarial training as they have for ST? That is, can they lead to more efficient fine-tuning and better, adversarially-robust generalization?*

Self-supervision has only recently been linked to the study of robustness. An approach is offered in [14], by incorporating the self-supervised task as a complementary objective, which is co-optimized with the conventional classification loss through the method of AT [21]. Their co-optimization approach presents scalability challenges, and does not enjoy the benefits of pretrained embeddings. Further, it leaves many unanswered questions, especially with respect to efficient tuning, which we tackle in this paper.

Contributions. This paper introduces a framework for self-supervised pretraining and fine-tuning into the adversarial robustness field. We motivate our study with the following three scientific questions:

Q1: *Is an adversarially pretrained model effective in boosting the robustness of subsequent fine-tuning?*

Q2: *Which provides the better accuracy and efficiency: adversarial pretraining or adversarial fine-tuning?*

Q3: *How does the type of self-supervised pretraining task affect the final model's robustness?*

Our contributions address the above questions and can be summarized as follows:

A1: We demonstrate for the first time that robust pretrained models leveraged for adversarial fine-tuning result in a **large performance gain**. As illustrated by Figure 1, the best pretrained model from a single self-supervised task (*Selfie*) leads to **3.83%** on robust accuracy[1] and **1.3%** on standard accuracy on CIFAR-10 when being adversarially fine-tuned, compared with the strong AT baseline. Even performing standard fine-tuning (which consumes fewer resources) with the robust pretrained models improves the resulting model's robustness.

A2: We systematically study all possible combinations between pretraining and fine-tuning. Our extensive results reveal that adversarial fine-tuning contributes to the dominant portion of **robustness improvement**, while robust pretraining mainly **speeds up** adversarial fine-tuning. That can also be read from Figure 1 (smaller marker sizes denote less training epochs needed).

A3: We experimentally show that the pretrained models resulting from different self-supervised tasks have diverse adversarial vulnerabilities. In view of that, we propose to pretrain with an ensemble of self-supervised tasks, in order to leverage their complementary strengths. On CIFAR-10, our ensemble strategy further contributes to an improvement of 3.59% on robust accuracy, while maintaining a slightly higher standard accuracy. Our approach establishes a **new benchmark result** on standard accuracy (86.04%) and robust accuracy (54.64%) in the setting of AT.

2. Related Work

Self-supervised pretraining. Numerous self-supervised learning methods have been developed in recent years, including: region/component filling (*e.g. inpainting* [6] and *colorization* [41]); rotation prediction [9]; category prediction [8]; and patch-base spatial composition prediction (*e.g.*, *Jigsaw* [25, 3] and *Selfie* [35]). All perform standard training, and do not tackle adversarial robustness. For example, *Selfie* [35], generalizes BERT to image domains. It masks out a few patches in an image, and then attempts to classify a right patch to reconstruct the original image. Selfie is first pretrained on unlabeled data and fine-tuned towards the downstream classification task.

Adversarial robustness. Many defense methods have been proposed to improve model robustness against adversarial attacks. Approaches range from adding stochasticity [7], to label smoothening and feature squeezing [27, 38], to denoising and training on adversarial examples [22, 19]. A handful of recent works point out that those empirical defenses could still be easily compromised [1]. Adversarial training (AT) [21] provides one of the strongest current defenses, by training the model over the adversarially perturbed training data, and has not yet been fully compromised by new attacks. [10, 16] showed AT is also effective in compressing or accelerating models [42] while preserving learned robustness.

Several works have demonstrated model ensembles [32, 34] to boost adversarial robustness, as the ensemble diversity can challenge the transferability of adversarial examples. Recent proposals [26, 37] formulate the diversity as a training regularizer for improved ensemble defense. Their success inspires our ensembled self-supervised pretraining.

Unlabeled data for adversarial robustness. Self-supervised training learns effective representations for improving performance on downstream tasks, without requiring labels. Because robust training methods have higher sample complexity, there has been significant recent attention on how to effectively utilize unlabeled data to train robust models.

Results show that unlabeled data can become a competitive alternative to labeled data for training adversarially robust models. These results are concurred by [39], who also finds that learning with more unlabeled data can result in better adversarially robust generalization. Both works [31, 4] use unlabeled data to form an *unsupervised* auxiliary loss (*e.g.*, a label-independent robust regularizer or a pseudo-label loss).

[1] Throughout this paper, we follow [40] to adopt their defined standard accuracy and robust accuracy, as two metrics to evaluate our method's effectiveness: a desired model shall be high in both.

To the best of our knowledge, [14] is the only work so far that utilizes unlabeled data via *self-supervision* to train a robust model given a target supervised classification task. It improves AT by leveraging the rotation prediction self-supervision as an auxiliary task, which is co-optimized with the conventional AT loss. Our self-supervised pretraining and fine-tuning differ from all above settings.

3. Our Proposal

In this section, we introduce *self-supervised pretraining* to learn feature representations from unlabeled data, followed by *fine-tuning* on a target supervised task. We then generalize adversarial training (AT) to different self-supervised pretraining and fine-tuning schemes.

3.1. Setup

Self-Supervised Pretraining Let $\mathcal{T}_{\mathrm{p}}$ denote a pretraining task and $\mathcal{D}_{\mathrm{p}}$ denote the corresponding (unlabeled) pretraining dataset. The goal of self-supervised pretraining is to learn a model from $\mathcal{D}_{\mathrm{p}}$ itself without explicit manual supervision. This is often cast as an optimization problem, in which a proposed pretraining loss $\ell_{\mathrm{p}}(\boldsymbol{\theta}_{\mathrm{p}}, \boldsymbol{\theta}_{\mathrm{pc}}; \mathcal{D}_{\mathrm{p}})$ is minimized to determine a model parameterized by $\boldsymbol{\theta}_{\mathrm{p}}$. Here $\boldsymbol{\theta}_{\mathrm{pc}}$ signifies additional parameters *customized* for a given $\mathcal{T}_{\mathrm{p}}$. In the rest of the paper, we focus on the following self-supervised pretraining tasks (details on each pretraining task are provided in the supplement):

Selfie [35]: By masking out select patches in an image, *Selfie* constructs a classification problem to determine the correct patch to be filled in the masked location.

Rotation [9]: By rotating an image by a random multiple of 90 degrees, *Rotation* constructs a classification problem to determine the degree of rotation applied to an input image.

Jigsaw [25, 3]: By dividing an image into different patches, *Jigsaw* trains a classifier to predict the correct permutation of these patches.

Supervised Fine-tuning Let $\mathbf{r}(\mathbf{x}; \boldsymbol{\theta}_{\mathrm{p}})$ denote the mapping (parameterized by $\boldsymbol{\theta}_{\mathrm{p}}$) from an input sample $\mathbf{x}$ to its embedding space learnt from the self-supervised pretraining task $\mathcal{T}_{\mathrm{p}}$. Given a target finetuning task $\mathcal{T}_{\mathrm{f}}$ with the labeled dataset $\mathcal{D}_{\mathrm{f}}$, the goal of fine-tuning is to determine a classifier, parameterized by $\boldsymbol{\theta}_{\mathrm{f}}$, which maps the represetnation $\mathbf{r}(\mathbf{x}; \boldsymbol{\theta}_{\mathrm{p}})$ to the *label* space. To learn the classifier, one can minimize a common supervised training loss $\ell_{\mathrm{f}}(\boldsymbol{\theta}_{\mathrm{p}}, \boldsymbol{\theta}_{\mathrm{f}}; \mathcal{D}_{\mathrm{f}})$ with a *fixed* or *re-trainable* model $\boldsymbol{\theta}_{\mathrm{p}}$, corresponding to *partial fine-tuning* and *full fine-tuning*, respectively.

AT versus standard training (ST) AT is known as one of the most powerful methods to train a robust classifier against adversarial attacks [21, 1]. Considering an ϵ-tolerant ℓ_∞ attack $\boldsymbol{\delta}$ subject to $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$, an adversarial example of a benign input $\mathbf{x}$ is given by $\mathbf{x} + \boldsymbol{\delta}$. With the aid of adversarial examples, AT solves a min-max optimization problem of the generic form

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \ \mathbb{E}_{\mathbf{x} \in \mathcal{D}} \left[\underset{\|\boldsymbol{\delta}\|_\infty \leq \epsilon}{\text{maximize}} \ \ell(\boldsymbol{\theta}, \mathbf{x} + \boldsymbol{\delta}) \right], \qquad (1)$$

where $\boldsymbol{\theta}$ denotes the parameters of an ML/DL model, $\mathcal{D}$ is a given dataset, and ℓ signifies a classification loss evaluated at the model $\boldsymbol{\theta}$ and the perturbed input $\mathbf{x} + \boldsymbol{\delta}$. By fixing $\boldsymbol{\delta} = 0$, problem (1) then simplifies to the ST framework $\text{minimize}_{\boldsymbol{\theta}} \ \mathbb{E}_{\mathbf{x} \in \mathcal{D}} \left[\ell(\boldsymbol{\theta}, \mathbf{x}) \right]$.

3.2. AT meets self-supervised pretraining and fine-tuning

AT given by (1) can be specified for either *self-supervised pretraining* or *supervised fine-tuning*. For example, AT for self-supervised pretraining can be cast as problem (1) by letting $\boldsymbol{\theta} := [\boldsymbol{\theta}_{\mathrm{p}}^T, \boldsymbol{\theta}_{\mathrm{pc}}^T]^T$ and $\mathcal{D} := \mathcal{D}_{\mathrm{p}}$, and specifying ℓ as ℓ_{p}. In Table 1, we summarize all the possible scenarios when AT meets self-supervised pretraining.

Table 1: Summary of self-supervised pretraining scenarios.

Scenario	Pretraining method	Loss ℓ in (1)	Variables $\boldsymbol{\theta}$ in (1)	dataset $\mathcal{D}$ in (1)
$\mathcal{P}_1$	None[1]	NA[2]	NA	NA
$\mathcal{P}_2$	ST[3]	ℓ_{p}	$[\boldsymbol{\theta}_{\mathrm{p}}^T, \boldsymbol{\theta}_{\mathrm{pc}}^T]^T$	$\mathcal{D}_{\mathrm{p}}$
$\mathcal{P}_3$	AT	ℓ_{p}	$[\boldsymbol{\theta}_{\mathrm{p}}^T, \boldsymbol{\theta}_{\mathrm{pc}}^T]^T$	$\mathcal{D}_{\mathrm{p}}$

[1] None: the model form of $\boldsymbol{\theta}_{\mathrm{p}}$ is known in advance.

[2] NA: Not applicable.

[3] ST: A special case of (1) with $\boldsymbol{\delta} = \mathbf{0}$.

Table 2: Summary of fine-tuning scenarios.

Scenario	Fine-tuning type	Fine-tuning method	Loss ℓ in (1)	Variables $\boldsymbol{\theta}$ in (1)	dataset $\mathcal{D}$ in (1)
$\mathcal{F}_1$	Partial (with *fixed* $\boldsymbol{\theta}_{\mathrm{p}}$)[1]	ST	ℓ_{f}	$\boldsymbol{\theta}_{\mathrm{f}}$	$\mathcal{D}_{\mathrm{f}}$
$\mathcal{F}_2$	Partial (with *fixed* $\boldsymbol{\theta}_{\mathrm{p}}$)	AT	ℓ_{f}	$\boldsymbol{\theta}_{\mathrm{f}}$	$\mathcal{D}_{\mathrm{f}}$
$\mathcal{F}_3$	Full[2]	ST	ℓ_{f}	$[\boldsymbol{\theta}_{\mathrm{p}}^T, \boldsymbol{\theta}_{\mathrm{f}}^T]^T$	$\mathcal{D}_{\mathrm{f}}$
$\mathcal{F}_4$	Full	AT	ℓ_{f}	$[\boldsymbol{\theta}_{\mathrm{p}}^T, \boldsymbol{\theta}_{\mathrm{f}}^T]^T$	$\mathcal{D}_{\mathrm{f}}$

[1] *Fixed* $\boldsymbol{\theta}_{\mathrm{p}}^*$ signifies the model learnt in a given pretraining scenario.

[2] Full fine-tuning retrains $\boldsymbol{\theta}_{\mathrm{p}}$.

Given a pretrained model $\boldsymbol{\theta}_{\mathrm{p}}$, adversarial fine-tuning could have two forms: a) AT for partial fine-tuning and b) AT for full fine-tuning. Here the former case a) solves a supervised fine-tuning task under the fixed model ($\boldsymbol{\theta}_{\mathrm{p}}$), and the latter case b) solves a supervised fine-tuning task by retraining $\boldsymbol{\theta}_{\mathrm{p}}$. In Table 2, we summarize different scenarios when AT meets supervised fine-tuning.

It is worth noting that our study on the integration of AT with a pretraining+fine-tuning scheme $(\mathcal{P}_i, \mathcal{F}_j)$ provided by Tables 1-2 is different from [14], which conducted one-shot AT over a supervised classification task integrated with a rotation self-supervision task.

In order to explore the network robustness against different configurations $\{(\mathcal{P}_i, \mathcal{F}_j)\}$, **we ask**: *is AT for robust pretraining sufficient to boost the adversarial robustness of fine-tuning? What is the influence of fine-tuning strategies (partial or full) on the adversarial robustness of image classification? How does the type of self-supervised pretraining task affect the classifier's robustness?*

We provide detailed answers to the above questions in Sec. 4.3, Sec. 4.4 and Sec. 4.5. In a nutshell, we find that robust representation learnt from adversarial pretraining is transferable to down-stream fine-tuning tasks to some extent. However, a more significant robustness improvement is obtained by adversarial fine-tuning. Moreover, AT for full fine-tuning outperforms that for partial fine-tuning in terms of both robust accuracy and standard accuracy (except the *Jigsaw*-specified self-supervision task). Furthermore, different self-supervised tasks demonstrate diverse adversarial vulnerability. As will be evident later, such diversified tasks provide complementary benefits to model robustness and therefore can be combined.

3.3. AT by leveraging ensemble of multiple self-supervised learning tasks

In what follows, we generalize AT to learn a robust pretrained model by leveraging the diversified pretraining tasks. More specifically, consider M self-supervised pretraining tasks $\{\mathcal{T}_{\mathrm{p}}^{(i)}\}_{i=1}^{M}$, each of which obeys the formulation in Section 3.1. We generalize problem (1) to

$$\underset{\boldsymbol{\theta}_r, \{\boldsymbol{\theta}_{\mathrm{pc}}^{(i)}\}}{\text{minimize}} \; \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_p} \left[\mathcal{L}_{\mathrm{adv}}(\boldsymbol{\theta}_{\mathrm{p}}, \{\boldsymbol{\theta}_{\mathrm{pc}}^{(i)}\}, \mathbf{x}) \right], \tag{2}$$

where $\mathcal{L}_{\mathrm{adv}}$ denotes the adversarial loss given by

$$\begin{aligned} &\mathcal{L}_{\mathrm{adv}}(\boldsymbol{\theta}_{\mathrm{p}}, \{\boldsymbol{\theta}_{\mathrm{pc}}^{(i)}\}, \mathbf{x}) \\ := &\underset{\{\|\boldsymbol{\delta}^{(i)}\|_\infty \leq \epsilon\}}{\text{maximize}} \sum_{i=1}^{M} \ell_{\mathrm{p}}^{(i)}(\boldsymbol{\theta}_{\mathrm{p}}, \boldsymbol{\theta}_{\mathrm{pc}}^{(i)}, \mathbf{x} + \boldsymbol{\delta}^{(i)}) \\ &\qquad + \lambda g(\boldsymbol{\theta}_{\mathrm{p}}, \{\boldsymbol{\theta}_{\mathrm{pc}}^{(i)}\}, \{\boldsymbol{\delta}^{(i)}\}). \end{aligned} \tag{3}$$

In (2), for ease of notation, we replace $\{\cdot\}_{i=1}^{M}$ with $\{\cdot\}$, $\boldsymbol{\theta}_{\mathrm{p}}$ denotes the common network shared among different self-supervised tasks, and $\boldsymbol{\theta}_{\mathrm{pc}}^{(i)}$ denotes a sub-network customized for the ith task. We refer readers to Figure 2 for an overview of our proposed model architecture. In (3), $\ell_{\mathrm{p}}^{(i)}$ denotes the ith pretraining loss, g denotes a *diversity-promoting regularizer*, and $\lambda \geq 0$ is a regularization parameter. Note that $\lambda = 0$ gives the averaging ensemble strategy. In our case, we perform grid search to tune λ around the value chosen in [26]. Details are referred to the supplement.

Spurred by [26, 37], we quantify the diversity-promoting regularizer g through the orthogonality of input gradients of different self-supervised pretraining losses,

$$g(\boldsymbol{\theta}_{\mathrm{p}}, \{\boldsymbol{\theta}_{\mathrm{pc}}^{(i)}\}, \{\boldsymbol{\delta}^{(i)}\}) := \log\det(\mathbf{G}^T\mathbf{G}), \tag{4}$$

where each column of $\mathbf{G}$ corresponds to a *normalized* input gradient $\{\nabla_{\boldsymbol{\delta}_i} \ell_p^{(i)}(\boldsymbol{\theta}_{\mathrm{p}}, \boldsymbol{\theta}_{\mathrm{pc}}^{(i)}, \mathbf{x} + \boldsymbol{\delta}^{(i)})\}$, and g reaches the maximum value 0 as input gradients become orthogonal, otherwise it is negative. The rationale behind the diversity-promoting adversarial loss (3) is that we aim to design a robust model $\boldsymbol{\theta}_{\mathrm{p}}$ by defending attacks from diversified perturbation directions.

4. Experiments and Results

In this section, we design and conduct extensive experiments to examine the network robustness against different configurations $\{(\mathcal{P}_i, \mathcal{F}_j)\}$ for image classification. *First*, we show adversarial self-supervised pretraining (namely, $\mathcal{P}_3$ in Table 1) improves the performance of downstream tasks. We also discuss the influence of different fine-tuning strategies $\mathcal{F}_j$ on the adversarial robustness. *Second*, we show the diverse impacts of different self-supervised tasks on their resulting pretrained models. *Third*, we ensemble those self-supervised tasks to perform adversarial pretraining. At the fine-tuning phase, we also ensemble three best models with the configuration ($\mathcal{P}_3$, $\mathcal{F}_4$) and show its performance superiority. *Last*, we report extensive ablation studies to reveal the influence of the size of the datasets $\mathcal{D}_{\mathrm{p}}$ and the resolution of images in $\mathcal{D}_{\mathrm{p}}$, as well as other defense options beyond AT.

4.1. Datasets

Dataset Details We consider four different datasets in our experiments: CIFAR-10, CIFAR-10-C [13], CIFAR-100 and **R-ImageNet-224** (a specifically constructed "restricted" version of ImageNet, with resolution 224 × 224). For the last one, we indeed to demonstrate our approach on high-resolution data despite the computational challenge. We follow [29] to choose 10 super classes which contain a total of 190 ImageNet classes. The detailed classes distribution of each super class can be found in our supplement.

For the ablation study of different pretraining dataset sizes, we sample more training images from the 80 Million Tiny Images dataset [33] where CIFAR-10 was selected from. Using the same 10 super classes, we form CIFAR-30K (*i.e.*, 30,000 for images), CIFAR-50K, CIFAR-150K for training, and keep another 10,000 images for hold-out testing.

Dataset Usage In Sec. 4.3, Sec. 4.4 and Sec. 4.5, for all results, we use CIFAR-10 training set for both pretraining and fine-tuning. We evaluate our models on the CIFAR-10 testing set and CIFAR-10-C. In Sec. 4.6, we use CIFAR-10, CIFAR-30K, CIFAR-50K, CIFAR-150K and R-ImageNet-224 for pretraining, and CIFAR-10 training set for fine-tuning, while evaluating on CIFAR-10 testing set. We also validate our approaches on CIFAR-100 in the supplement. In all of our experiments, we randomly split the original training set into a training set and a validation set (the ratio is 9:1).

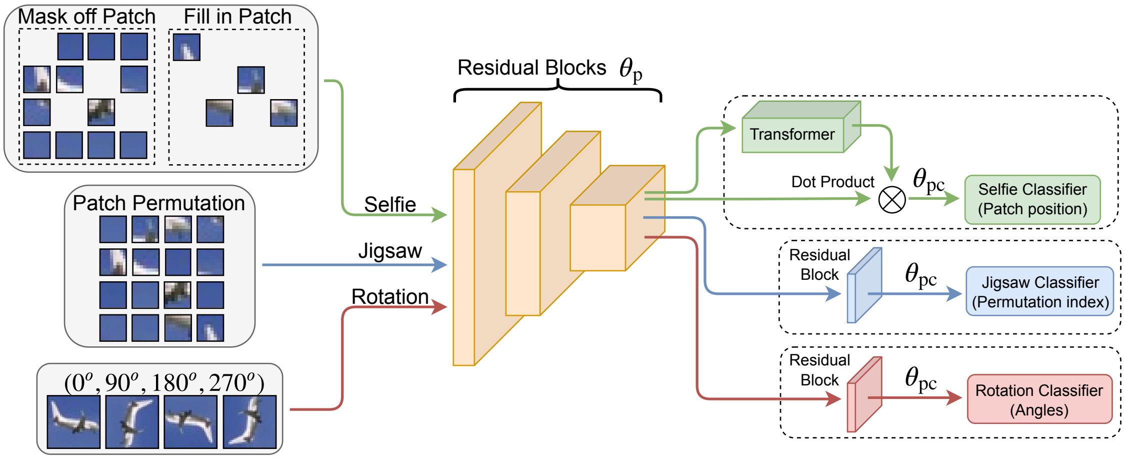

Figure 2: The overall framework of ensemble adversarial pretraining. The pretrained weights θ_{p} are the first three blocks of ResNet-50v2 [11]; *Green arrows* (➔), *Blue arrows* (➔) and *Red arrows* (➔) represent the feed forward paths of *Selfie*, *Jigsaw* and *Rotation*, respectively.

4.2. Implementation Details

Model Architecture: For pretraining with the *Selfie* task, we identically follow the setting in [35]. For *Rotation* and *Jigsaw* pretraining tasks, we use ResNet-50v2 [12]. For the fine-tuning, we use ResNet-50v2 for all. Each fine-tuning network will inherit the corresponding robust pretrained weights to initialize the first three blocks of ResNet-50v2, while leaving the remaining blocks randomly initialized.

Training & Evaluation Details: All pretraining and fine-tuning tasks are trained using SGD with 0.9 momentum. We use batch sizes of 256 for CIFAR-10, ImageNet-32 and 64 for R-ImageNet-224. All pretraining tasks adopt cosine learning rates. The maximum and minimum learning rates are 0.1 and 10^{-6} for *Rotation* and *Jigsaw* pretraining; 0.025 and 10^{-6} for *Selfie* pretraining; and 0.001 and 10^{-8} for ensemble pretraining. All fine-tuning phases follow a multi-step learning rate schedule, starting from 0.1 and decayed by 10 times at epochs 30 and 50 for a 100 epochs training.

We use 10-step and 20-step ℓ_∞ PGD attacks [21] for adversarial training and evaluation, respectively. Unless otherwise specified, we follow [14]'s setting with $\epsilon = \frac{8.0}{225}$ and $\alpha = \frac{2.0}{255}$. For all adversarial evaluations, we use the full testing datasets ($i.e$ $10,000$ images for CIFAR-10) to generate adversarial images. We also consider unforeseen attacks [17, 13].

Evaluation Metrics & Model Picking Criteria: We follow [40] to use: i) *Standard Testing Accuracy (TA)*: the classification accuracy on the clean test dataset; II) *Robust Testing Accuracy (RA)*: the classification accuracy on the attacked test dataset. In our experiments, we use TA to pick models for a better trade-off with RA. Results of models picked using RA criterion are included in the supplement.

4.3. Adversarial self-supervised pertraining & fine-tuning helps classification robustness

We systematically study all possible configurations of pretraining and fine-tuning considered in Table 1 and Table 2, where recall that the expression $(\mathcal{P}_i, \mathcal{F}_j)$ denotes a specified pretraining+fine-tuning scheme. The *baseline* schemes are given by the end-to-end standard training (ST), namely, $(\mathcal{P}_1, \mathcal{F}_3)$ and the end-to-end adversarial training (AT), namely, $(\mathcal{P}_1, \mathcal{F}_4)$. Table 3 shows TA, RA, and iteration complexity of fine-tuning (in terms of number of epochs) under different pretraining+fine-tuning strategies involving different self-supervised pretraining tasks, *Selfie*, *Rotation* and *Jigsaw*. In what follows, we analyze the results of Table 3 and provide additional insights.

We begin by focusing on the scenario of integrating the *standard pretraining* strategy $\mathcal{P}_2$ with fine-tuning schemes $\mathcal{F}_3$ and $\mathcal{F}_4$ used in baseline methods. Several observations can be made from the comparison $(\mathcal{P}_2, \mathcal{F}_3)$ vs. $(\mathcal{P}_1, \mathcal{F}_3)$ and $(\mathcal{P}_2, \mathcal{F}_4)$ vs. $(\mathcal{P}_1, \mathcal{F}_4)$ in Table 3. 1) The use of self-supervised pretraining consistently improves TA and/or RA even if only standard pretraining is conducted; 2) The use of adversarial fine-tuning $\mathcal{F}_4$ (against standard fine-tuning $\mathcal{F}_3$) is crucial, leading to significantly improved RA under both $\mathcal{P}_1$ and $\mathcal{P}_2$; 3) Compared $(\mathcal{P}_1, \mathcal{F}_4)$ with $(\mathcal{P}_2, \mathcal{F}_4)$, the use of self-supervised pretraining offers better eventual model robustness (around 3% improvement) and faster fine-tuning speed (almost saving the half number of epochs).

Next, we investigate how the *adversarial pretraining* (namely, $\mathcal{P}_3$) affects the eventual model robustness. It is shown by $(\mathcal{P}_3, \mathcal{F}_1)$ and $(\mathcal{P}_3, \mathcal{F}_2)$ in Table 3 that the robust feature representation learnt from $\mathcal{P}_3$ benefits adversarial robustness even in the case of partial fine-tuning, but the use of *adversarial partial fine-tuning*, namely, $(\mathcal{P}_3, \mathcal{F}_2)$, yields a 30% more improvement. We also observe from the case of $(\mathcal{P}_3, \mathcal{F}_3)$ that the *standard full fine-tuning* harms the ro-

Table 3: Evaluation Results of Eight Different $(\mathcal{P}_i, \mathcal{F}_j)$ Scenarios. Table 1 and Table 2 provide detailed definitions for $\mathcal{P}_1$ (without pre-training), $\mathcal{P}_2$ (standard self-supervision pre-training), $\mathcal{P}_3$ (adversarial self-supervision pre-training), $\mathcal{F}_1$ (partial standard fine-tuning), $\mathcal{F}_2$ (partial adversarial fine-tuning), $\mathcal{F}_3$ (full standard fine-tuning), and $\mathcal{F}_4$ (full adversarial fine-tuning). The best results are highlighted (1^{st},2^{nd}) under each column of different self-supervised pretraining tasks.

Scenario	*Selfie* Pretraining			*Rotation* Pretraining			*Jigsaw* Pretraining		
	TA (%)	RA (%)	Epochs	TA (%)	RA (%)	Epochs	TA (%)	RA (%)	Epochs
$(\mathcal{P}_1, \mathcal{F}_3)$	94.24	0.00	92	94.24	0.00	92	94.24	0.00	92
$(\mathcal{P}_1, \mathcal{F}_4)$	84.72	47.22	99	84.72	47.22	99	84.72	47.22	99
$(\mathcal{P}_2, \mathcal{F}_3)$	95.09	0.00	97	95.45	0.00	92	93.93	0.00	89
$(\mathcal{P}_2, \mathcal{F}_4)$	85.56	50.42	60	86.66	50.95	45	85.18	50.94	46
$(\mathcal{P}_3, \mathcal{F}_1)$	78.93	6.30	82	86.83	18.22	99	80.47	2.68	87
$(\mathcal{P}_3, \mathcal{F}_2)$	74.30	37.65	64	82.32	45.10	47	72.76	32.59	51
$(\mathcal{P}_3, \mathcal{F}_3)$	94.69	0.00	86	94.79	0.00	92	93.06	0.00	93
$(\mathcal{P}_3, \mathcal{F}_4)$	86.02	51.05	50	85.66	50.40	46	84.50	49.61	48

bust feature representation learnt from $\mathcal{P}_3$, leading to 0% RA. Furthermore, when the *adversairal full fine-tuning* is adopted, namely, $(\mathcal{P}_3, \mathcal{F}_4)$, the most significant robustness improvement is acquired. This observation is consistent with $(\mathcal{P}_2, \mathcal{F}_4)$ against $(\mathcal{P}_2, \mathcal{F}_3)$.

Third, at the first glance, *adversarial full fine-tuning* (namely, $\mathcal{F}_4$) is the most important step to improve the final mode robustness. However, *adversarial pretraining* is also a key, particularly for reducing the computation cost of fine-tuning; for example, less than 50 epochs in $(\mathcal{P}_3, \mathcal{F}_4)$ vs. 99 epochs in the end-to-end AT $(\mathcal{P}_1, \mathcal{F}_4)$.

Last but not the least, we note that the aforementioned results are consistent against different self-supervised prediction tasks. However, *Selfie* and *Rotation* are more favored than *Jigsaw* to improve the final model robustness. For example, in the cases of adversarial pretraining followed by standard and adversarial partial fine-tuning, namely, $(\mathcal{P}_3, \mathcal{F}_1)$ and $(\mathcal{P}_3, \mathcal{F}_2)$, *Selfie* and *Rotation* yields at least 3.5% improvement in RA. As the adversarial full fine-tuning is used, namely, $(\mathcal{P}_3, \mathcal{F}_4)$, *Selfie* and *Rotation* outperform *Jigsaw* in both TA and RA, where *Selfie* yields the largest improvement, around 2.5% in both TA and RA.

4.4. Comparison with one-shot AT regularized by self-supervised prediction task

In what follows, we compare our proposed adversarial pretraining followed by adversarial fine-tuning approach, namely, $(\mathcal{P}_3, \mathcal{F}_4)$ in Table 3 with the one-shot AT that optimizes a classification task regularized by the self-supervised *rotation* prediction task [14]. In addition to evaluating this comparison in TA and RA (evaluated at ℓ_∞ PGD attack [21]), we also measure the robustness in eventual classification against 12 unforeseen attacks that are not used in AT [17]. More results can be found in the supplement.

Figure 3 presents the multi-dimensional performance comparison of our approach vs. the baseline method in [14]. As we can see, our approach yields 1.97% improvement on TA while 0.74% degradation on RA. However, our approach yields consistent robustness improvement in defending all 12 unforeseen attacks, where the improvement ranges from 1.03% to 6.53%. Moreover, our approach separates pretraining and fine-tuning such that the target image classifier can be learnt from a warm start, namely, the adversarial pretrained representation network. This mitigates the computation drawback of one-shot AT in [14], recalling that our advantage in saving computation cost was shown in Table 3. Next, Figure 4 presents the performance of our approach under different types of self-supervised prediction task. As we can see, *Selfie* provides consistently better performance than others, where *Jigsaw* performs the worst.

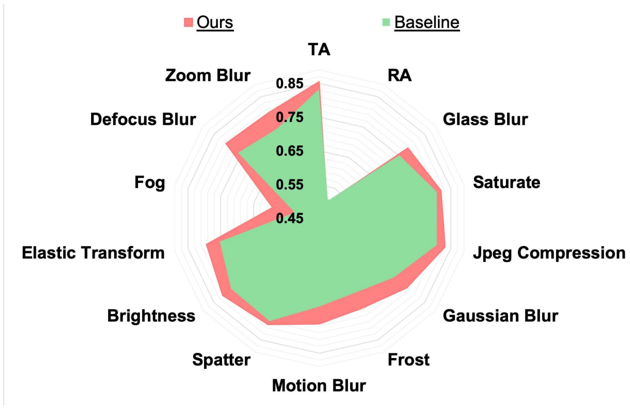

Figure 3: The summary of the accuracy over unforeseen adversarial attackers. Our models are obtained after adversarial fine-tuning with adversarial *Rotation* pretraining. Baseline are co-optimized models with *Rotation* auxiliary task [14].

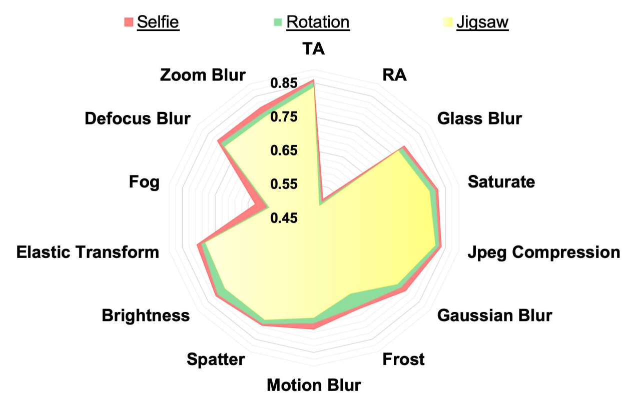

Figure 4: The summary of the accuracy over unforeseen adversarial attackers. Competition among adversarial fine-tuned models with *Selfie*, *Rotation* and *Jigsaw* adversarial pretraining.

4.5. Diversity vs. Task Ensemble

In what follows, we show that different self-supervised prediction tasks demonstrate a *diverse adversarial vulnerability* even if their corresponding RAs remain similar. We evaluate such a diversity through the transferability of adversarial examples generated from robust classifiers fine-tuned from the adversarially pretrained models using different self-supervised prediction tasks. We then demonstrate the performance of our proposed adversarial pretraining method (2) by leveraging an ensemble of *Selfie*, *Rotation*, and *Jigsaw*.

In Table 4, we present the transferbility of PGD attacks generated from the final model trained using adversarial pretraining followed by adversarial full fine-tuning, namely, $(\mathcal{P}_3, \mathcal{F}_4)$, where for ease of presentation, let $\mathrm{Model}(t)$ denote the classifier learnt using the self-supervised pretraining task $t \in \{\textit{Selfie}, \textit{rotation}, \textit{Jigsaw}\}$. Given the PGD attacks from $\mathrm{Model}(t)$, we evaluate their transferbility, in terms of attack success rate (ASR[2]), against $\mathrm{Model}(t')$. If $t' = t$, then ASR reduces to $1 - \mathrm{RA}$. If $t' \neq t$, then ASR reflects the attack transferbility from $\mathrm{Model}(t)$ to $\mathrm{Model}(t')$. As we can see, the diagonal entries of Table 4 correspond to the largest ASR at each column. This is not surprising, since transferring to another model makes the attack being weaker. One interesting observation is that ASR suffers a larger drop when transferring attacks from $\mathrm{Model}(\textit{Jigsaw})$ to other target models. This implies that $\mathrm{Model}(\textit{Selfie})$ and $\mathrm{Model}(\textit{Rotation})$ yields better robustness, consistent with our previous results like Figure 4.

At the first glance, the values of ASR of transfer attacks from $\mathrm{Model}(t)$ to $\mathrm{Model}(t')$ $(t' \neq t)$ keep similar, e.g., the first column of Table 4 where $t = \textit{Selfie}$ and $t' = \textit{Rotation}$ $(38.92\%$ ASR) or $t' = \textit{Jigsaw}$ $(38.96\%$ ASR). However, Figure 5 shows that the seemingly similar transferability are built on more *diverse* adversarial examples that succeed to attack $\mathrm{Model}(\textit{Rotation})$ and $\mathrm{Model}(\textit{Jigsaw})$, respectively. As we can see, there exist at least 14% transfer examples that are non-overlapped when successfully attacking $\mathrm{Model}(\textit{Rotation})$ and $\mathrm{Model}(\textit{Jigsaw})$. This diverse distribution of transferred adversarial examples against models using different self-supervised pretraining tasks motivates us to further improve the robustness by leveraging an ensemble of diversified pretraining tasks.

In Figure 2, we demonstrate the effectiveness of our proposed adversarial pretraining via diversity-promoted ensemble (AP + DPE) given in (2). Here we consider 4 baseline methods: 3 single task based adversarial pretraining, and adversarial pretraining via standard ensemble (AP + SE), corresponding to $\lambda = 0$ in (2). As we can see in Table 5, AP + DPE yields at least 1.17% improvement on RA while at most 3.02% degradation on TA, comparing with the best single fine-tuned model. In addition to the ensemble at the pretraining stage, we consider a simple but the most computationally intensive ensemble strategy, an averaged predictions over three final robust models learnt using adversarial pretraining $\mathcal{P}_3$ followed by adversarial fine-tuning $\mathcal{F}_4$ over *Selfie*, *rotation*, and *Jigsaw*. As we can see in Table 6, the best combination, ensemble of three fine-tuned models, yields at least 3.59% on RA while maintains a slight higher TA. More results of other ensemble configurations can be found in the supplement.

Table 4: The vulnerability diversity among fine-tuned models with *Selfie*, *Rotation* and *Jigsaw* self-supervised adversarial pretraining. The results take full adversarial fine-tuning. The highest ASRs are highlighted (1^{st},2^{nd}) under each column of PGD attacks from different fine-tuned models. Ensemble model results to different PGD attacks can be found in our supplement.

$(\mathcal{P}_3, \mathcal{F}_4)$ Attack Evaluation	PGD attacks from Model(*Selfie*)	PGD attacks from Model(*Rotation*)	PGD attacks from Model(*Jigsaw*)
Model(*Selfie*)	48.95%	37.75%	36.65%
Model(*Rotaion*)	38.92%	49.60%	38.12%
Model(*Jigsaw*)	38.96%	39.56%	51.17%

4.6. Ablation Study and Analysis

For comparison fairness, we fine-tune all models in the same CIFAR-10 dataset. In each ablation, we show results under scenarios $(\mathcal{P}_3, \mathcal{F}_2)$ and $(\mathcal{P}_3, \mathcal{F}_4)$, where $\mathcal{P}_3$ represents adversarial pretraining, $\mathcal{F}_2$ represents partial adversarial fine-tuning and $\mathcal{F}_4$ represents full adversarial fine-tuning. More ablation results can be found in the supplement.

Ablation of the pretraining data size As shown in Table 7, as the pretraining dataset grows larger, the standard

[2] ASR is given by the ratio of *successful* adversarial examples over the total number of $10,000$ test images.

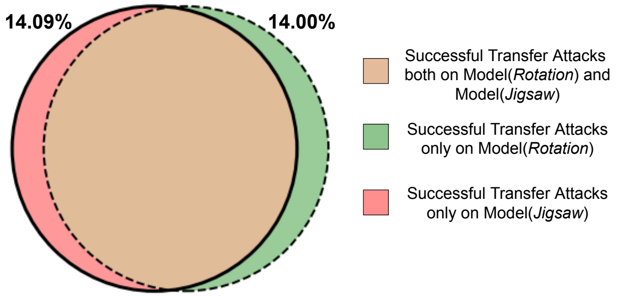

Figure 5: The VENN plot between sets of successful transfer adversarial examples from Model(*Selfie*) to Model(*Rotation*) and Model(*Selfie*) to Model(*Jigsaw*). The overlapping *Brown area* (🟫) represents the successful transfer attacks both on Model(*Rotation*) and Model(*Jigsaw*) from Model(*Selfie*). The *Pink area* (🟥) represents the successful transfer attacks only on Model(*Jigsaw*) from Model*Selfie*. The *Green area* (🟩) represents the successful transfer attacks only on Model(*Rotation*) from Model(*Selfie*).

Table 5: Results comparison between fine-tuned model from single task pretraining and fine-tuned model from tasks ensemble pretraining. AP + SE represents adversarial pretraining via standard ensemble. AP + DPE represents adversarial pretraining via diversity-promoted ensemble. The best results are highlighted (1^{st},2^{nd}) under each column of evaluation metrics.

Models	TA (%)	RA (%)	Epochs
Selfie Pretraining	86.02	51.05	50
Rotation Pretraining	85.66	50.40	46
Jigsaw Pretraining	83.74	48.83	48
AP + SE	84.44	49.53	47
AP + DPE	83.00	52.22	56

Table 6: Ensemble results of fine-tuned models with different adversarial pretrainings. The best results are highlighted (1^{st},2^{nd}) under each column of evaluation metrics.

Fine-tuned Models $(\mathcal{P}_3, \mathcal{F}_4)$	TA (%)	RA (%)
Jigsaw + Rotation	85.36	53.08
Jigsaw + Selfie	85.64	53.32
Rotation + Selfie	86.51	53.83
Jigsaw + Rotation + Selfie	86.04	54.64

and robust accuracies both demonstrate steady growth. Under the $(\mathcal{P}_3, \mathcal{F}_4)$ scenario, when the pretraining data size increases from 30K to 150K, we observe a 0.97% gain on robust accuracy with nearly the same standard accuracy. That aligns with the existing theory [30]. Since self-supervised pretraining requires no label, we could in future grow the unlabeled data size almost for free to continuously boost the pretraining performance.

Table 7: Ablation results of the size of pretraining datasets. All pretraining datasets have 32 × 32 resolution and 10 classes.

Scenario	CIFAR-30K		
	TA (%)	RA (%)	Epochs
$(\mathcal{P}_3, \mathcal{F}_2)$	65.65	30.00	70
$(\mathcal{P}_3, \mathcal{F}_4)$	85.29	49.64	42
Scenario	CIFAR-50K		
	TA (%)	RA (%)	Epochs
$(\mathcal{P}_3, \mathcal{F}_2)$	66.87	30.42	87
$(\mathcal{P}_3, \mathcal{F}_4)$	85.26	49.66	61
Scenario	CIFAR-150K		
	TA (%)	RA (%)	Epochs
$(\mathcal{P}_3, \mathcal{F}_2)$	67.73	30.24	95
$(\mathcal{P}_3, \mathcal{F}_4)$	85.18	50.61	55

Table 8: Ablation results of defense approaches. Instead of adversarial training, we perform random smoothing [5] for pretraining.

Random	*Selfie* Pretrainng			*Rotation* Pretraining			*Jigsaw* Pretraining		
Smoothing	TA (%)	RA (%)	Epochs	TA (%)	RA (%)	Epochs	TA (%)	RA (%)	Epochs
$\mathcal{F}_2$	71.9	30.57	61	74.7	34.23	78	74.66	33.84	68
$\mathcal{F}_4$	85.14	50.23	48	85.62	51.25	46	85.18	50.94	46

Ablation of defense approaches in pretraining In Table 8, we use random smoothing [5] in place of AT to robustify pretraining, while other protocols remain all unchanged. We obtain consistent results to using adversarial pretraining: robust pretraining speed up adversarial fine-tuning and helps final model robustness, while the full adversarial fine-tuning contributes the most to the robustness boost.

5. Conclusions

In this paper, we combine adversarial training with self-supervision to gain robust pretrained models, that can be readily applied towards downstream tasks through fine-tuning. We find that adversarial pretraining can not only boost final model robustness but also speed up the subsequent adversarial fine-tuning. We also find adversarial fine-tuning to contribute the most to the final robustness improvement. Further motivated by our observed diversity among different self-supervised tasks in pretraining, we propose an ensemble pretraining strategy that boosts robustness further. Our results observe consistent gains over state-of-the-art AT in terms of both standard and robust accuracy, leading to new benchmark numbers on CIFAR-10. In the future, we are interested to explore several promising directions revealed by our experiments and ablation studies, including incorporating more self-supervised tasks, extending the pretraining dataset size, and scaling up to high-resolution data.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *2018 ICML*, arXiv preprint arXiv:1802.00420, 2018. 2, 3

[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007. 1

[3] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019. 1, 2, 3

[4] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019. 2

[5] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019. 8

[6] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004. 1, 2

[7] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018. 2

[8] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015. 1, 2

[9] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 1, 2, 3

[10] Shupeng Gui, Haotao N Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. In *Advances in Neural Information Processing Systems*, pages 1283–1294, 2019. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 5

[13] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. 4, 5

[14] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019. 1, 3, 5, 6

[15] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 1

[16] Ting-Kuei Hu, Tianlong Chen, Haotao Wang, and Zhangyang Wang. Triple wins: Boosting accuracy, robustness and efficiency together by enabling input-adaptive inference. In *International Conference on Learning Representations*, 2020. 2

[17] Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019. 5, 6

[18] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016. 1

[19] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2018. 2

[20] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Towards understanding the transferability of deep representations. *arXiv preprint arXiv:1909.12031*, 2019. 1

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *2018 ICLR*, arXiv preprint arXiv:1706.06083, 2018. 1, 2, 3, 5, 6

[22] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017. 2

[23] Sina Mohseni, Mandar Pitale, JBS Yadawa, and Zhangyang Wang. Self-supervised learning for generalizable out-of-distribution detection. *AAAI*, 2020. 1

[24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016. 1

[25] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 1, 2, 3

[26] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. *arXiv preprint arXiv:1901.08846*, 2019. 2, 4

[27] Nicolas Papernot and Patrick McDaniel. Extending defensive distillation. *arXiv preprint arXiv:1705.05264*, 2017. 2

[28] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007. 1

[29] Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Computer vision with a single (robust) classifier. *arXiv preprint arXiv:1906.09453*, 2019. 4

[30] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pages 5014–5026, 2018. 1, 8

[31] Robert Stanforth, Alhussein Fawzi, Pushmeet Kohli, et al. Are labels required for improving adversarial robustness? *arXiv preprint arXiv:1905.13725*, 2019. 2

[32] Thilo Strauss, Markus Hanselmann, Andrej Junginger, and Holger Ulmer. Ensemble methods as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1709.03423*, 2017. 2

[33] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008. 4

[34] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 2

[35] Trieu H Trinh, Minh-Thang Luong, and Quoc V Le. Selfie: Self-supervised pretraining for image embedding. *arXiv preprint arXiv:1906.02940*, 2019. 1, 2, 3, 5

[36] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010. 1

[37] Jingkang Wang, Tianyun Zhang, Sijia Liu, Pin-Yu Chen, Jiacen Xu, Makan Fardad, and Bo Li. Towards a unified min-max framework for adversarial exploration and robustness, 2019. 2, 4

[38] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017. 2

[39] Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019. 2

[40] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019. 2, 5

[41] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 1, 2

[42] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*, 2020. 2