Adversarial Robustness: From Self-Supervised Pretraining to Fine-Tuning

Tianlong Chen\textsuperscript{1}, Sijia Liu\textsuperscript{2}, Shiyu Chang\textsuperscript{2}, Yu Cheng\textsuperscript{3}, Lisa Amini\textsuperscript{2}, Zhangyang Wang\textsuperscript{1}
\textsuperscript{1}Texas A\&M University, \textsuperscript{2}MIT-IBM Watson AI Lab, IBM Research \textsuperscript{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.

1. Introduction

Supervised training of deep neural networks requires massive, labeled datasets, which may be unavailable and costly to assemble \cite{14, 2, 27, 35}. 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 \cite{19, 22}. Examples of tasks proven to attain high accuracy through self-supervised pretraining include position predicting tasks (Selfie \cite{34}, Jigsaw \cite{24, 3}), rotation predicting tasks (Rotation \cite{8}), and a variety of other perception tasks \cite{5, 40, 7}.

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 \cite{17, 23}. As suggested by \cite{29}, the sample complexity of learning an adversarially robust model with current methods is significantly higher than that of standard learning. Adversarial training (AT) \cite{20}, 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 \cite{13}, by incorporating the self-supervised task as a complementary objective, which is co-optimized with the conventional classification loss through the method of AT \cite{20}. Their co-optimization

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 (●, 100 epochs) > Ensemble Pretrain (●, 56 epochs) > Selfie (●, 50 epochs) > Jigsaw (●, 48 epochs) > Rotation (●, 46 epochs)
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 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.

---

1Throughout this paper, we follow [39] 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.

### 2. Related Work

**Self-supervised pretraining.** Numerous self-supervised learning methods have been developed in recent years, including: region/component filling (e.g. inpainting [5] and colorization [40]); rotation prediction [8]; category prediction [7]; and patch-base spatial composition prediction (e.g., Jigsaw [24, 3] and Selfie [34]). All perform standard training, and do not tackle adversarial robustness. For example, Selfie [34], 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 [6], to label smoothening and feature squeezing [26, 37], to denoising and training on adversarial examples [21, 18]. A handful of recent works point out that those empirical defenses could still be easily compromised [1]. Adversarial training (AT) [20] 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. [9, 15] showed AT is also effective in compressing or accelerating models [41] while preserving learned robustness.

Several works have demonstrated model ensembles [31, 33] to boost adversarial robustness, as the ensemble diversity can challenge the transferability of adversarial examples. Recent proposals [25, 36] 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 [38], who also finds that learning with more unlabeled data can result in better adversarially robust generalization. Both works [30, 4] use unlabeled data to form an unsupervised auxiliary loss (e.g., a label-independent robust regularizer or a pseudo-label loss).

To the best of our knowledge, [13] 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 \( T_p \) denote a pretraining task and \( D_p \) denote the corresponding (unlabeled) pretraining dataset. The goal of self-supervised pretraining is to learn a model from \( D_p \) itself without explicit manual supervision. This is often cast as an optimization problem, in which a proposed pretraining loss \( \ell_p(f_p(\theta_p, \theta_{pc}; D_p)) \) is minimized to determine a model parameterized by \( \theta_p \). Here, \( \theta_{pc} \) signifies additional parameters customized for a given \( T_p \). 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** [34]: 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** [8]: 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** [24, 3]: By dividing an image into different patches, Jigsaw trains a classifier to predict the correct permutation of these patches.

**Supervised Fine-tuning** Let \( r(x; \theta_p) \) denote the mapping (parameterized by \( \theta_p \)) from an input sample \( x \) to its embedding space learnt from the self-supervised pretraining task \( T_p \). Given a target finetuning task \( T_f \) with the labeled dataset \( D_f \), the goal of fine-tuning is to determine a classifier, parameterized by \( \theta_f \), which maps the representation \( r(x; \theta_p) \) to the label space. To learn the classifier, one can minimize a common supervised training loss \( \ell_f(\theta_p, \theta_f; D_f) \) with a fixed or re-trainable model \( \theta_p \), 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 [20, 1]. Considering an \( \epsilon \)-tolerant \( \ell_\infty \) attack \( \delta \) subject to \( \|\delta\|_\infty \leq \epsilon \), an adversarial example of a benign input \( x \) is given by \( x + \delta \). With the aid of adversarial examples, AT solves a min-max optimization problem of the generic form

\[
\min_{\theta} \mathbb{E}_{x \in D} \left[ \max_{\|\delta\|_\infty \leq \epsilon} \ell(\theta, x + \delta) \right], \quad (1)
\]

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

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 \( \theta := [\theta_p^T, \theta_{pc}^T]^T \) and \( D := D_p \), and specifying \( \ell \) as \( \ell_p \). In Table 1, we summarize all the possible scenarios when AT meets self-supervised pretraining.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Pretraining method</th>
<th>Loss ( \ell ) in (1)</th>
<th>Variables ( \theta ) in (1)</th>
<th>dataset ( D ) in (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_1 )</td>
<td>None(^1)</td>
<td>NA(^1)</td>
<td>NA</td>
<td>( D_p )</td>
</tr>
<tr>
<td>( P_2 )</td>
<td>ST(^2)</td>
<td>( \ell_p )</td>
<td>( [\theta^T_p, \theta_{pc}^T]^T )</td>
<td>( D_p )</td>
</tr>
<tr>
<td>( P_3 )</td>
<td>AF</td>
<td>( \ell_p )</td>
<td>( [\theta^T_p, \theta_{pc}^T]^T )</td>
<td>( D_p )</td>
</tr>
</tbody>
</table>

1: None: the model form of \( \theta_p \) is known in advance.
2: NA: Not applicable.
3: ST: A special case of (1) with \( \delta = 0 \).

**Table 2:** Summary of fine-tuning scenarios.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Fine-tuning type</th>
<th>Fine-tuning method</th>
<th>Loss ( \ell ) in (1)</th>
<th>Variables ( \theta ) in (1)</th>
<th>dataset ( D ) in (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_1 )</td>
<td>Partial (with fixed ( \theta_f ))</td>
<td>ST</td>
<td>( \ell_f )</td>
<td>( \theta_f )</td>
<td>( D_f )</td>
</tr>
<tr>
<td>( F_2 )</td>
<td>Partial (with fixed ( \theta_f ))</td>
<td>AT</td>
<td>( \ell_f )</td>
<td>( \theta_f )</td>
<td>( D_f )</td>
</tr>
<tr>
<td>( F_3 )</td>
<td>Full(^2)</td>
<td>ST</td>
<td>( \ell_f )</td>
<td>( [\theta^T_f, \theta_{pc}^T]^T )</td>
<td>( D_f )</td>
</tr>
<tr>
<td>( F_4 )</td>
<td>Full</td>
<td>AT</td>
<td>( \ell_f )</td>
<td>( [\theta^T_f, \theta_{pc}^T]^T )</td>
<td>( D_f )</td>
</tr>
</tbody>
</table>

1: Fixed \( \theta_f \) signifies the model learnt in a given pretraining scenario.
2: Full fine-tuning retains \( \theta_p \).

Given a pretrained model \( \theta_p \), 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 \( \theta_p \), and the latter case b) solves a supervised fine-tuning task by retraining \( \theta_p \). 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 \( \{P_i, F_j\} \) provided by Tables 1-2 is different from [13], 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 \( \{\{P_i, F_j\}\} \), **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 downstream 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 $\{T_p^{(i)}\}_i$, each of which obeys the formulation in Section 3.1. We generalize problem (1) to

$$\min_{\theta_p, \{\theta_{pc}^{(i)}\}} \mathbb{E}_{x \sim D_p} \left[ L_{adv}(\theta_p, \{\theta_{pc}^{(i)}\}, x) \right], \quad (2)$$

where $L_{adv}$ denotes the adversarial loss given by

$$L_{adv}(\theta_p, \{\theta_{pc}^{(i)}\}, x) := \max_{\|\delta^{(i)}\|_\infty \leq \epsilon} \sum_{i=1}^M \ell_p^{(i)}(\theta_p, \theta_{pc}^{(i)}, x + \delta^{(i)}) + \lambda g(\theta_p, \{\theta_{pc}^{(i)}\}, \{\delta^{(i)}\}). \quad (3)$$

In (2), for ease of notation, we replace $\{\cdot\}_i$ with $\{\cdot\}$. $\theta_p$ denotes the common network shared among different self-supervised tasks, and $\theta_{pc}^{(i)}$ denotes a sub-network customized for the $i$th task. We refer readers to Figure 2 for an overview of our proposed model architecture. In (3), $\ell_p^{(i)}$ denotes the $i$th 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 $\lambda$ around the value chosen in [25]. Details are referred to the supplement.

Spurred by [25, 36], we quantify the diversity-promoting regularizer $g$ through the orthogonality of input gradients of different self-supervised pretraining losses,

$$g(\theta_p, \{\theta_{pc}^{(i)}\}, \{\delta^{(i)}\}) := \log \det(G^T G), \quad (4)$$

where each column of $G$ corresponds to a normalized input gradient. $\{\nabla_{\delta^{(i)}} \ell_p^{(i)}(\theta_p, \theta_{pc}^{(i)}, x + \delta^{(i)})\}$ 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 $\theta_p$ by defending attacks from diversified perturbation directions.

4. Experiments and Results

4.1. Datasets

**Dataset Details** We consider three different datasets in our experiments: CIFAR-10, CIFAR-10-C [12] 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 [28] 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 [32] 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.

4.2. Implementation Details

**Model Architecture:** For pretraining with the Selfie task, we identically follow the setting in [34]. For Rotation and Jigsaw pretraining tasks, we use ResNet-50v2 [11]. 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.

We use 10-step and 20-step $\ell_\infty$ PGD attacks [20] for adversarial training and evaluation, respectively. Unless otherwise specified, we follow [13]'s setting with $\epsilon = \frac{8}{255}$ and $\alpha = \frac{2}{255}$. We also consider unforeseen attacks [16, 12].

**Evaluation Metrics**: We follow [39] 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.

### 4.3. Adversarial self-supervised pretraining & 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 $(P_i, F_j)$ denotes a specified pretraining+fine-tuning scheme. The baseline schemes are given by the end-to-end standard training (ST), namely, $(P_1, F_3)$ and the end-to-end adversarial training (AT), namely, $(P_1, F_4)$. 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 $P_2$ with fine-tuning schemes $F_3$ and $F_4$ used in baseline methods. Several observations can be made from the comparison $(P_2, F_3)$ vs. $(P_1, F_3)$ and $(P_2, F_4)$ vs. $(P_1, F_4)$ 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 $F_4$ (against standard fine-tuning $F_3$) is crucial, leading to significantly improved RA under both $P_1$ and $P_2$; 3) Compared $(P_1, F_4)$ with $(P_2, F_4)$, 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, $P_3$) affects the eventual model robustness. It is shown by $(P_3, F_1)$ and $(P_3, F_2)$ in Table 3 that the robust feature representation learnt from $P_3$ benefits adversarial robustness even in the case of partial fine-tuning, but the use of adversarial partial fine-tuning, namely, $(P_3, F_2)$, yields a 30% more improvement. We also observe from the case of $(P_3, F_3)$ that the standard full fine-tuning harms the robust feature representation learnt from $P_3$, leading to 0% RA. Furthermore, when the adversarial full fine-tuning is adopted, namely, $(P_3, F_4)$, the most significant robustness improvement is acquired. This observation is consistent with $(P_2, F_4)$ against $(P_2, F_3)$.

Third, at the first glance, adversarial full fine-tuning (namely, $F_4$) 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 $(P_3, F_4)$ vs. 100 epochs in the end-to-end AT $(P_1, F_4)$.

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, $(P_3, F_1)$ and $(P_3, F_2)$, Selfie and Rotation yields at least 3.5% improvement in RA. As the adversarial full fine-tuning is used,
Table 3: Evaluation Results of Eight Different ($P_i, F_j$) Scenarios. The best results are highlighted ($1^{st}, 2^{nd}$) under each column of different self-supervised pretraining tasks.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Selfie Pretraining</th>
<th>Rotation Pretraining</th>
<th>Jigsaw Pretraining</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TA (%)</td>
<td>RA (%)</td>
<td>Epochs</td>
</tr>
<tr>
<td>($P_1, F_3$)</td>
<td>94.24</td>
<td>0.00</td>
<td>92</td>
</tr>
<tr>
<td>($P_1, F_4$)</td>
<td>84.72</td>
<td>47.22</td>
<td>100</td>
</tr>
<tr>
<td>($P_2, F_3$)</td>
<td>95.09</td>
<td>0.00</td>
<td>97</td>
</tr>
<tr>
<td>($P_2, F_4$)</td>
<td>85.56</td>
<td>50.42</td>
<td>60</td>
</tr>
<tr>
<td>($P_3, F_1$)</td>
<td>78.93</td>
<td>6.30</td>
<td>82</td>
</tr>
<tr>
<td>($P_3, F_2$)</td>
<td>74.30</td>
<td>37.65</td>
<td>64</td>
</tr>
<tr>
<td>($P_3, F_3$)</td>
<td>94.69</td>
<td>0.00</td>
<td>86</td>
</tr>
<tr>
<td>($P_3, F_4$)</td>
<td>86.02</td>
<td>51.05</td>
<td>50</td>
</tr>
</tbody>
</table>

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, ($P_3, F_4$) in Table 3 with the one-shot AT that optimizes a classification task regularized by the self-supervised rotation prediction task [13]. In addition to evaluating this comparison in TA and RA (evaluated at $\ell_\infty$ PGD attack [20]), we also measure the robustness in eventual classification against 12 unforeseen attacks that are not used in AT [16]. More results can be found in the supplement.

Figure 3 presents the multi-dimensional performance comparison of our approach vs. the baseline method in [13]. 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 [13], 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.

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 fine-tuning, namely, (P₃, F₄), where for ease of presentation, let Model(t) denote the classifier learnt using the self-supervised pretraining task t ∈ {Selfie, rotation, Jigsaw}. Given the PGD attacks from Model(t), we evaluate their transferbility, in terms of attack success rate (ASR²), against Model(t′). If t′ = t, then ASR reduces to 1 − RA. If t′ ≠ t, then ASR reflects the attack transferbility from Model(t) to 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 Model(Jigsaw) to other target models. This implies that Model(Selfie) and Model(Rotation) yields better robustness, consistent with our previous results like Figure 4.

At the first glance, the values of ASR of transfer attacks from Model(t) to Model(t′) (t′ ≠ t) keep similar, e.g., the first column of Table 4 where t = Selfie and t′ = Rotation (38.92% ASR) or t′ = Jigsaw (38.96% ASR). However, Figure 5 shows that the seemingly similar transferability are built on more diverse adversarial examples that succeed to attack Model(Rotation) and Model(Jigsaw), respectively. As we can see, there exist at least 14% transfer examples that are non-overlapped when successfully attacking Model(Rotation) and Model(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 λ = 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 P₃ followed by adversarial fine-tuning F₄ over Selfie, rotation, and Jigsaw. As we can see in Table 5, the best combination, ensemble of three fine-tuned models, yields at least 3.59% on RA while maintains a slight higher TA.

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 (1ₚ, 2ₚ) under each column of PGD attacks from different fine-tuned models.

<table>
<thead>
<tr>
<th>(P₃, F₄) Attack Evaluation</th>
<th>PGD attacks from Model(Selfie)</th>
<th>PGD attacks from Model(Rotation)</th>
<th>PGD attacks from Model(Jigsaw)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model(Selfie)</td>
<td>48.95%</td>
<td>37.75%</td>
<td>36.63%</td>
</tr>
<tr>
<td>Model(Rotation)</td>
<td>38.92%</td>
<td>49.66%</td>
<td>38.12%</td>
</tr>
<tr>
<td>Model(Jigsaw)</td>
<td>38.96%</td>
<td>39.56%</td>
<td>51.17%</td>
</tr>
</tbody>
</table>

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 (1ₚ, 2ₚ) under each column of evaluation metrics.

<table>
<thead>
<tr>
<th>Models</th>
<th>TA (%)</th>
<th>RA (%)</th>
<th>Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selfie Pretraining</td>
<td>86.02</td>
<td>51.05</td>
<td>50</td>
</tr>
<tr>
<td>Rotation Pretraining</td>
<td>85.66</td>
<td>50.40</td>
<td>46</td>
</tr>
<tr>
<td>Jigsaw Pretraining</td>
<td>83.74</td>
<td>48.83</td>
<td>48</td>
</tr>
<tr>
<td>AP + SE</td>
<td>84.44</td>
<td>49.53</td>
<td>47</td>
</tr>
<tr>
<td>AP + DPE</td>
<td>83.00</td>
<td>52.22</td>
<td>56</td>
</tr>
</tbody>
</table>

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 (P₃, F₂) and (P₃, F₄), where P₃ represents adversarial pretraining, F₂ represents partial adversarial fine-tuning and F₄ represents full adversarial fine-tuning.

²ASR is given by the ratio of successful adversarial examples over the total number of 10,000 test images.
Table 6: Ensemble results of fine-tuned models with different adversarial pretrainings. The best results are highlighted (1st, 2nd) under each column of evaluation metrics.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>CIFAR-30K</th>
<th>CIFAR-50K</th>
<th>CIFAR-150K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TA (%)</td>
<td>RA (%)</td>
<td>Epochs</td>
</tr>
<tr>
<td>Jigsaw + Rotation</td>
<td>85.36</td>
<td>53.08</td>
<td>70</td>
</tr>
<tr>
<td>Jigsaw + Selfie</td>
<td>85.64</td>
<td>53.32</td>
<td>61</td>
</tr>
<tr>
<td>Rotation + Selfie</td>
<td>86.51</td>
<td>53.83</td>
<td>66</td>
</tr>
<tr>
<td>Jigsaw + Rotation + Selfie</td>
<td>86.04</td>
<td>54.64</td>
<td>65</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Scenario</th>
<th>CIFAR-30K</th>
<th>CIFAR-50K</th>
<th>CIFAR-150K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TA (%)</td>
<td>RA (%)</td>
<td>Epochs</td>
</tr>
<tr>
<td>(P₃, F₃)</td>
<td>65.65</td>
<td>30.00</td>
<td>70</td>
</tr>
<tr>
<td>(P₄, F₄)</td>
<td>85.29</td>
<td>49.64</td>
<td>66</td>
</tr>
<tr>
<td>(P₃, F₂)</td>
<td>66.87</td>
<td>30.42</td>
<td>87</td>
</tr>
<tr>
<td>(P₄, F₂)</td>
<td>85.26</td>
<td>49.66</td>
<td>61</td>
</tr>
<tr>
<td>(P₃, F₃)</td>
<td>67.73</td>
<td>30.24</td>
<td>95</td>
</tr>
<tr>
<td>(P₄, F₃)</td>
<td>85.18</td>
<td>50.61</td>
<td>55</td>
</tr>
</tbody>
</table>

Table 8: Ablation results of the image resolution in the pretraining datasets Dₚ. Both datasets have 30,000 images for pretraining.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>CIFAR-30K (32 × 32)</th>
<th>R-ImageNet-224 (224 × 224)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TA (%)</td>
<td>RA (%)</td>
</tr>
<tr>
<td>(P₃, F₂)</td>
<td>68.04</td>
<td>31.53</td>
</tr>
<tr>
<td>(P₄, F₄)</td>
<td>85.29</td>
<td>49.64</td>
</tr>
</tbody>
</table>

Table 9: Ablation results of defense approaches. Instead of adversarial training, we perform random smoothing during pretraining.

<table>
<thead>
<tr>
<th>Random Smoothing</th>
<th>Jigsaw Pretraining</th>
<th>Rotation Pretraining</th>
<th>Jigsaw Pretraining</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TA (%)</td>
<td>RA (%)</td>
<td>Epochs</td>
</tr>
<tr>
<td>P₂</td>
<td>85.14</td>
<td>51.25</td>
<td>46</td>
</tr>
<tr>
<td>P₃</td>
<td>85.29</td>
<td>49.64</td>
<td>70</td>
</tr>
</tbody>
</table>

Ablation of the pretraining data size As shown in Table 7, as the pretraining dataset grows larger, the standard and robust accuracies both demonstrate steady growth. Under the (P₃, F₃) scenario, when the pretraining data size increases from 30K to 150K, we observe a 0.9% gain on robust accuracy with nearly the same standard accuracy. That aligns with the existing theory [29]. 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.

Ablation of the image resolution in the pretraining dataset As shown in Table 8, fine-tuned models from CIFAR-30K (32 × 32) pretraining outperform ones from R-ImageNet-224 (224 × 224), in terms of both standard and robust accuracy.

There exist two possible reasons accounting for this (somehow unexpected) observation. First, pretraining with 224 × 224 images might lead to learning features that have a scale mismatch, with the fine-tuning task on the 32 × 32 images. That is consistent with [19]’s conclusion regarding the domain (mis)match between pretraining and fine-tuning datasets. Second, a restricted set of image classes is considered in ImageNet in order to make adversarial pretraining feasible on the 224 × 224 scale. However, the restricted version of ImageNet might induce bias during fine-tuning. We plan to conduct further experiments with both pretraining and fine-tuning on the 224 × 224 scale to test our conjecture.

Ablation of defense approaches in pretraining In Table 9, we use random smoothing 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


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


