A Study of the Effects of Transfer Learning on Adversarial Robustness

Anonymous authors
Paper under double-blind review

Abstract

The security and robustness of AI systems are critical in real-world deployments. While prior works have developed methods to train robust networks, these works implicitly assume that sufficient labeled data for robust training is present. However, in deployment scenarios with insufficient training data, robust networks cannot be trained using existing techniques. In such low-data regimes, non-robust training methods traditionally rely on *transfer learning*. First, a network is pre-trained on a large, possibly labeled dataset and then fine-tuned for a new task using the smaller set of training samples. The effectiveness of transfer learning with respect to adversarial robustness, though, is not well-studied. It is unclear if transfer learning can improve adversarial performance in low-data scenarios. In this paper, we perform a broad analysis of the effects of pre-training with respect to empirical and certified adversarial robustness. Using both supervised and self-supervised pre-training methods across a range of downstream tasks, we identify the circumstances necessary to train robust models on small-scale datasets. Our work also represents the first successful demonstration of training networks with high certified robustness for small-scale datasets.

1 Introduction

Transfer learning has been extensively studied for improving standard generalization in machine learning systems across various data availability scenarios (Yosinski et al., 2014; Kornblith et al., 2019; He et al., 2019). In the context of adversarial robustness, however, there are only limited works that studied the benefits of transfer learning (Hendrycks et al., 2019; Chen et al., 2020a). These works generally limit themselves to empirical robustness by solely using adversarial training (Madry et al., 2018) in their experiments. Furthermore, they only study the scenario where abundant data is available for the downstream tasks, *i.e.,* well-represented tasks (*e.g.,* CIFAR-10, CIFAR-100). The exact effect of transfer learning on empirical robustness when there is a lack of abundant data for the downstream tasks, *i.e.,* under-represented tasks, is therefore unknown.

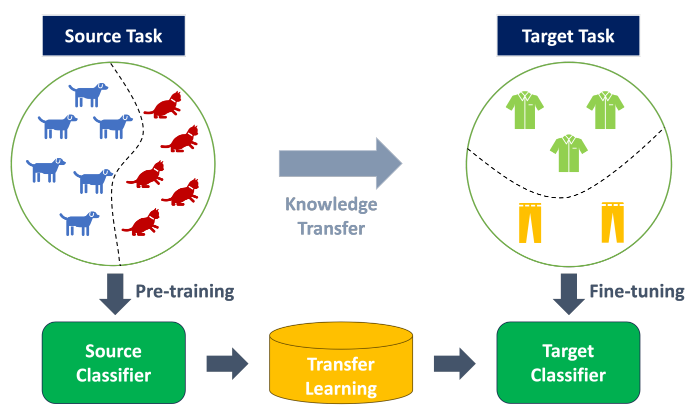

Figure 1: Through transfer learning, one can obtain high-performance networks in settings where it would otherwise be infeasible, *i.e.,* low-data regimes. First, the user trains a classifier on a source task with a large dataset to learn generalizable features. Next, the classifier is fine-tuned on a target task with a small dataset.

It is also unclear whether the findings in the context of empirical robustness would apply to certified robustness training methods, specifically randomized smoothing-based methods (Cohen et al., 2019; Salman et al., 2019; Zhai et al., 2020; Jeong & Shin, 2020; Jeong et al., 2021) which provide state-of-the-art certified robustness in the ℓ_2-space. This is

Table 1: Summarizing the findings of prior works regarding the usefulness of transfer learning towards standard generalization and adversarial robustness.

		Is Transfer Learning Useful?			
		Supervised		Self-Supervised	
		Low-Data	High-Data	Low-Data	High-Data
Standard Generalization		✔ (Kornblith et al., 2019)	✘ (He et al., 2019)	✔ (Chen et al., 2020b)	✔ (Chen et al., 2020b)
Adversarial Robustness	Empirical	?	✔ (Hendrycks et al., 2019)	?	✔ (Chen et al., 2020a)
	Certified	?	?	?	?

because both these classes of methods rely on fundamentally different ways of measuring and encoding adversarial robustness, and so classifiers trained using them inherit different properties. Case in point, Kireev et al. (2022) demonstrated that empirical and certified training methods exhibit dissimilar levels of robustness against common image corruptions. Finally, there is little work that studies the effect of self-supervised pre-training on adversarial robustness, with existing works limiting themselves to well-represented tasks.

Table 1 summarizes the findings of prior works in regard to improving performance and robustness in a range of data availability scenarios. The effects of transfer learning on adversarial robustness are largely unexplored (limited to empirical robustness and well-represented tasks). Furthermore, we note that self-supervised pre-training has become an important component of the transfer learning framework of late as it alleviates the need for labeled data for pre-training. The models fine-tuned using pre-trained weights generated via self-supervised learning have exhibited unprecedented generalization ability, unlocking large-scale commercial applications that were infeasible only a few years back. However, using self-supervision to train highly secure ML models is a topic that has largely been overlooked. Therefore, in this paper, we make adversarial robustness our primary focus and broadly study the effects of self-supervised pre-training on it.

We summarize the **contributions** of this work as follows:

- We perform a comprehensive study on the utility of transfer learning towards certified and empirical robustness across a range of downstream tasks. First, a model is robustly pre-trained on a large-scale dataset (*i.e.,* ImageNet) using supervised or self-supervised methods and then robustly fine-tuned on the downstream task. Our experimental results show that such pre-training is beneficial toward improving adversarial performance on downstream tasks compared to training on the downstream task directly.

- We further show that during transfer learning, only the fine-tuning portion of the pipeline needs to rely on robust training methods. This finding eases the overhead of training robust models. Also, regardless of the amount of labeled data available for either pre-training or fine-tuning, models with high adversarial robustness can be trained on downstream tasks.

- Finally, our work demonstrates the first successful demonstration of training models with high certified robustness on downstream tasks irrespective of the amount of labeled data available, either during pre-training or fine-tuning.

Our findings serve as a useful tool for ML practitioners wanting to deploy highly robustness models in a range of data availability scenarios.

2 Background

In this paper, we focus on transfer learning for image classification tasks. More specifically, we explore whether transfer learning can be used to train deep neural network-based image classifiers with high (empirical and certified) adversarial robustness in a range of data availability scenarios. In this section, we provide readers with the necessary background regarding transfer learning (Section 2.1) and adversarial robustness of deep neural networks (Section 2.2).

2.1 Transfer Learning

In transfer learning (Caruana, 1994; Pan & Yang, 2009; Bengio et al., 2011; Bengio, 2012; Yosinski et al., 2014; Huh et al., 2016), a network is pre-trained on a source task and then fine-tuned on a target task. Through pre-training, the network learns features that enable it to generalize better when fine-tuned on the target task (Yosinski et al., 2014). This is true even when the source and target tasks are dissimilar. For example, prior works (Sermanet et al., 2013; Girshick et al., 2014) re-purposed networks trained for ImageNet (Deng et al., 2009) classification task to achieve breakthroughs on object detection tasks. Pre-training has also been shown to be an effective solution for training high-performance networks when available training data is insufficient for standard training (Pan & Yang, 2009). However, He *et al.* (He et al., 2019) showed that, in the presence of abundant training data, similar levels of generalization can be achieved whether pre-training is performed or not. In such cases, the only benefit of transfer learning then is faster convergence and, therefore, savings in training time. Other studies found that transfer learning effectively transfers other desirable properties like shape bias (Utrera et al., 2020), robustness to common image corruptions (Yamada & Otani, 2022) and adversarial perturbations (Hendrycks et al., 2019).

2.1.1 Self-supervised Pre-training

Traditionally, pre-training was performed in a supervised fashion on large-scale labeled datasets, which can be challenging to acquire in many domains. However, unlabeled data tends to be widely available. To leverage these unlabeled datasets, self-supervised pre-training was proposed to enable models to learn generalizable features by optimizing a custom training objective. Contrastive learning (Chen et al., 2020b; Grill et al., 2020; He et al., 2020; Caron et al., 2020; Goyal et al., 2022) is one such approach. Models are trained to maximize the similarity between positive pairs (semantically similar data samples) while minimizing the similarity between negative pairs (semantically dissimilar data samples) in the feature space. SimCLR (Chen et al., 2020b), one of the most popular contrastive learning methods, generates the positive pairs by applying two different sets of input transformations (like cropping, color distortion, and blurring) to the same image. Negative pairs are generated using transformed versions of different images. Self-supervised methods often achieve state-of-the-art results in a range of applications such as image classification, object detection, and sentiment analysis after fine-tuning on relatively small amounts of labeled data.

2.2 Adversarial Robustness

Neural networks are known to be susceptible to adversarial evasion attacks, which attempt to modify a given input imperceptibly with the goal of triggering misclassification. Since the discovery of this vulnerability, several methods have been proposed to train neural networks that are robust against such attacks. These methods can be broadly classified as *empirical* and *certified* methods based on the nature of the robustness guarantees they provide.

2.2.1 Empirical Adversarial Robustness

Empirical adversarial robustness is traditionally measured using the strongest possible attack within a predetermined threat model. Robustness training methods that rely on this strategy train the neural network to be robust against this strongest attack and, in turn, gain robustness against all possible attacks within the same threat model. However, such robustness is not provable in nature and can be challenged by an adaptive adversary (Carlini & Wagner, 2017; Athalye et al., 2018; Tramer et al., 2020). Adversarial training (Madry et al., 2018), is one of the most promising empirical robustness methods, as is evident from the fact that the current state-of-the-art methods (Zhang et al., 2019; Wu et al., 2020) is derived from the basic framework proposed by Madry *et al.* (Madry et al., 2018). This framework involves generating adversarial inputs on the fly during training and updating the neural network's weights using them. Furthermore, several works (Tsipras et al., 2019; Ilyas et al., 2019; Augustin et al., 2020) still study the models trained by Madry *et al.* to learn more about adversarial robustness in general. Due to its prominence and in an attempt to fall in line with prior works, we use adversarial training as a representative of empirical robustness training methods.

2.2.2 Certified Adversarial Robustness

Despite the progress made towards developing empirical robustness methods with strong robustness guarantees, the lack of provability remains an issue. Provably/certifiably robust training methods remedy this concern by maximizing the lower bound of a neural network's output corresponding to the correct class within a certain range of input perturbations. If, for a given input, the lower bound of the correct class output is higher than the upper bound of all other class outputs, the neural network is provably robust for that input. Computing and maximizing this lower bound for a multi-layer neural network is an NP-hard problem (Katz et al., 2017). In recent literature, several methods have been proposed to approximately compute this lower bound and incorporate it in the training process of the neural network in a scalable manner. Of these, randomized smoothing based methods (Cohen et al., 2019; Salman et al., 2019; Zhai et al., 2020; Jeong & Shin, 2020; Jeong et al., 2021) yield state-of-the-art robustness in the ℓ_2-space for modern neural networks. Therefore, in this paper, we focus on these methods.

First formalized by Cohen *et al.* (Cohen et al., 2019), randomized smoothing defines the concept of a smooth classifier. Given a base classifier f_θ, the **smooth classifier** g_θ, is defined as follows:

$$g_\theta(x) = \arg\max_{c \in \mathcal{Y}} P_{\eta \sim \mathcal{N}(0,\sigma^2 I)}(f_\theta(x+\eta) = c) \tag{1}$$

Simply put, the smooth classifier returns the class c, which has the highest probability mass under the Gaussian distribution $\mathcal{N}(x, \sigma^2 I)$. If, for a given input x, the smooth classifier's output c is equal to the ground truth label y, it is said to be certifiably robust (with high probability) at x. The **certified radius**, *i.e.,* the input radius in which x's prediction is consistent, is given by:

$$\begin{aligned} CR(g_\theta; x, y) = \frac{\sigma}{2}[&\Phi^{-1}(P_\eta(f_\theta(x+\eta) = y)) - \\ &\Phi^{-1}(\max_{y' \neq y} P_\eta(f_\theta(x+\eta) = y'))] \end{aligned} \tag{2}$$

Randomized smoothing-based robustness training methods focus on maximizing the average certified radius for a given dataset (Cohen et al., 2019; Salman et al., 2019; Zhai et al., 2020; Jeong et al., 2021). Cohen *et al.* (Cohen et al., 2019) simply augmented the training data with Gaussian noise when training the base classifier. Salman *et al.* (Salman et al., 2019) modified the adversarial training objective to work in this new framework. Zhai *et al.* (Zhai et al., 2020) derived a differentiable approximation of the certified radius and directly maximized it during training. Jeong *et al.* (Jeong & Shin, 2020) find that the certified robustness of a smooth classifier can be greatly improved by enforcing the base classifier's outputs over several noisy copies of a given input to be consistent. They achieve this consistency by using a regularization loss that forces the output for a noisy copy of the input to be closer to the expected output over several noisy copies. Finally, Jeong *et al.* (Jeong et al., 2021) identified that the certified radius of the smooth classifier is aligned with its prediction confidence and used a combination of adversarial training and *mixup* (Zhang et al., 2018) to favorably calibrate the prediction confidence.

3 Empirical is not the same as Certified

Broadly, the process of training adversarially robust classifiers can be dissected into two key stages: (i) quantifying the adversarial risk across the training data distribution and (ii) minimizing this adversarial risk during the training process. Empirical methods, such as Adversarial Training, measure the adversarial risk by determining the maximum loss achievable for an input subjected to adversarial manipulations. This entails employing the most potent attack within a predefined threat model. On the other hand, certified methods utilizing randomized smoothing measure adversarial risk by measuring the largest possible radius around a given input within which the classifier's output remains consistent (Equation 2). The fundamental disparity

Table 2: Performance comparison of (base) classifiers trained using AT (Madry et al., 2018) and CR (Jeong & Shin, 2020) under varying levels of Gaussian noise. The AT classifier shows a gradual decline in performance as the noise severity increases, whereas the CR classifier overfits to the level of noise encountered during training (*i.e.,* $\sigma = 0.5$).

Method	**Noise Stddev (σ)**			
	0.001	**0.01**	**0.1**	**0.5**
Adversarial Training	54.5	54.5	28.5	0.1
Consistency Regularization	19.0	19.1	21.8	60.5

Table 3: Certified accuracy at different ℓ_2 radii of smooth classifiers deploying base classifiers trained with AT (Madry et al., 2018) and CR (Jeong & Shin, 2020). The smoothing is performed using $\sigma = 0.01$ and 0.5, respectively. AT smooth classifier exhibits no meaningful certified robustness.

Method	**ℓ_2 radius**			
	0.0	**0.5**	**1.0**	**1.5**
Adversarial Training	49.6	0.0	0.0	0.0
Consistency Regularization	54.8	50.1	43.8	33.5

in how adversarial risk is assessed encourages empirical and certified classifiers to manifest distinct robustness properties that the other may not possess. We perform a brief investigation using Gaussian noise image corruption that illustrates the differences between these two robustness classes. Through this demonstration, we underscore the caution required when extending findings from empirical robustness to the realm of certified robustness.

First, we train two ResNet-50 classifiers on ImageNet: (i) using an empirical robustness training method, *i.e.,* Adversarial Training (AT) (Madry et al., 2018), and (ii) using a certified robustness training method, *i.e.,* Consistency Regularization (CR) (Jeong & Shin, 2020). For CR, we use a Gaussian noise distribution with standard deviation $\sigma = 0.5$. Subsequently, we assess the performance of these classifiers on the test set, perturbed by Gaussian noise using different values of σ. The outcomes are detailed in Table 2. We observe that the AT classifier's performance declines as the value of σ increases, whereas the CR classifier performs well only when the σ value for the test data is exactly the same as the value used during training. This phenomenon aligns with observations made by Kireev et al. (2022), indicating that employing Gaussian data augmentation during training, such as in CR, results in the classifier overfitting to the noise at the value of σ used during training. Classifiers trained using AT do not exhibit this overfitting behavior.

Next, we evaluate the certified test accuracy of the smooth classifier derived from the aforementioned base classifiers, presenting the results in Table 3. Certified accuracy at a given ℓ_2 radius, denoted as r, represents the proportion of test samples with a certified radius (as per Equation 2) greater than r. Recall that a smooth classifier makes predictions through majority voting over outputs from multiple noisy copies of a given input (see Equation 1). Consequently, selecting an appropriate value for the noise parameter σ for the smoothing process, one that the base classifier can "handle", is crucial. We make this selection informed by the findings presented in Table 2. The CR base classifier is smoothed using $\sigma = 0.5$, consistent with the value used during training, aligning with established practices in randomized smoothing literature (Cohen et al., 2019). In the case of the AT base classifier, we opt for $\sigma = 0.01$ since the base classifier demonstrates sharp drop in performance at higher σ values. Once again, we observe noteworthy variations in performance depending on the training method of the base classifier. The CR smooth classifier reports non-zero certified test accuracy at various radii, whereas the AT smooth classifier exhibits no certified robustness, i.e., the certified accuracy for any ℓ_2 radius $r > 0$ is negligible.

Empirical methods, such as AT, have been extensively scrutinized in the literature, with previous studies (Tsipras et al., 2019; Moosavi-Dezfooli et al., 2019; Ilyas et al., 2019; Kireev et al., 2022) unveiling various distinctive properties introduced by these methods. Given the close conceptual connection between certified robustness and empirical robustness, there might be a tendency to extrapolate findings in the context of empirical robustness without verification. However, the results presented in this section emphasize that these two classes of methods exhibit more dissimilarity than one might presume, and straightforwardly translating findings between them may lead to erroneous assumptions. Therefore, as part of our contributions through this study, we diligently validate the conclusions drawn in the realm of empirical robustness and transfer learning to certified robustness (Section 4.3).

4 Transfer Learning for Adversarially Robust ML

Commercial systems are becoming increasingly reliant on AI. However, adversarial attacks remain an ever-present issue when considering the trustworthiness of these systems. Unfortunately, training models with high adversarial robustness using current methods requires access to large amounts of labeled data (Schmidt et al., 2018), which is hard to achieve in many deployment scenarios, even in the actively studied image domain. Except for public datasets such as ImageNet, most vision tasks may only have a handful of labeled data samples for training.

In non-robust scenarios, transfer learning is one solution to alleviate the need for abundant training data for a given task. It involves pre-training on a data-rich (source) task followed by fine-tuning on the low-data downstream task to achieve state-of-the-art performance. Unfortunately, the relationship between transfer learning and adversarial robustness has only been studied in one specific scenario, when the downstream task has abundant labeled training samples and *empirical* adversarial robustness is the property of interest. To our knowledge, there are no works that explore using transfer learning to enable the deployment of empirically robust models on small-scale datasets. Furthermore, there are no works that study the relationship between transfer learning and *certified* adversarial robustness.

We present the first comprehensive study on the utility of transfer learning towards adversarial robustness. In Section 4.1, we describe our experiment setup. In Sections 4.2 and 4.3, we examine the benefits of transfer learning in the context of empirical and certified robustness in a range of data availability scenarios. Here, we use different pre-training methods (robust and non-robust), and perform fine-tuning robustly. In Section 5, we will examine the need for robustness training during the different phases of transfer learning, *i.e.,* pre-training and fine-tuning.

4.1 Setup

In this section, we describe our experimental setup. Additional implementation details are available in Appendix A.

Dataset and Model. For pre-training (supervised and self-supervised), we use the standard ImageNet dataset. For fine-tuning, we use a suite of 12 downstream datasets (Kornblith et al., 2019) often used in transfer learning literature. Training is done using a ResNet-50 classifier. All images are scaled to 224×224 in order to be compatible with ImageNet pre-trained weights.

Threat Model. We measure the adversarial robustness with respect to a white-box ℓ_2 adversarial attack. Our choice of adversary is motivated by the fact that randomized smoothing (Cohen et al., 2019), our choice of certified robustness method, defines robustness in the ℓ_2 space. This enables us to easily compare both adversarial metrics during evaluation.

Supervised Training. As a baseline for comparison, for every downstream task, we train a randomly initialized model using only the downstream task's labeled data. When studying the effects of transfer learning on empirical robustness, we use Adversarial Training (**AT**) (Madry et al., 2018) for baseline training, pre-training, and fine-tuning. AT uses a PGD attack with $\epsilon = 0.5$, step size $= 2\epsilon/3$, and 3 steps. We note that higher values of ϵ will only result in reducing the overall performance of the models. When studying the effects of transfer learning on certified robustness, we use Consistency Regularization (**CR**) (Jeong &

Table 4: Evaluating the benefits of pre-training for empirical adversarial robustness. Given a target task, we train three ResNet-50 classifiers: one using random weight initialization and two using weights pre-trained on a source task (ImageNet). Pre-training is performed using supervised (adversarial training) and self-supervised (SimCLR) objectives. During fine-tuning, the full network is trained using AT. Pre-training improves empirical adversarial robustness across all target tasks.

Target Task	**Random Init.**		**Sup. Pre-Training**		**Self-Sup. Pre-Training**	
	SA (%)	**RA** (%)	**SA** (%)	**RA** (%)	**SA** (%)	**RA** (%)
Food	74.5	62.3	81.6	69.2	82.2	68.6
CIFAR-100	71.8	62.5	80.1	70.6	80.9	70.3
CIFAR-10	93.3	88.8	95.8	91.7	95.9	91.2
Birdsnap	65.2	50.8	61.8	48.3	60.4	44.4
SUN397	51.0	41.7	55.5	44.4	59.0	44.3
Caltech-256	61.4	54.4	70.6	62.5	76.8	65.4
Cars	88.3	83.0	87.9	82.2	85.8	76.1
Aircraft	76.4	68.6	77.9	69.6	76.3	64.6
DTD	54.3	48.1	65.8	59.7	72.6	58.9
Pets	73.2	63.3	86.9	78.4	88.6	74.5
Caltech-101	66.7	61.5	88.5	83.1	91.9	83.6
Flowers	78.0	72.6	93.7	90.1	93.7	86.1

Shin, 2020) for baseline training, pre-training, and fine-tuning. For CR, we use $\sigma = 0.5$, number of Gaussian noise samples $m = 2$, $\lambda = 5$, and $\eta = 0.5$.

Self-supervised Training. Due to its popularity in current literature, we study the benefits of self-supervised pre-training on adversarial robustness. Unfortunately, most existing adversarially robust self-supervised methods (Jiang et al., 2020; Fan et al., 2021; Luo et al., 2022) have not been evaluated on ImageNet, but on smaller datasets instead. The one method we found that uses ImageNet (Gowal et al., 2020) does not have code publicly available. Thus, we use the SimCLR (Chen et al., 2020b) training method, a contrastive learning approach.

Evaluation. For measuring the robustness of empirically robust models during evaluation, we measure accuracy against the autoPGD attack (Croce & Hein, 2020) with $\epsilon = 0.5$, *i.e.,* robust accuracy (**RA**). Prior work (Croce & Hein, 2020) has demonstrated that AutoAttack, a more comprehensive attack for evaluating, only slightly reduces the robust accuracy (only a difference of $\mathbf{0.71}\%$ in RA computed using AutoAttack and AutoPGD) and we found AutoAttack is significantly slower.

For measuring the robustness of certifiably robust models, we use the certification process proposed by Cohen et al. (2019) and report the fraction of inputs with certified radius (Equation 2) greater than $\epsilon = 0.5$, called certified robust accuracy. Additionally, we report the average radius around an input within which the model's prediction remains consistent, denoted Average Certified Radius (**ACR**). For both evaluations, we also report the accuracy on the clean test set, *i.e.,* standard accuracy (**SA**).

4.2 Empirical Adversarial Robustness

Prior work (Hendrycks et al., 2019; Chen et al., 2020a) has demonstrated that, unlike with standard generalization, empirical robustness benefits from transfer learning for well-represented downstream tasks. We begin our study by validating their findings and then extending them to a wider range of data availability scenarios. On a suite of 12 target tasks, we train three versions of a ResNet-50 classifier: (i) using randomly initialized weights, (ii) using pre-trained weights obtained by performing Adversarial Training (AT) (Madry et al., 2018) on ImageNet, and (iii) using pre-trained weights obtained by performing SimCLR (Chen et al., 2020b) on ImageNet. The standard accuracy (SA) and robust accuracy (RA) of the resultant classifiers are reported in Table 4.

First, we see that, as prior work also demonstrated (Hendrycks et al., 2019; Chen et al., 2020a), transfer learning using a model pre-trained using AT improves performance (SA) and robustness (RA) on well-represented downstream tasks (*i.e.,* CIFAR-10, CIFAR-100, and Food). However, our experiments also show that pre-training with AT improves SA and RA even on under-represented downstream tasks (*e.g.,* Flowers, Pets, and Caltech-101). On average, across all tasks, pre-training with AT improves SA and RA relative to random initialization by 11.4% and 12.6%, respectively. We also note that SimCLR pre-training yields consistent improvements in SA and RA, averaging 14.1% and 9.9%, respectively. While improvements in SA were expected, the improvements in RA are surprising given that SimCLR, unlike other self-supervised methods we surveyed (Jiang et al., 2020; Fan et al., 2021; Luo et al., 2022), does not specifically design its objective function with adversarial robustness in mind.

We suspect that improvements in RA due to transfer learning are largely due to the overall improvement in SA rather than the robustness being "transferred" from the source task (ImageNet) to the target tasks. On Birdsnap, for example, both pre-training methods result in lower SA, which is mirrored by lower RA compared to random initialization. In Figure 2, we plot the relative increase in RA vs. the relative increase in SA due to pre-training. We observe a strong linear correlation between the two quantities for both the pre-training methods, with R^2 value of 0.98 for AT and 0.94 for SimCLR.

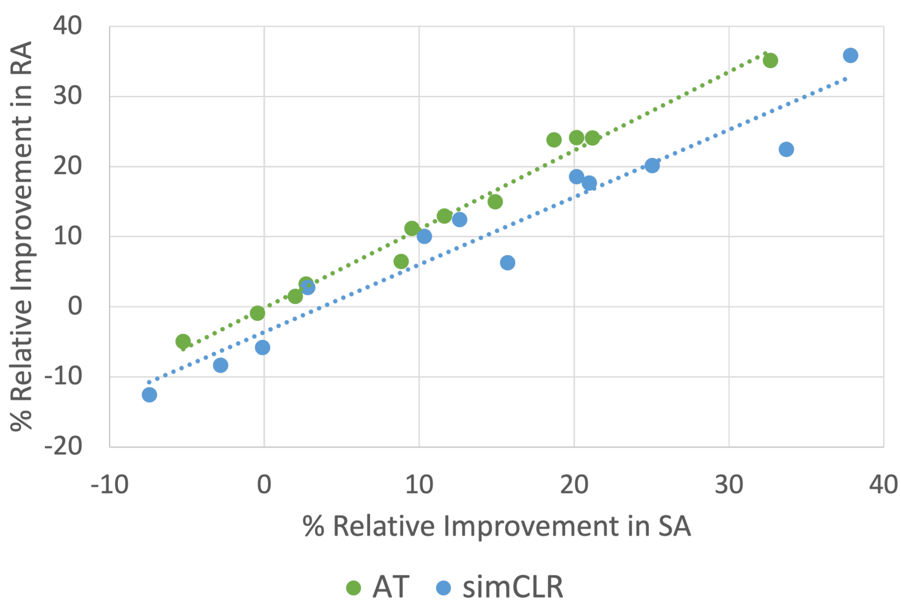

Figure 2: Plotting the improvement (%) introduced by pre-training relative to random initialization across all 12 target tasks. Improvement in RA is linearly correlated with improvement in SA for both pre-training methods, with R^2 value of 0.98 for AT and 0.94.

4.3 Certified Adversarial Robustness

To the best of our knowledge, there exist no works that explicitly study the utility of transfer learning in the context of certified adversarial robustness for either supervised or self-supervised pre-training. As before, we train three versions of a ResNet-50 classifier on each target task: (i) using randomly initialized weights, (ii) using pre-trained weights obtained by performing Consistency Regularization (CR) (Jeong & Shin, 2020) on ImageNet, and (iii) using pre-trained weights obtained by performing SimCLR on ImageNet. In order to achieve certified robustness during inference, we convert the ResNet-50 classifiers into smooth classifiers following Equation 1. The standard accuracy (SA), certified robust accuracy (RA), and Average Certified

Table 5: Evaluating the benefits of pre-training on certified adversarial robustness. Given a target task, we train three ResNet-50 classifiers: one using random weight initialization and two using weights pre-trained on a source task (ImageNet). Pre-training is performed using supervised (consistency regularization) and self-supervised (SimCLR) objectives. During fine-tuning, the full network is trained using CR. In all three cases, training on target tasks is performed using consistency regularization. Similar to empirical adversarial robustness, pre-training improves certified adversarial robustness across all target tasks.

Target Task	**Random Init.**			**Sup. Pre-Training**			**Self-Sup. Pre-Training**		
	SA (%)	**RA** (%)	**ACR** (ℓ_2)	**SA** (%)	**RA** (%)	**ACR** (ℓ_2)	**SA** (%)	**RA** (%)	**ACR** (ℓ_2)
Food	63.0	53.9	0.891	63.2	53.5	0.874	64.4	57.6	0.923
CIFAR-100	70.0	62.8	1.075	70.8	65.2	1.101	72.8	65.0	1.089
CIFAR-10	89.6	86.0	1.508	93.4	89.2	1.601	93.2	90.4	1.619
Birdsnap	42.0	34.7	0.538	41.6	32.4	0.504	41.0	35.3	0.541
SUN397	37.0	32.5	0.519	42.3	37.4	0.586	44.1	36.2	0.585
Caltech-256	54.0	47.4	0.835	60.9	57.3	1.001	65.4	58.5	1.000
Cars	81.9	77.5	1.358	79.1	73.9	1.285	77.7	70.1	1.158
Aircraft	70.1	63.4	1.065	68.1	60.9	1.022	69.0	61.4	0.991
DTD	44.9	39.4	0.699	50.2	45.3	0.790	55.5	49.8	0.849
Pets	66.7	61.8	1.068	70.8	64.5	1.088	75.2	67.2	1.089
Caltech-101	62.8	58.3	1.019	78.6	76.0	1.339	80.3	73.7	1.300
Flowers	75.2	72.7	1.306	87.5	82.0	1.538	84.6	78.7	1.407

[*] The above results are generated by evaluating a smooth classifier. This entails performing the computationally expensive process of certification, which scales poorly with input dimension. Since all our datasets are ImageNet size (*i.e.*, 224×224), we follow the standard practice (Cohen et al., 2019) and perform certification using only 500 evenly spaced images in the test set.

Radius (ACR) of the smooth classifiers are reported in Table 5. We compute these quantities using the prediction and certification process described by Cohen et al. (2019).

We observe that supervised and self-supervised pre-training improves performance (SA) and certified robustness (RA and ACR) on downstream tasks. Pre-training with CR results in an average relative improvement of 7.1%, 7.5%, and 7.3% compared to no pre-training on SA, RA, and ACR, respectively. Similarly, Pre-training with SimCLR results in an average relative improvement of 9.9%, 9.1%, and 6.9% compared to no pre-training on SA, RA, and ACR, respectively. As before, we note that the improvements in RA and ACR are not necessarily due to the "transfer" of robustness of the pre-trained model. Rather, the improvement in SA seems to result in an overall increase in RA and ACR. In Figure 3, we plot both the relative improvement in SA vs. RA and SA vs. ACR from pre-training and see a strong linear correlation between these quantities. For CR pre-training, the R^2 value for linear correlation between SA and RA is 0.92, and between SA and ACR is 0.94. For SimCLR pre-training, the R^2 value for linear correlation between SA and RA is 0.89, and between SA and ACR is 0.86.

5 Discussion

In Section 4, we demonstrated that a robust transfer learning pipeline is an effective method to train robust models, especially on downstream tasks with small amounts of labeled data. In fact, our self-supervised pre-training results highlight that a large **labeled** pre-training dataset is also unnecessary. However, there remains a question as to which parts of the robust transfer learning pipeline need to use robust training methods. As robust training methods impose a higher training overhead compared to non-robust training methods (Shafahi et al., 2019a; Vaishnavi et al., 2022), we perform two additional experiments to understand which parts of the transfer learning pipeline must use robust training methods.

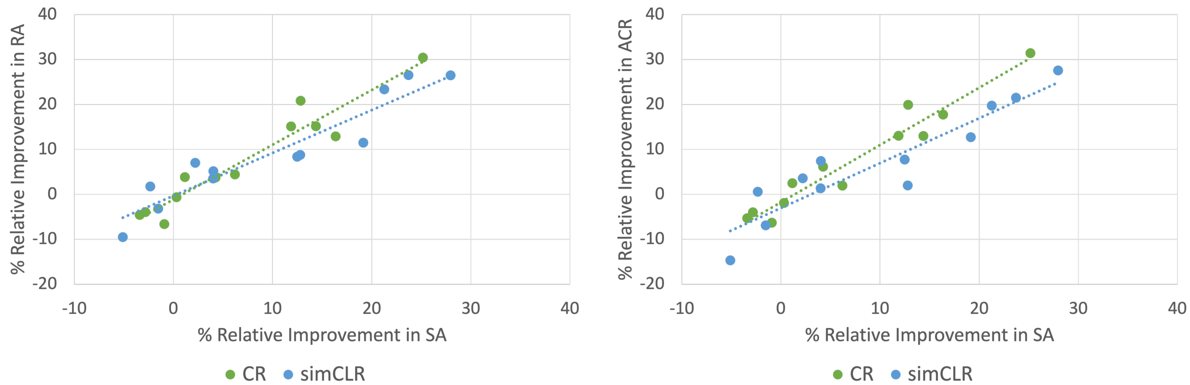

Figure 3: Plotting the improvement (%) introduced by pre-training relative to random initialization across all 12 target tasks. Improvements in both RA (left) and ACR (right) are linearly correlated with improvement in SA for both pre-training methods. For the left plot, R^2 values for CR and SimCLR are 0.92 and 0.89. For the right plot, R^2 values are 0.94 and 0.86.

5.1 Is Robust Pre-training Necessary?

Transfer learning is designed to improve standard performance on downstream tasks. In Section 4, we observed a strong linear correlation between improvements in SA and RA. This observation suggests that the robustness of the pre-trained model may be irrelevant. The SimCLR results provide further evidence as this training method does not optimize for robustness, and the models trained with it are not empirically or certifiably robust. Using the same experimental setup as in Section 4, we pre-train a ResNet-50 model using standard training (**ST**), *i.e.,* minimizing the cross entropy loss, which is also a non-robust pre-training method like SimCLR. We still use robust fine-tuning of the full network. In Table 6, we measure the empirical and certified robustness of models pre-trained with ST on two downstream datasets and compare it to pre-training with SimCLR and the respective robust pre-training method. We only see minor performance differences when using ST and SimCLR compared to a robust pre-training method, suggesting that robust pre-training is unnecessary for improving robustness on the downstream task.

5.2 Is Robust Fine-tuning Necessary?

In our initial experiments with a robust pre-trained model, we found that we could not use standard training and fine-tune the entire model. The resulting model exhibited neither empirical nor certified robustness as it was biased towards maximizing standard performance. However, Shafahi et al. (2019b) showed that it was possible to train an empirically robust network if standard fine-tuning was only done on the last model layer, thus freezing the rest of the model which was pre-trained using AT. The intuition is that the frozen layer of the model pre-trained with AT acts as a robust feature extractor that can be fine-tuned non-robustly while preserving robustness. Their method results in a less robust model compared to robust fine-tuning but is computationally more efficient. Thus, we replicate their experiments for certified robustness by first pre-training a ResNet-50 network on ImageNet using Consistency Regularization (CR) with $\sigma = 0.5$ and then fine-tuning the final layer only on CIFAR-10 and CIFAR-100 using Standard Training. During inference, we convert the ResNet-50 classifier into a smooth classifier (with $\sigma = 0.5$) following Equation 1 to measure certified robustness.

In Table 7, we report the performance and robustness of our ResNet-50 classifiers when converted in a smooth classifier with $\sigma = 0.5$. We observe that on both datasets, non-robust fine-tuning of the last layer results in a classifier with trivial standard accuracy (SA), robust accuracy (RA), and average certified radius (ACR). Recall from Equation 1 that a smooth classifier g_θ performs prediction by taking majority voting over several copies of a given input x sampled from the distribution $\mathcal{N}(x, \sigma^2 I)$. Thus, the base classifier should be trained using a noisy distribution (*i.e.,* $\sigma = 0.5$). Standard fine-tuning is equivalent to training with $\sigma = 0$. Thus, the smooth classifier's performance suffers. We see that if we instead use $\sigma = 0$, the SA of the smooth classifier is restored, though it has zero RA and ACR (follows directly from Equation 2). From these results,

Table 6: Effect of the pre-training method on empirical and certified robustness. The full network is fine-tuned using AT and CR, respectively. Robustness is not a requirement during pre-training in order to observe improvement in robustness on downstream tasks.

Task	**Empirical Robustness**			**Certified Robustness**			
	Pre-Training	**SA (%)**	**RA (%)**	**Pre-Training**	**SA (%)**	**RA (%)**	**ACR (ℓ_2)**
CIFAR-10	ST	95.4	91.2	ST	93.0	88.6	1.584
	SimCLR	95.9	91.2	SimCLR	93.2	90.4	1.619
	AT	95.8	91.7	CR	93.4	89.2	1.601
CIFAR-100	ST	78.5	68.1	ST	70.2	60.6	1.050
	SimCLR	80.9	70.3	SimCLR	72.8	65.0	1.089
	AT	80.1	70.6	CR	70.8	65.2	1.101

Table 7: Studying whether certified robustness is preserved on fine-tuning the final layer of a pre-trained model non-robustly using standard training (*i.e.,* $\sigma = 0$). Using different values of σ during training and inference causes the smooth classifier to exhibit poor SA, RA, and ACR.

Task	$\sigma = \mathbf{0.5}$			$\sigma = \mathbf{0.0}$		
	SA (%)	**RA (%)**	**ACR (ℓ_2)**	**SA (%)**	**RA (%)**	**ACR (ℓ_2)**
CIFAR-10	8.4	5.4	0.073	91.0	0.0	0.000
CIFAR-100	0.4	0.4	0.008	75.6	0.0	0.000

we conclude that robust fine-tuning is a necessary step for robust transfer learning to avoid catastrophic forgetting of robustness on the downstream task. Although Shafahi et al. (2019b) demonstrate a potential alternative for this finding in the context of empirical robustness, it significantly lowers the performance and robustness of the fine-tuned model, and as we demonstrated, it doesn't extend to certified robustness.

6 Conclusion

In summary, our research demonstrates that transfer learning, typically employed to enhance classifiers' standard generalization on tasks with limited labeled training data, can effectively contribute to improving adversarial robustness on downstream tasks, even when the sample complexity of robust generalization is significantly higher than that of standard generalization. This result holds irrespective of the amount of training data available for the downstream task.

Our experiments reveal that utilizing non-robust training methods during pre-training can still yield benefits for adversarial robustness on downstream tasks, provided robust training methods are employed during fine-tuning. We show for the first time that, contrary to traditional beliefs, the gains in robustness on the downstream task can be attributed to an increase in standard accuracy—a byproduct of transfer learning—rather than a direct "transfer" of robustness from the source to the target task.

We also find that pre-training can be performed using unlabeled data only by leveraging self-supervised training methods. Across 12 downstream tasks, employing (non-robust) self-supervised pre-training on ImageNet enhances average empirical and certified robustness by 9.9% and 6.9%, respectively. Our work stands as the first demonstration of training certifiably robust classifiers on tasks with extremely limited labeled training data.

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018. 3

Maximilian Augustin, Alexander Meinke, and Matthias Hein. Adversarial robustness on in-and out-distribution improves explainability. In *European Conference on Computer Vission (ECCV)*, 2020. 3

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings, 2012. 3

Yoshua Bengio, Frédéric Bastien, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, et al. Deep learners benefit more from out-of-distribution examples. In *Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2011. 3

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *10th ACM workshop on artificial intelligence and security*, 2017. 3

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 2020. 3

Rich Caruana. Learning many related tasks at the same time with backpropagation. *Advances in Neural Information Processing Systems*, 7, 1994. 3

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020a. 1, 2, 7, 8

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 2020b. 2, 3, 7

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019. 1, 4, 5, 6, 7, 9, 16

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, 2020. 7

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on computer vision and pattern recognition (CVPR)*, 2009. 3

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library). https://github.com/MadryLab/robustness, 2019. 15

Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *Advances in neural information processing systems*, 2021. 7, 8

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

Sven Gowal, Po-Sen Huang, Aaron van den Oord, Timothy Mann, and Pushmeet Kohli. Self-supervised adversarial robustness for the low-label, high-data regime. In *International Conference on Learning Representations*, 2020. 7

Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*, 2022. 3

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 2020. 3

Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *International Conference on Computer Vision*, 2019. 1, 2, 3, 15

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF conference on computer vision and pattern recognition*, 2020. 3

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, 2019. 1, 2, 3, 7, 8

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 3

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3, 6

Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 2020. 1, 4, 5, 6, 8, 15, 16

Jongheon Jeong, Sejun Park, Minkyu Kim, Heung-Chang Lee, Do-Guk Kim, and Jinwoo Shin. Smoothmix: Training confidence-calibrated smoothed classifiers for certified robustness. *Advances in Neural Information Processing Systems*, 2021. 1, 4

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *Advances in neural information processing systems (NeurIPS)*, 2020. 7, 8

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, 2017. 4

Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. In *Uncertainty in Artificial Intelligence*, 2022. 2, 5, 6

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *IEEE/CVF conference on computer vision and pattern recognition*, 2019. 1, 2, 6, 15

Rundong Luo, Yifei Wang, and Yisen Wang. Rethinking the effect of data augmentation in adversarial contrastive learning. In *International Conference on Learning Representations*, 2022. 7, 8

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 3, 5, 6, 7

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 2009. 3

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 15

Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sébastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, 2019. 1, 4

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *Advances in Neural Information Processing Systems*, 2020. 15

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, 2018. 6

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 3

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 2019a. 9

Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. In *International Conference on Learning Representations*, 2019b. 10, 11

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 2020. 3

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representation (ICLR)*, 2019. 3, 6

Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better: Illustration on image classification. In *International Conference on Learning Representations*, 2020. 3

Pratik Vaishnavi, Kevin Eykholt, and Amir Rahmati. Accelerating certified robustness training via knowledge transfer. *Advances in Neural Information Processing Systems*, 2022. 9

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 2020. 3

Yutaro Yamada and Mayu Otani. Does robustness on imagenet transfer to downstream tasks? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 2014. 1, 3

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*, 2020. 1, 4

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019. 3

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 4

A Implementation Details

To promote reproducibility, we provide all necessary implementation details in this Appendix. Statistics regarding all the datasets we used in our experiments are provided in Table 8. To train deep neural networks, we use the open-source PyTorch library (Paszke et al., 2019). For adversarial training, we use the open-source Robustness library (Engstrom et al., 2019) developed by Madry Lab. For autoPGD attack evaluation, we use the AutoAttack official code.[1] For training and evaluation using the randomized smoothing framework, we use the code provided by Jeong *et al.* (Jeong & Shin, 2020).[2] All the code and instructions for replicating our experiments are available at `[REDACTED]`.[3]

Table 8: Statistics for all datasets used in our experiments.

Dataset	# Train Images	# Classes	# Test Images	Skip	# Certified Images
ImageNet	1,281,167	1,000	50,000	100	500
Food	75,750	101	25,250	50	505
CIFAR-10/100	50,000	10/100	10,000	20	500
Birdsnap	32,677	500	8,171	16	511
☼	19,850	397	19,850	39	509
Caltech-256	15,420	257	15,189	30	506
Cars	8,141	196	8,041	16	503
Aircraft	6,667	100	3,333	6	556
DTD	3,760	47	1,880	4	470
Pets	3,680	37	3,669	7	524
Caltech-101	3,030	101	5,647	11	513
Flowers	2,040	102	6,149	12	512

Input Pre-processing. For all experiments, we fix the dimension of the input image to 224×224. For cases where the image is of smaller resolution (*i.e.,* CIFAR-10 and CIFAR-100), we upscale it first during the input pre-processing stage. The complete set of pre-processing steps we perform are as follows:

```
TRAIN_TRANSFORMS = transforms.Compose([
    transforms.RandomSizedCrop(224),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize()
])

TEST_TRANSFORMS = transforms.Compose([
    transforms.Scale(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize()
])
```

We follow prior works (He et al., 2019; Kornblith et al., 2019; Salman et al., 2020) and only use normalization for the ImageNet, CIFAR-10, and CIFAR-100 datasets.

[1] https://github.com/fra31/auto-attack
[2] https://github.com/jh-jeong/smoothing-consistency
[3] Redacted to honor the double-blind review process.

Training. When training from scratch, we perform hyperparameter tuning by performing grid search over lr $\in \{0.1, 0.01, 0.05, 0.001\}$, batch size $\in \{256, 128, 64, 32\}$, and weight decay $\in \{1e-04, 1e-03, 1e-02\}$. Before terminating training, the learning rate is decayed twice by a factor of 0.1 when the performance on validation set doesn't improve for 30 epochs. For ImageNet pre-training, we use publicly available weights for Adversarial Training[4] and SimCLR.[5] Since ImageNet pre-trained weights are not publicly available for the Consistency Regularization method, we generate them ourselves using hyperparameter details provided by the authors (Jeong & Shin, 2020). For all training, we use the Stochastic Gradient Descent (SGD) optimizer.

Certification Using Randomized Smoothing. During certification, we use $\sigma = 0.5$ and follow Cohen *et al.* (Cohen et al., 2019) for all other hyperparameters, *i.e.,* $N_0 = 100$, $N = 100,000$, and failure probability $\alpha = 0.001$. Also following prior works, we certify about 500 test images for each dataset, by skipping every n^{th} image in the complete test set (see Table 8 for skip factor used).

[4] https://github.com/microsoft/robust-models-transfer

[5] https://github.com/facebookresearch/vissl/blob/main/MODEL_ZOO.md