# Fine-tuning for Randomised Smoothing

**MSc Thesis** *(Afstudeerscriptie)*

written by

**Abra Ganz**

under the supervision of **Prof. Wilker Aziz, Prof. Martin Vechev, Marc Fischer** and **Mark Müller**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

**MSc in Logic**

at the *Universiteit van Amsterdam*.

| **Date of the public defense:** | **Members of the Thesis Committee:** |
| --- | --- |
| *TBD* | Prof. Ekaterina Shutova |
| | Prof. Wilker Aziz |
| | Prof. Martin Vechev |
| | Saba Amiri |



INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

**Abstract**

Deep neural network classifiers have become powerful tools permeating various domains of human life such as cancer detection and self-driving cars. Yet neural classifiers are vulnerable to deception via adversarial examples. To address this problem, randomised smoothing provides a viable method to certify a model's robustness to attacks without requiring any changes to the internal architecture. In order for robust, randomised-smoothing-based models to enter common usage, it is necessary for to be able to be train them without the use of excessive compute. One method which can achieve this is fine-tuning. This thesis is, to the best of the authors' knowledge, the first investigation into whether the robustness properties of RS-models are conserved through, or can be introduced via, fine-tuning. In this study, we demonstrate the feasibility of fine-tuning models for robustness on new datasets, surpassing the performance of models solely trained for robustness on that dataset. We investigate three methods to customize fine-tuning for the robustness context and find that adjusting the noise-level applied to images throughout training results in a near-uniform improvement in certified accuracy. Additionally, we find that using data augmentation in pre-training consistently enhances the performance of randomised smoothing models. Overall, this thesis establishes the feasibility of using fine-tuning alongside randomised smoothing to create robust models with minimal computational resources. Moreover, it highlights the potential of this method to enhance model robustness over non-fine-tuned methods, paving the way for practical implementation of robust models in real-world applications.

# Contents

**Contents** **3**

**1 Introduction** **1**
1.1 Motivation . . . . . 1
1.2 Scope . . . . . 2

**2 Background and related work** **5**
2.1 Background . . . . . 5
2.1.1 Transfer learning and fine-tuning . . . . . 5
2.1.2 Certifiable robustness . . . . . 6
2.1.3 Randomised smoothing . . . . . 6
2.1.4 Randomised smoothing implementation . . . . . 8
2.1.5 AugMix . . . . . 8
2.2 Related Work . . . . . 9
2.2.1 Fine-tuning . . . . . 9
2.2.2 The adversarial robustness and generalisation trade-off 9

**3 Experiments and Results** **11**
3.1 Set-up . . . . . 11
3.1.1 Datasets . . . . . 11
3.1.2 Baselines: Vanilla RS . . . . . 12
3.2 The feasibility of fine-tuning . . . . . 13
3.2.1 Transferring robustness via fine-tuning . . . . . 13
3.2.2 Introducing robustness via fine-tuning . . . . . 15
3.3 Adapting fine-tuning for the robustness context . . . . . 17
3.3.1 Learning rate . . . . . 17
3.3.2 Training regimes . . . . . 18
3.3.3 Data augmentation . . . . . 20
3.4 Applications beyond robustness transfer . . . . . 30
3.4.1 Transferring robustness into accuracy via fine-tuning . 31

3.4.2 Pre-training with augmentation . . . . . 31

**4 Discussion** **33**
4.1 Summary . . . . . 33
4.2 Further work . . . . . 35

**5 Conclusion** **37**

**6 Appendix** **39**
6.1 Training and testing . . . . . 39
6.1.1 Training . . . . . 39
6.1.2 Certification . . . . . 39
6.1.3 Prediction . . . . . 39
6.2 Augmentation algorithms . . . . . 39

**Bibliography** **41**

Chapter 1

# Introduction

## 1.1 Motivation

Neural networks have proven to be powerful tools, surpassing human performance at life-saving tasks such as cancer detection ((Lång et al., 2023)) as well as ordinary chores such as vacuuming Lee and Banerjee ((2015)). For this reason they are entering ever more domains of human life and being relied upon to a greater and greater degree. Yet these models are susceptible to being tricked via adversarial examples causing them to incorrectly classify input ((Dong et al., 2018; Goodfellow et al., 2015; Szegedy et al., 2014)). In the context of cancer detection or self-driving cars, such incorrect classification can have life-threatening consequences. Although there are safeguards in place, most defensive methods are heuristic, do not have any guarantee of robustness ((Athalye et al., 2018; Carlini and Wagner, 2017; Madry et al., 2018)), and have been shown to break when facing attacks from more powerful adversaries ((Athalye et al., 2018; Uesato et al., 2018)). The alternative to this inadequate empirical protection is certifiable robustness, a class of methods which prove the model in question is robust to adversarial attacks. There has already been a great degree of research into certifiable robustness ((Hein and Andriushchenko, 2017; Raghunathan et al., 2018; Mirman et al., 2018; Weng et al., 2018; Cohen et al., 2019; Lecuyer et al., 2019; Carlini et al., 2023)) but to the best of the authors knowledge (at time of writing) no method is yet in common usage.

A promising certifiable robustness method is *randomised smoothing*[1] (henceforth, 'RS') ((Cohen et al., 2019)), which utilises multiple passes of the image with varying degrees of added Gaussian noise to determine the most likely class of an image. It is especially desirable due to its ability to be prepended onto an existing model without requiring any changes to the internal architecture ((Hao et al., 2022)).

[1] "Randomized Smoothing" in the original.

In the modern image classification landscape, the most practical approach for training a top-tier model on an unfamiliar image dataset—without resorting to prohibitive amounts of time and computation—resides in the technique of fine-tuning. This method adopts and adapts a model that has undergone extensive pre-training on large-scale image datasets, thereby leveraging previously understood features and applying them to unfamiliar datasets. Consequently, this clears the path for significant time and computational efficiency.

Despite the ubiquity of fine-tuning in the modern image classifier development process, to the author's knowledge there has been no study into the feasibility of transferring or introducing RS robustness guarantees via fine-tuning. This thesis will investigate whether fine-tuning models for robustness is feasible and how the basic fine-tuning method could be adapted for the context of robustness. Finally, we explore two techniques inspired by our research into fine-tuning for robustness transfer that could be used more generally.

## 1.2 Scope

The thesis covers nine questions in three areas.

We first investigate whether transferring or introducing robustness via fine-tuning is possible:

1. Can a model trained for RS on one task (e.g. ImageNet), be fine-tuned for RS on another task (e.g. CIFAR10)?
2. Can a model trained for standard accuracy be fine-tuned for RS on the same dataset?
3. Can a model trained for standard accuracy be fine-tuned for RS on a different dataset?

We then investigate adaptions to the fine-tuning method in order to increase the performance of fine-tuned models in the robustness context:

4. Does using a different learning rate increase the performance of fine-tuning with randomised smoothing?
5. Does using a training regime increase the performance of fine-tuning with randomised smoothing?
6. Can the fine-tuning process be adapted via further data-augmentation to improve certified accuracy?
7. What aspect of augmentation leads to the increased accuracy of the resulting RS model?

Finally, having shown that fine-tuning for robustness transfer and introduction is feasible, and having explored some ways to improve the performance of fine-tuned models, we extend the investigation beyond the context of fine-tuning for robustness. During the course of our investigation two further questions naturally arose:

8. Can a model trained for RS on one task be fine-tuned for a non-RS applications on another task?
9. Can augmenting the base training data increase the performance of RS models?

Chapter 2

# Background and related work

## 2.1 Background

Here we briefly review the background necessary for the remainder of this document: first transfer learning, focusing on fine-tuning, then certifiable robustness and randomised smoothing in particular, then finally the data-augmentation method `AugMix`.

### 2.1.1 Transfer learning and fine-tuning

Transfer learning is the task of transferring knowledge of one dataset to a different dataset, usually with very little data from the new dataset ((Zhuang et al., 2021; Pan and Yang, 2010)).

Formally, we define a domain $\mathcal{D}$ as $\{\mathcal{X}, \mathbb{P}(X)\}$ where $\mathcal{X}$ is the feature space and $\mathbb{P}(X)$ for $X = \{x_1, \ldots, x_n\}, x_i \in \mathcal{X}$ is the marginal distribution over the feature space. Traditional classification tasks in supervised machine learning have the following objective: given a label space $\mathcal{Y}$ and data $(X, Y) \subseteq (\mathcal{X}, \mathcal{Y})$, learn a function $f : \mathcal{X} \to \mathcal{Y}$ such that $f$ satisfies $\arg\min_{y \in \mathcal{Y}} \mathbb{P}(f(x) = y$ and $y \neq y^*)$ where $y^*$ is the ground truth.

Then, the objective of transfer learning is: Given a source model $f_S$ that satisfies $\arg\min_{y \in \mathcal{Y}} \mathbb{P}(f(x) = y$ and $y \neq y_t)$ and that has been trained on data taken from a source domain $\mathcal{D}_S$, learn the target conditional probability distribution $\mathbb{P}(\mathcal{Y}_T \mid \mathcal{X}_T)$ in a target domain $\mathcal{D}_T$ where $\mathcal{D}_S \neq \mathcal{D}_T$.

Fine-tuning is one approach to transfer learning in which a model $f$ trained in a source domain $\mathcal{D}_S$ is trained, without modification, on further data $(X_T, Y_T) \subseteq (\mathcal{X}_T, \mathcal{Y}_T)$ drawn from the target domain $\mathcal{D}_T$.

Naturally, there are a variety of fine-tuning methods, the main distinction being between fine-tuning the entire model or freezing earlier layers and only fine-tuning the last layers. In this paper we focus on the former of these

two methods, fully fine-tuning, as for larger datasets this method has been shown to be more effective ((Yosinski et al., 2014)).

### 2.1.2 Certifiable robustness

As discussed in section 1.1, deep neural networks are vulnerable to adversarial examples. One way to mitigate this risk is to create a model that can be proven to be robust to adversarial examples.

However, the metric used to determine robustness to adversarial examples is debated within the community and various metrics are used ((Pydi and Jog, 2023)). One of the most commonly used metrics is the 'perturbation budget'. This measures the degree to which an image can be perturbed and still be correctly classified by the model. Intuitively, the greater the degree of perturbation an adversary uses, the more likely they will be able to push the perturbed input into another class and hence the more risky the perturbation. Hence we have the following definition:

**Definition 2.1 ($\varepsilon$–Robust)** *A model $f$ is said to be $\varepsilon-$robust if for all inputs $x$ and for all $\delta < \varepsilon$, $f(x+\delta) = f(x)$.*

In this work, we use an $\ell_2$-norm to determine the $\varepsilon$-ball around a given input $x$, but other norms are also used in the literature ((Goodfellow et al., 2015; Lecuyer et al., 2019; Madry et al., 2018)), most commonly $\ell_\infty$.

Throughout the thesis, the robustness of a model is measured by *certified accuracy*, which is the proportion of all points at radius $r$ that are both correctly classified and could be perturbed by up to $r$ without changing classification. Formally,

**Definition 2.2 (Certified Accuracy)** *Given data points $\{x_1, \ldots, x_N\}$ and classifications $\{f(x_1) = c_1, \ldots, f(x_N) = c_N\} \subseteq \mathcal{Y}$, the certified accuracy, $A_r$ for a given radius, $r$, is defined as:*

$$A_r = \frac{\sum_{\{i|x_i \in R\}} \mathbb{I}_{c_i = c_i^*}}{N}$$

*where $R = \{x_i$ such that $f(x+\varepsilon) = f(x)$ for $\varepsilon < r\}$ and $c_i^*$ is the ground truth classification of $x_i$.*

### 2.1.3 Randomised smoothing

Randomised Smoothing, introduced by Cohen et al. ((2019)), is a certifiable adversarial robustness tool that can be used with pre-existing architectures to achieve high robustness guarantees.

Consider classification problem mapping $\mathbb{R}^d$ to $\mathcal{Y}$. Then, given a classifier $f$ and data point $x$, RS creates a new classifier $g$ such that $g$ returns the

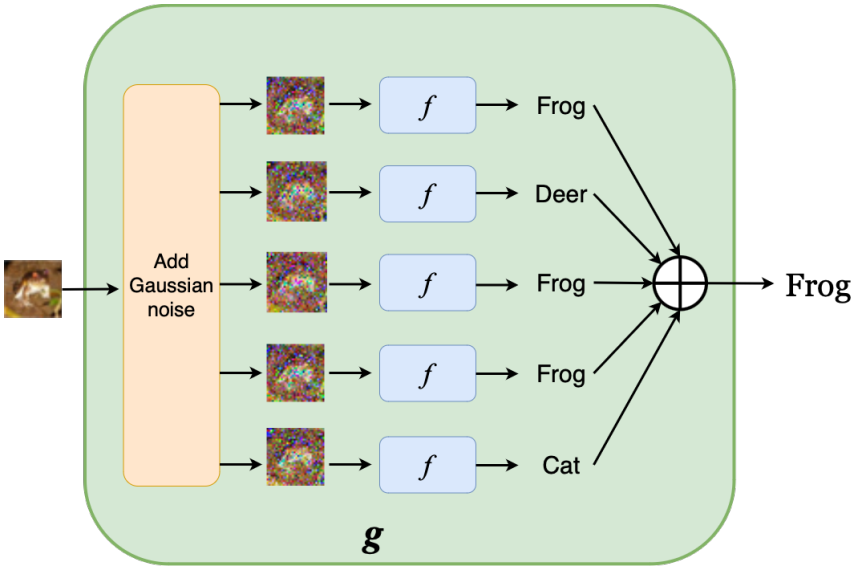



**Figure 2.1:** A schematic representation of the RS process, depicting how a base model, $f$, has been 'smoothed' to create an RS-model, $g$.

class most likely under $f$ when $x$ is perturbed by isotropic Gaussian noise. Specifically,

$$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(x+\varepsilon) = c), \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2 I) \tag{2.1}$$

A schematic diagram of this process can be seen in Fig. 2.1. Intuitively, the resulting classifier $g$ has smoothed classification boundaries in comparison to $f$ meaning that small changes to any input $x$ are less likely to result in a change of class.

Equation 2.1 leads to an $\ell_2$-robustness guarantee for the smoothed classifier $g$ around a point $x$. Specifically, suppose the most probable class, $c_A$, is returned with probability $p_A$ and the second most probable class, $c_B$, is returned with probability $p_B$. Let $\underline{p_A}$ be the lower bound of $p_A$, $\overline{p_B}$ be the upper bound of $p_B$, and $\Phi^{-1}$ be the inverse of the standard Gaussian CDF. Then:

**Theorem 2.3** (Adapted from ((Cohen et al., 2019))) *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Let $g$ be defined as in 2.1. Suppose $c_A, c_B \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy:*

$$\mathbb{P}(f(x+\varepsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\varepsilon) = c). \tag{2.2}$$

*Then $g(x+\delta) = c_A$ for all $|| \delta ||_2 < R$, where*

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})). \tag{2.3}$$

Note that Thm. 2.3 assumes nothing about $f$ meaning that there are no requirements place on the classifier - randomised smoothing can be applied to any modern deep architecture.

### 2.1.4 Randomised smoothing implementation

Unless otherwise specified we used the code for the original randomised smoothing paper ((Cohen et al., 2019)) for the base training. This follows the methodology set out in section 2.1.3 when certifying models with two adjustments:

**Monte-Carlo** The original definition of a smoothed classifier as in Eq. 2.1 necessitates finding the class with maximum probability across all possible noise levels drawn from $\mathcal{N}(0, \sigma I)$. However, as not all possible variations of $x + \varepsilon$ can be tested on a non-linear $f$ under finite time in any RS implementation, a Monte Carlo algorithm is used to predict the outcome of $g(x)$ to an arbitrary degree of certainty instead.

**Abstention** When predicting classes, the algorithm is allowed to abstain from making a prediction. This occurs if no single class is given as the output of $f(x + \varepsilon)$ much more than any other. The exact threshold for 'much more' is determined by the user.

Note that in order for 'smoothed models' (i.e. RS-models) to gain high levels of certified accuracy at a given radius $r$, they must be able to classify images with all possible perturbations of a degree less than $r$. Thus, the data that they are trained on must also have had Gaussian noise applied so that they are able to learn how to classify distorted images. All further training details can be found in Appendix section 6.1.1.

### 2.1.5 AugMix

First introduced by Hendrycks et al. ((2020)), AugMix is a data augmentation tool that aims to improve the robustness of image classifiers. If AugMix is applied to a given dataset, for every image in the dataset, between one and twelve randomly chosen augmentation-applications are applied to the image, each to varying degrees. We further modified the code in order for the possible augmentations to be chosen from a smaller subset than the original nine. Specifically, given an image, `image`, a list of augmentations, `Augmentations`, and parameter $\alpha > 0 \in \mathbb{R}$ the pseudocode is as follows:

**Algorithm 1:** AugMix

```
1  def AUGMIX (image : matrix, augmentations : list, α : float):
2      b ← B(α, α)
3      for i ← 0 in 3:
4          d ← Dir(α)
5          for j ← 0 in RANDOM.RANDINT(1,4):
6              AUGMENT ← RANDOM.CHOICE(augmentations)
7              aug_image ← AUGMENT(image)
8          mix ← mix + d · aug_image
9      mixed_image ← (1 − b)· image + b · aug_image
10     return mixed_image
```

Where '$B$' in line 2 refers to the Beta distribution and 'Dir' in line 4 refers to the Dirchlet distribution.

## 2.2 Related Work

### 2.2.1 Fine-tuning

Fine-tuning has been shown to not only enable the use of pre-trained models on new datasets with minimal additional training but also to improve model performance, especially in the realm of computer vision ((Razavian et al., 2014)). Other methods of knowledge transfer such as distillation have already been shown to increase the speed of certifiable robustness training by eight times ((Vaishnavi et al., 2022)). Thus far, fine-tuning has commonly been used as part of a greater method for creating a certifiably robust classified ((Lecuyer et al., 2019)), and Salman *et al.* showed that fine-tuning of adversarially trained models can improve transfer performance on downstream tasks over models trained on clean data ((Salman et al., 2020a)). This thesis, in contrast, will focus on whether fine-tuning can be used to transfer robustness properties and whether fine-tuning could be adapted in improve those robustness properties.

### 2.2.2 The adversarial robustness and generalisation trade-off

This trade-off has been extensively observed within robustness research and the prevalent view is that it is inherent. Indeed, various theoretical analyses have shown that in simple cases such as binary classification or linear regression the trade-off necessarily exists ((Javanmard et al., 2020; Tsipras et al., 2019)). Multiple methods have been proposed to combat the

trade-off such as robust self-training ((Raghunathan et al., 2020)), redefining the loss function ((Zhang et al., 2019)) and incorporating unlabeled data ((Carmon et al., 2019)). Fine-tuning specifically has been found to reduce a model's robustness to distribution shift ((Kumar et al., 2022; Radford et al., 2021)) suggesting that RS-trained models certifiable robustness guarantees would also decrease with fine-tuning. This paper investigates potential ways to mitigate a robustness-loss due to fine-tuning in section 3.2 as well as whether the robustness-accuracy trade-off can be used to trade robustness for accuracy in section 3.4.1.

Chapter 3

# Experiments and Results

In this chapter we cover the results of all questions posed in section 1.2. We show that transferring and introducing robustness via fine-tuning is feasible in section 3.2. After this, in section 3.3, we investigate methods to improve the performance of fine-tuning in the robustness context. Finally, we examine two questions that naturally arose throughout the investigation, but are beyond the original scope of fine-tuning for robustness. Before these main results, we briefly outline the datasets and method used for training and testing throughout the thesis in section 3.1.

## 3.1 Set-up

### 3.1.1 Datasets

We used the following datasets:

**CIFAR-10 (CIF)** A collection of 60,000 32x32 images separated into 10 equally-sized classes and labelled by hand. There are 50,000 training images and 10,000 test iamges. Introduced in Krizhevsky ((2009)).

**ImageNet** A collection of 1.8 million hand-labelled images of varying sizes separated into 1,000 classes with approximately 1,000 images per class. There are 1.2 million images in the training set, 100,000 in the test set, and 50,000 in the validation set. Introduced in Deng et al. ((2009)).

**CINIC-10** A combination of CIFAR-10 and the images in ImageNet that correspond to CIFAR-10 categories, reduced to 32x32. Contains 270,000 images in total, with equally sized train, validation and test splits, and each class within these subsets containing 9000 images. Introduced in Darlow et al. ((2018)).

**Images-10 (Im10)** The ImageNet subset of CINIC-10, thus having no overlap with CIFAR-10. It contains 210,000 images equally distributed over

classes and train, validation, and test splits.

**OpenImages-10 (OI)** The subset of the OpenImages dataset ((Kuznetsova et al., 2020)) that corresponds to the CIFAR-10 classes, containing 259,722 images in total resized to 32x32. Classes vary in size from 1,480 (frog) to 113,813 (car) with a median class size of 14,571. Although the classes are unbalanced, they are consistently unbalanced across the train and test sets, with test sets are on average 10% the size of the corresponding train class. Introduced in this thesis.

#### Data preparation

All images were cropped and horizontally-flipped at random before use using the corresponding functions from the `torchvision.transforms` library.

### 3.1.2 Baselines: Vanilla RS

To put later results in context, we include Fig. 3.1 showing models that have solely been pre-trained at various noise levels to give baseline certified accuracies (as in definition 2.2). As expected, models trained on lower levels of noise attain higher levels of certified accuracy at lower radii but drop off quickly in accuracy as the radius increases, while models trained at higher radii generally have a lower starting accuracy but also maintain their levels of certified accuracy better (e.g. in Fig. 3.1 we see that the model pre-trained on CIFAR-10 at $\sigma = 0.50$ has a 20% certified accuracy at radius = 1).

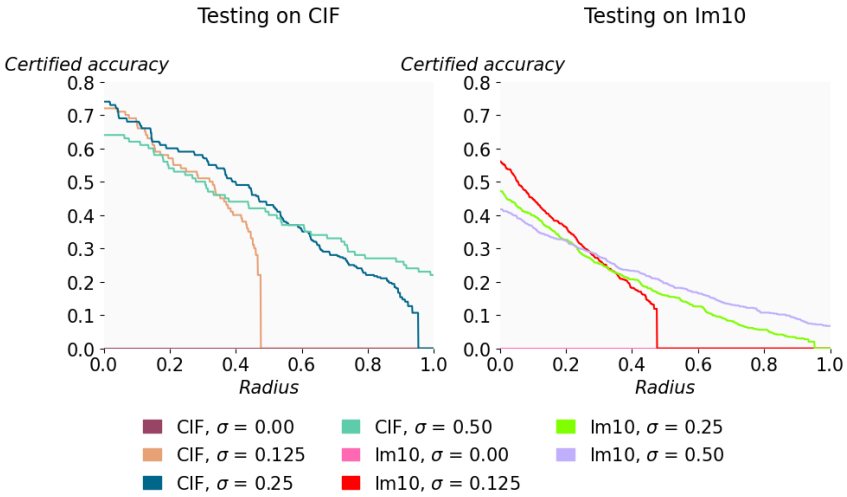



**Figure 3.1:** A plot showing how the performance of RS-based models pre-trained on CIFAR-10 and Images-10 vary given different noise levels and datasets. The models are trained and tested on the same dataset and the noise level used for training and testing is also the same.

It can also be seen that certified accuracy drops sharply to 0 past a certain point for each noise-level. This stems from the particularities of randomised smoothing as can be seen in Cohen et al. ((2019)): the maximum possible certifiable radius is a function of the noise level, $\sigma$, the number of samples $n$, and the allowed probability of returning an incorrect certified radius, $\alpha$, specifically, $R = \sigma\Phi^{-1}(\alpha^{\frac{1}{n}})$ where $\Phi^{-1}$ is the inverse of the standard Gaussian CDF. Thus at $\sigma = 0.25$, for example, since we certify with $\alpha = 0.001$ and $n = 100,000$, the maximum possible certifiable radius is $0.25 \cdot \Phi^{-1}(0.001^{\frac{1}{100,000}}) = 0.953$ (3 s.f.). Finally, note that for the models trained at $\sigma = 0$ the certified accuracy is 0 at every radius. This is due to the fact that during the certification phase, RS methods are used to classify the robustness of the model; this means feeding a large number of Gaussian-noise augmented versions of the image through the model and choosing the class with the overall highest number of predictions in its favour. However, as training with $\sigma = 0$ means that the model has not been exposed to any images that have had Gaussian noise added, the model fails to correctly classify the majority of Gaussian-noise augmented images that it sees and hence does not fulfill the criteria of $f(x + \varepsilon) = f(x)$ for the vast majority of data points $x$. This leaves the model with a certified accuracy of 0 for all radii.

## 3.2 The feasibility of fine-tuning

The first step of investigating whether we can utilise fine-tuning to create robust models quickly and with little data is to show that the method is feasible. There are two ways in which this can happen. First, a robust model is created via RS methods on a large dataset and is then fine-tuned, again using RS methods, for a new dataset where the robustness properties are maintained. Second, a pre-existing non-robust model is taken and then fine-tuned using RS methods in order to introduce robustness properties. These questions correspond to the first three questions of section 1.2 and they are answered in turn in this section.

### 3.2.1 Transferring robustness via fine-tuning

As proof that transferring robustness properties via fine-tuning is feasible, we experimented with transferring models between two datasets, CIFAR-10 and Images-10, which share the same classes but are distribution-shifted. Having trained an RS-model on one dataset for 90 epochs, we then fine-tune it for 30 epochs on the second dataset. The results can be seen in Fig. 3.2.

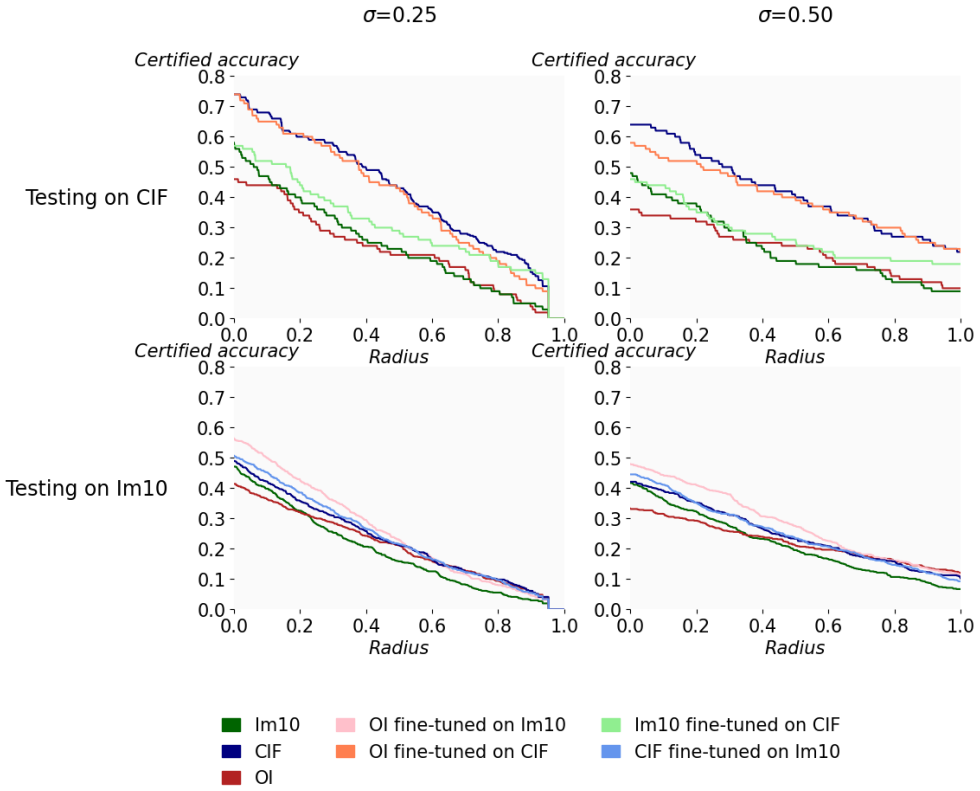



**Figure 3.2:** The top row shows various models tested on the CIFAR-10 (CIF) dataset and the bottom row shows the performance of models tested on the Images-10 (Im-10) dataset. For each combination of dataset and $\sigma$ there is: a baseline model that has only been pre-trained on that dataset; models in darker colours that have only been pre-trained on other datasets (■, ■, ■); models in corresponding lighter colours that were pre-trained on other datasets (■, ■, ■, ■) and then fine-tuned on the dataset at hand. Note that for each sigma, and hence each column, all models tested have been trained on the corresponding $\sigma$-noise level.

While CIFAR-10 (CIF), OpenImages (OI), and Images-10 (Im10) pre-trained models all achieve good results on other datasets (40-60% certified accuracy at $r = 0$), fine-tuning on the target dataset greatly improves certified accuracy: the fine-tuned models (■, ■, ■, ■) nearly uniformly outperform the pre-trained models (■, ■, ■) when tested on new datasets.

Note that the Images-10 dataset is significantly more complex than CIFAR-10, as Images-10 is a susbet of images from ImageNet which has pictures of varying dimensions and clarity. Thus, when testing on Images-10, the generally lower performance is to be expected as well as the closer grouping of models. Similarly, the complexity of Images-10 and OpenImages leads to models that were pre-trained on these datasets performing worse (as is especially noticeable when testing on the CIFAR-10 dataset - the top row in Fig. 3.2) and having a greater boost in performance by via fine-tuned. The

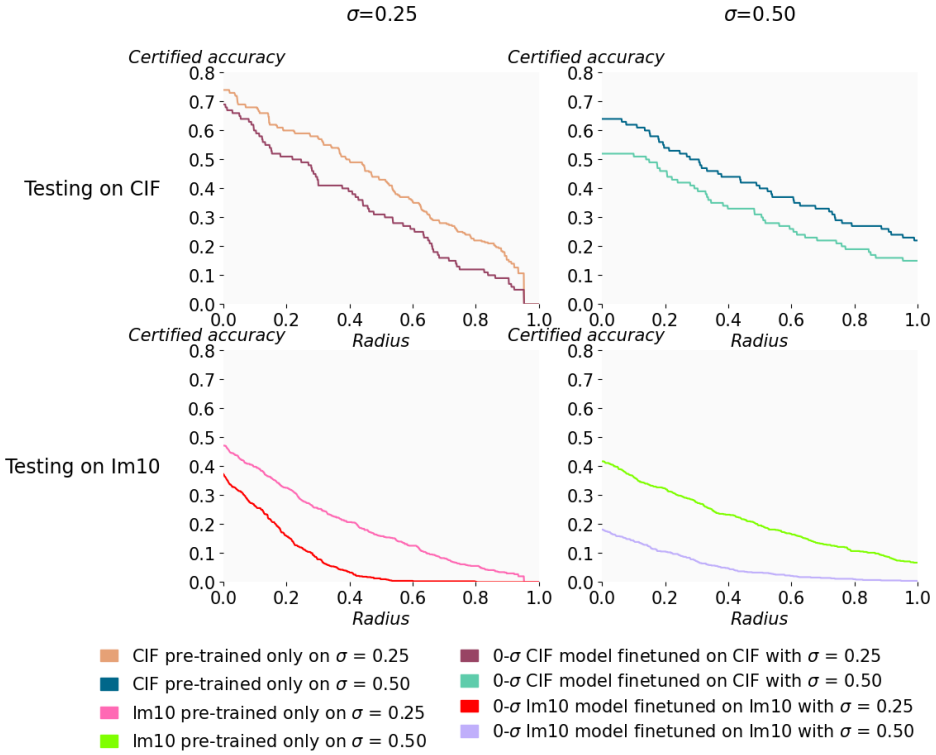



**Figure 3.3:** For each dataset and each noise-level, two models are trained. The first, a baseline model, is simply pre-trained on data which has Gaussian $\sigma$-noise applied to it (i.e. the normal RS training process). The second was pre-trained on data with no noise applied as if it were being trained purely for accuracy, it is then fine-tuned on the *same* dataset with Gaussian $\sigma$-noise applied.

fact that it is models trained on more complex datasets that increase their robustness most via fine-tuning is excellent sign for the viability of using fine-tuning methods with the large datasets commonly in use today.

Finally, we see that the fine-tuned models can perform not only on a par with models pre-trained on the target dataset (e.g. OI fine-tuned on CIF in the top row of Fig. 3.2) but even outperform them (OI fine-tuned on Im10 in the bottom row of Fig. 3.2).

Thus, we can conclude that fine-tuning is not only a feasible method for transferring robustness properties, but can in fact increase robustness levels by exposing the model to a greater variety of data.

### 3.2.2 Introducing robustness via fine-tuning

Now that we have shown it is possible to transfer robustness properties via fine-tuning, a natural next question is: is fine-tuning alone sufficient to introduce these robustness properties? Specifically, can a model that

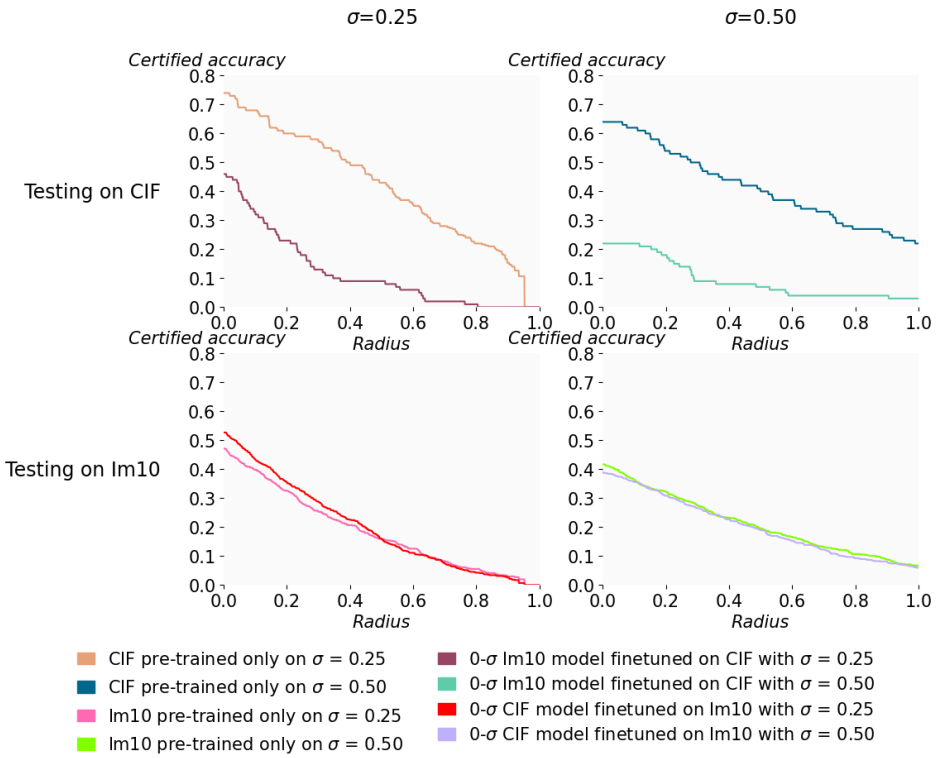



**Figure 3.4:** For each dataset and each noise-level, the figure compares a baseline model, pre-trained for 90 epochs on the dataset to be tested with Gaussian $\sigma$-noise applied to it, with a mode pre-trained on another non-noisy dataset and then fine-tuned on the target dataset with Gaussian $\sigma$-noise applied.

underwent standard pre-training be adapted via RS fine-tuning for the robustness setting. If so, this would allow us to take an off-the-shelf model and then quickly adapt it for the robustness setting via fine-tuning.

The possible scenarios in which this could be desirable are (a) in the same domain or (b) in a distribution-shifted domain. The results for fine-tuning to introduce robustness on the same domain are shown in Fig. 3.3 and for a distribution-shifted domain in Fig. 3.4.

As a proof of concept, it is clear that it is possible to achieve high levels of robustness purely via fine-tuning. Given a model which has no certified robustness (see the models which were evaluated with a flat line at $y = 0$ in Fig. 3.1), one can then apply 30 epochs of fine-tuning on noisy images to acquire relatively high levels of robustness (for the models trained and tested on CIFAR-10 the robustness levels nearly reach those of the models pre-trained on noisy images for 90 epochs). Note again that the levels of robustness achieved on the CIFAR-10 dataset are higher than on Images-10 since CIFAR-10 is a simpler dataset; hence, the it is quicker for a model to

adapt to a (noisy) version of it.

We see in Fig. 3.4 that, independent of the model being tested on, whether pre-training was on a simple or complex dataset seems to have the greatest effect. In Fig. 3.4, results much closer to those of a robustly pre-trained model are achieved for Images-10, but not for CIFAR-10 despite the expectation that fine-tuning on a simple dataset should allow for swifter adaption. The lowered results are likely due to the fact that it is much easier to learn a robust model from a simple dataset, as we see the CIFAR-10 pre-trained models out-perform the Images-10 ones. It appears that standard pre-training on a complex dataset such as Images-10 results in a model with very ragged decision boundaries, which are only partially removed via robust fine-tuning, thus resulting in lower certified accuracies.

Thus, we have found that a model trained for standard accuracy can be fine-tuned for RS on the same task as well as on a different task. However introducing robustness properties is significantly more effective when standard (i.e. non-robust) pre-training was done on a simple dataset.

## 3.3 Adapting fine-tuning for the robustness context

As we have shown that transferring and introducing robustness properties via fine-tuning is feasible, the next step for using fine-tuning to introduce robust models into common usage is to explore how we can adapt the fine-tuning process for the robustness context to increase fine-tuned models' certified accuracy. In this section we investigate three methods to improve the certified accuracy of fine-tuned models: using a different learning rate for fine-tuning; varying the noise-level throughout fine-tuning; augmenting the data used for fine-tuning.

Throughout this section we focus on transfer of robustness properties (i.e. all pre-trained models were trained using RS methods). We leave it to a future work to investigate the effectiveness of these methods for introducing robustness via fine-tuning from a standard model pre-trained solely for accuracy.

### 3.3.1 Learning rate

A common way to improve the performance of a model is to change its hyper-parameters ((Bergstra et al., 2011)). Up to this point, we have used a learning rate of 0.0001 for fine-tuning, which is simply the learning rate used for the final 30 epochs of pre-training. However there is of course the possibility that another learning rate during fine-tuning leads to the model achieving a higher certified accuracy. Hence we experimented with various different learning rates, showing 0.01 (the learning rate for the first 30 epochs

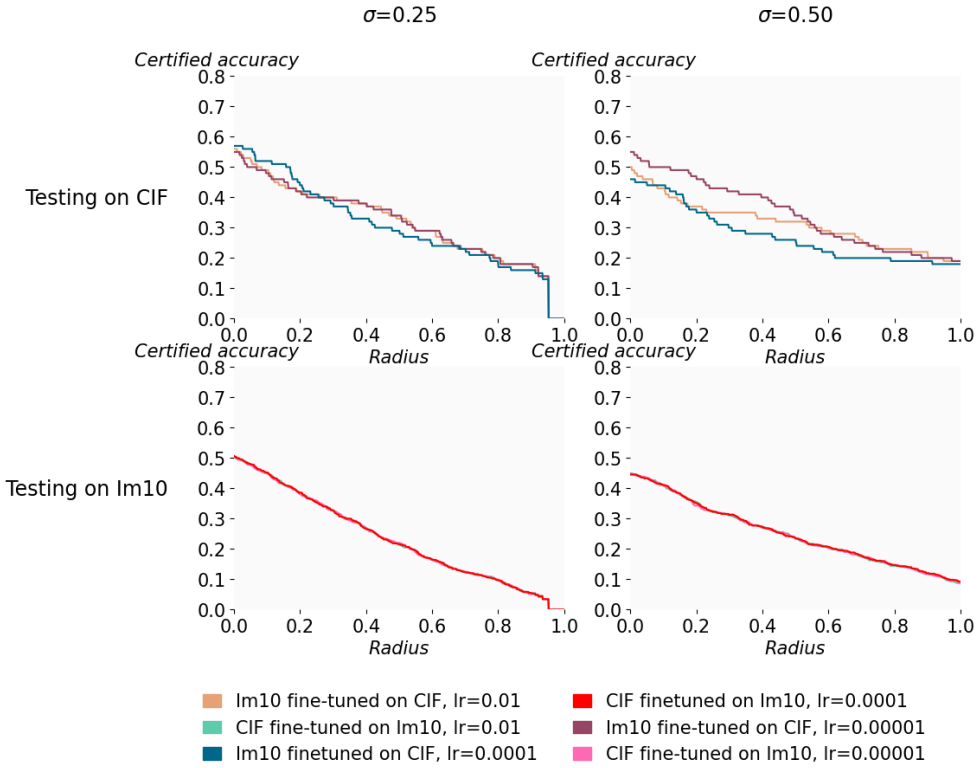



**Figure 3.5:** The figure compares models that are pre-trained on CIFAR-10 or Images-10 then finetuned on Images-10 or CIFAR-10 respectively. Three different learning rates ('lr') are used: 0.01, 0.0001, and 0.00001.

of fine-tuning), 0.0001 (the original fine-tuning learning rate), and 0.00001 in the figure below. These rates were chosen as they best maintain the balance between achieving a reasonable amount of fine-tuning and not completely retraining the model.

Surprisingly, there appears to be no difference at all in outcome when the models are pre-trained on CIFAR-10 and fine-tuned on Images-10. In the case of models fine-tuned and tested on CIFAR-10, on the other hand, there is a great variation in results, with both alternative learning rates of 0.01 and 0.00001 at times out-performing the original learning rate of 0.0001.

Overall, however, the fact that no learning rate consistently provides better results over a more complex dataset suggests that it is unlikely that changing the learning rate will result in a much improved performance.

### 3.3.2 Training regimes

Another hyper-parameter that could be changed in order to improve the robustness gained via fine-tuning is the noise level, $\sigma$. It is clear that different

levels of noise during training result in different accuracy levels for different radii, with lower $\sigma$ during training giving higher accuracy at lower radii and a higher $\sigma$ during training resulting in better accuracy at higher radii. Thus it is natural to ask: is it possible to gain the benefits of training at different noise levels during a single training run?

In order to do this, we experimented with using a training regime where we vary the noise level, $\sigma$, during the 30 fine-tuning epochs and, dependent on the variation of $\sigma$, refer to an 'increasing/decreasing-$\sigma$ regime'. As proof of concept, we experimented with a linearly increasing and decreasing level of noise. For the increasing-sigma regime we increased the noise level linearly from 0 to the chosen $\sigma$ over the course of 30 epochs with results shown in Fig. 3.6 . For the decreasing regime we decreased the noise level from $4\sigma$ to $\sigma$ over the 30 epochs with results shown in Fig. 3.7.

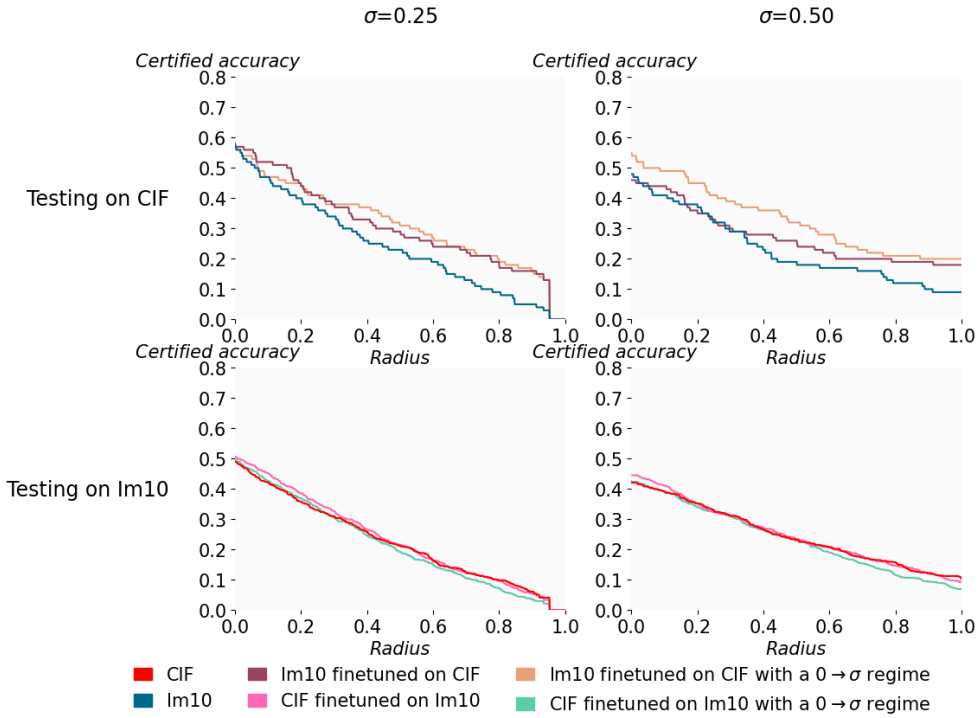



**Figure 3.6:** During fine-tuning the noise level increases linearly from $0 \to \sigma$ over 30 epochs.

In general, the results are as expected - the training regime in which $\sigma$ was increasing, which has more training at lower noise levels, has better performance at lower radii and vice-versa for the training regime with decreasing $\sigma$. Results are nearly *uniformly better*, however, for the decreasing-$\sigma$ regime, especially for the model fine-tuned on Images-10 at $\sigma = 0.25$ and the model

fine-tuned on CIFAR-10 at $\sigma = 0.50$. Surprisingly, even at low radii where we would expect it to have a lower certified accuracy, the decreasing-$\sigma$ regime nearly equals the performance of the other models.

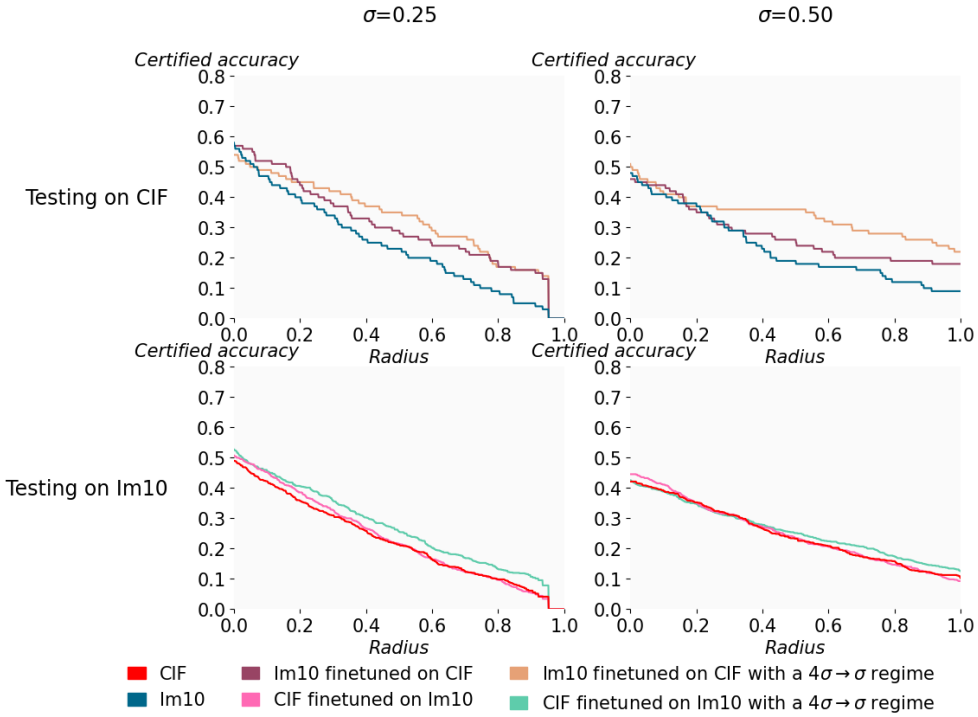



**Figure 3.7:** During fine-tuning the noise level decreases linearly from $4\sigma \to \sigma$ over 30 epochs.

Thus we have found at least preliminary results in favour of a $\sigma$ training regime increasing the performance of fine-tuning in the robustness setting.

### 3.3.3 Data augmentation

Beyond changing the level of noise during fine-tuning, we also explored changing the *type* of noise applied during fine-tuning. Rather than simply applying the same noise during the fine-tuning process as during the pre-training process (i.e. Gaussian), we experimented with applying `AugMix` data augmentation as well. The intuition being that the greater variety of images within a class the model is exposed to, the more robust the model becomes. This intuition is grounded by the fact that `AugMix` has been shown to improve model robustness in other non-RS contexts ((Hendrycks et al., 2020)).

The preliminary results in Fig. 3.8 show that by fine-tuning on another dataset that has been augmented with `AugMix`, certified accuracy is improved

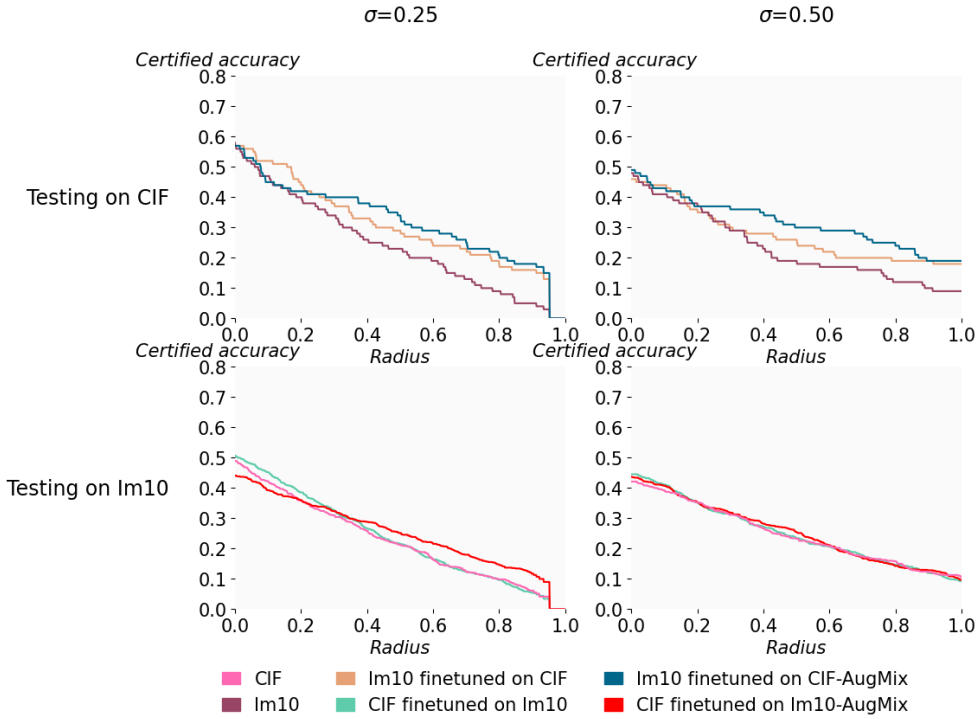



**Figure 3.8:** Models pre-trained on clean CIFAR-10 or Images-10 data and then fine-tuned on the alternative dataset using data augmented with the `AugMix` algorithm.

| Im10 fine-tuned on CIF | | CIF fine-tuned on Im10 | |
|---|---|---|---|
| **Augmentation** | **ACR** | **Augmentation** | **ACR** |
| autocontrast | 0.331 | equalize | 0.238 |
| equalize | 0.319 | autocontrast | 0.235 |
| posterize | 0.315 | posterize | 0.235 |
| shear_y | 0.306 | shear_x | 0.232 |
| rotate | 0.297 | translate_x | 0.229 |
| shear_x | 0.294 | rotate | 0.229 |
| translate_y | 0.292 | shear_y | 0.226 |
| translate_x | 0.283 | translate_y | 0.225 |
| solarize | 0.281 | solarize | 0.219 |

**Table 3.1:** The various augmentation-fine-tuned models ranked by average certified radius (ACR).

at higher radii. This is an exciting result, as it suggests that the robustness-accuracy trade-off is not the zero-sum game we had imagined.

In order to increase the fine-tuned models' performance yet further, we investigated what aspect(s) of `AugMix`-augmentation causes the the increase in certified accuracy. This corresponds to question 7 in in section 1.2.

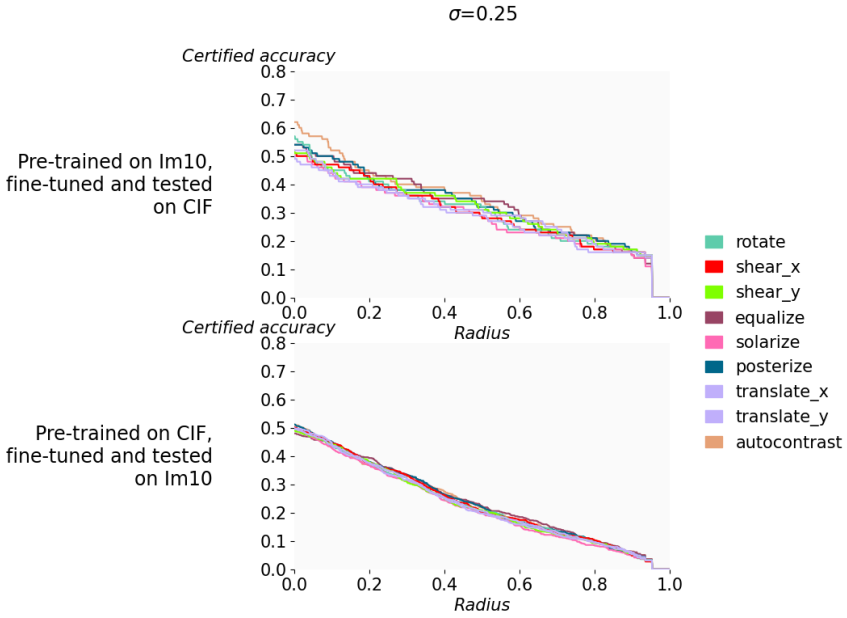



**Figure 3.9:** Models pre-trained on Images-10 and CIFAR-10 then fine-tuned on the other dataset augmented with one of the nine augmentations of `AugMix`.

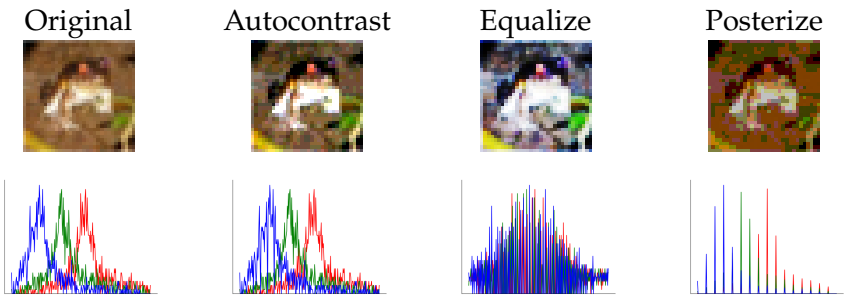



**Figure 3.10:** The top row shows the original CIFAR-10 image augmented in various ways, while the bottom row shows the histograms corresponding to the images.

We began by breaking down `AugMix` into its nine component augmentations and testing models fine-tuned on each of them in order to discover which had the greatest effect. This can be seen in Fig. 3.9. From this we hoped to deduce the underlying cause for the increased performance.

The variation in performance for the different augmentations can be seen in table 3.1. Note that when testing on CIFAR-10 there is a much greater spread in the performance as can be seen in Fig. 3.9. This phenomenon will be investigated further in *Dataset particularities*.

From table 3.1 we see that `autocontrast`, `equalize`, and `posterize` are the most effective augmentations in both directions. Figure 3.10 shows the effects

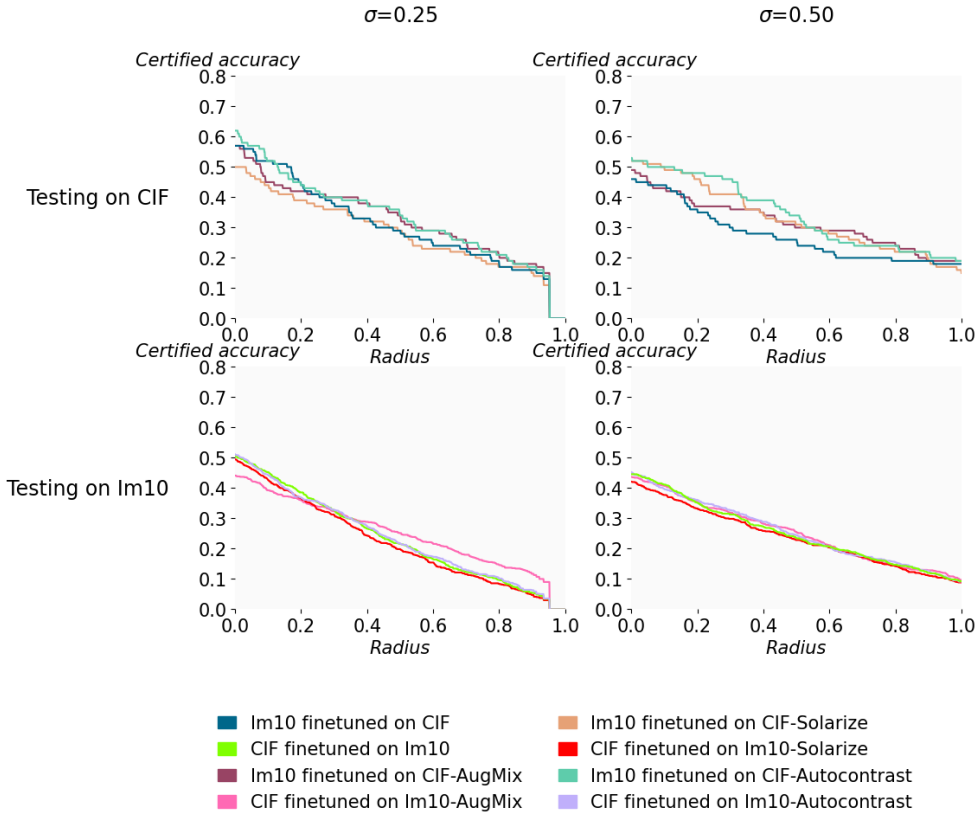



**Figure 3.11:** Fine-tuning an Images10 pre-trained model on CIFAR-10 data augmented with `autocontrast` notably improves model performance. However vice-versa does not particularly improve performance.

of the various augmentations on an example image, as well as the effects of the augmentations on the histograms of an image.

Indeed, by fine-tuning on data augmented with only a single one of these augmentations one achieves equal or better performance than using the whole range of `AugMix` augmentations. This can be seen in Fig. 3.11 where the performance of a model fine-tuned on `autocontrast`-augmented data is contrasted with a model that was fine-tuned on `solarize`-augmented data, the augmentation with the lowest performance in table 3.1. We can see that (except for the model fine-tuned on CIFAR-10 at $\sigma = 0.50$) the `solarize` models perform worse than the models fine-tuned on clean data. Hence, it is reasonable to conclude that the robustness benefits gained by fine-tuning on `AugMix`-augmented data are not simply an epiphenomenon arising from mixing augmentations, but likely to come at least in part from the benefits of training on data augmented with `autocontrast`, `equalize`, etc. We will thus investigate the underlying features of these augmentations in order to attempt to determine whence the observed increase in performance arises.

`autocontrast` 'stretches' an image's histogram so that the lightest pixel becomes white and the darkest becomes black. `equalize` applies a non-linear mapping to the histogram in order to smooth it. `posterize`, meanwhile, does not act directly on an image's histogram but reduces the number of bits for each color channel. The effects of all these functions on an image can be seen in Fig. 3.10 and further details can be found in Appendix section 6.2.

Since there is no property that the three functions clearly share, we investigated a variety of possible causes for the increase in performance:

1. Increased support
2. Histogram flattening
3. Effect of augmentation mixing
4. Dataset particularities

### Increased support

One potential cause is that some of the augmentations could help the algorithm distinguish between images by increasing range of the images' pixel intensities (support). Both `autocontrast` and `equalize` can increase the number of intensities in an image; the former by ensuring that every pixel intensity is reached and the later by smoothing the image's histogram.

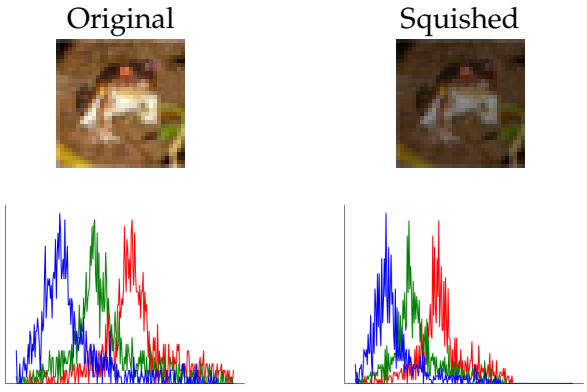



**Figure 3.12:** The effect of the `squish` algorithm on a CIFAR-10 image and its histogram.

In order to test this suggestion, we implemented an algorithm, `squish`, which decreased the support of images by decreasing the number of different intensities in images. This was implemented via adjusting the existing `autocontrast` function implemented in `ImageOps` - see algorithm 2. It can either be applied to a greyscale image or, with little adjustment, channel-wise. The effect of applying this algorithm to an image can be seen in Fig. 3.12 and the results of fine-tuning on a dataset augmented with this histogram-smoothing is seen in Fig. 3.13.

**Algorithm 2:** Squish

```
def MAPPING(image: matrix, map: list):
    new_image ← image
    for i, j in SIZE(image):
        new_image[i,j] ← map[image[i,j]]
    return new_image
def SQUISH(image: array, level: int):
    map ← []
    hi ← HIGHEST_INTENSITY(image)
    lo ← LOWEST_INTENSITY(image)
    scale ← LN((e−(hi−lo)·(1+level/10))/255)        /* Squish according to the
     width of the histogram */
    for i ← 0 in [0, 255]:
        i ← INT(scale · i)
        if i < 0:
            i ← 0
        elif i > 255:
            i ← 255
        map ← APPEND(map, i)
    return MAPPING(image, map)
```

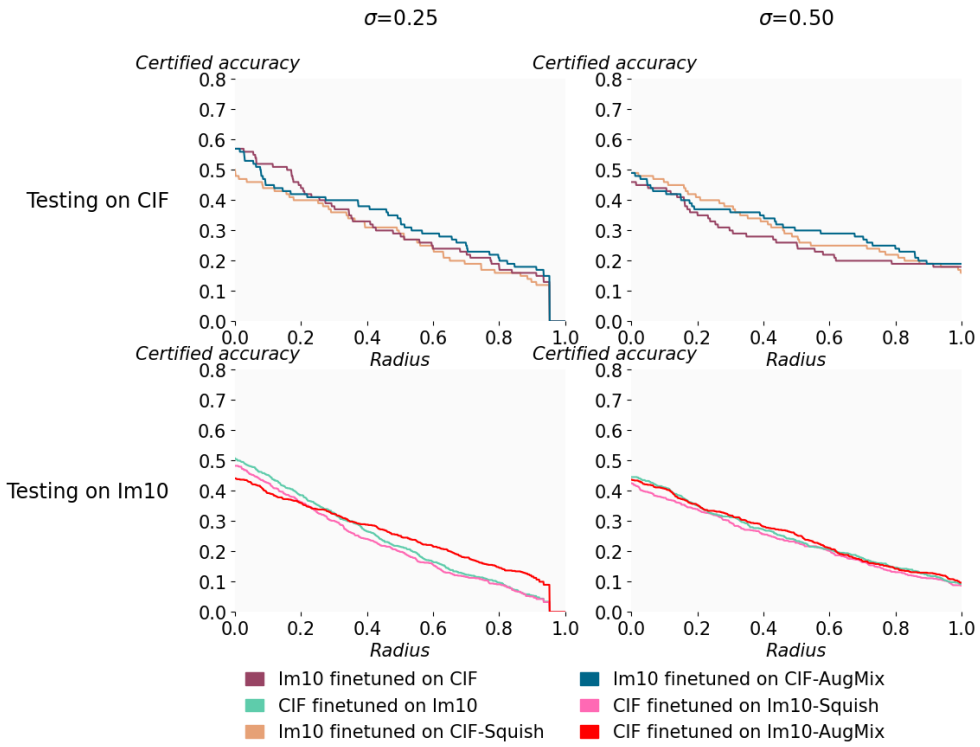



**Figure 3.13:** Models pre-trained on Images-10 and CIFAR-10, then fine-tuned on the opposite dataset. During fine-tuning data was: not augmented, augmented with `AugMix`, or augmented with `squish`.

Apart from the upper right-hand graph in Fig. 3.13, testing on CIFAR-10 at $\sigma = 0.50$, `squish` does not improve results beyond fine-tuning on clean data, and even slightly worsens performance in all other cases, especially at low radii. From this we can conclude that *decreasing* images support, as `squish` was designed to do, worsens model performance. From this, it seems likely that the support-increasing effects of `autocontrast` and `equalize` are indeed partially responsible for the overall improvement effects seen by fine-tuning data augmented with them, or on`AugMix` as a whole.

### Effect of augmentation mixing

A further possibility for why fine-tuning on `AugMix`-augmented data improves performance is that the variety of augmentations creates a more robust model. Hence, we also experimented by training on a subset of `AugMix` augmentations, using the top three performing augmentations according to table 3.1. The results are shown in Fig. 3.14.

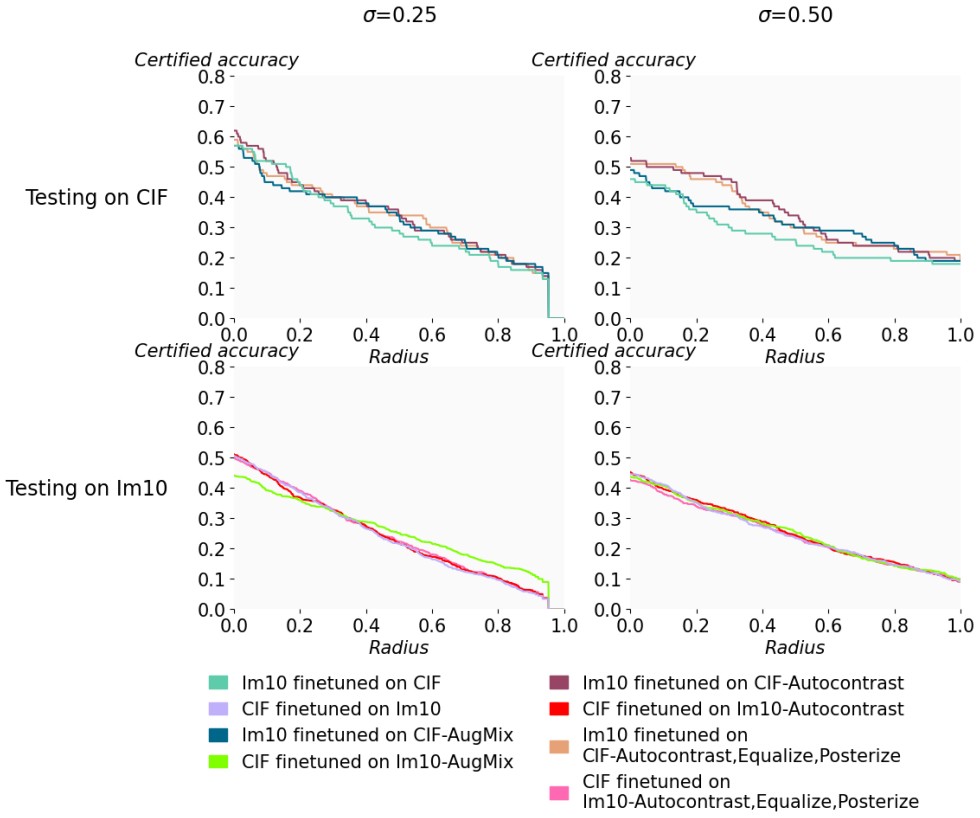



**Figure 3.14:** The figure compares models fine-tuned on a mix of three augmentations (`autocontrast`, `equalize`, and `posterize`) with those fine-tuned on only one (`autocontrast`) or all nine (AugMix). All models fine-tuned on Images-10 were pre-trained on CIFAR-10 and vice versa. As a baseline, models fine-tuned on clean data are given.

We see that by training on a mix of three augmentations, `autocontrast`, `equalize`, and `posterize`, performance is not improved compared to training on all `AugMix` augmentations, or on `autocontrast` alone. Indeed, when testing on CIFAR-10 (see the top row of Fig. 3.14), the model fine-tuned on `autocontrast`-augmented data nearly uniformly outperforms the model trained on three augmentations. Thus, as mixing different augmentations results in no apparent improvement over using a single augmentation, the improved performance of models fine-tuned on `AugMix`-augmented data does not stem from the mix of nine augmentations.

### Histogram flattening

Both `autocontrast` and `equalize` flatten an image's histogram, hence one hypothesis for why these functions improve fine-tuned performance is something to do with this flattening aspect. In order to test this hypothesis, we created a new flattening algorithm, given in algorithm 3, that manipulated the image in order for its histogram to be as close to completely flat as possible – that is, for every one of the 256 pixel intensities to have $N/256$ pixels. It can either be applied to a greyscale image or channel-wise; the results of applying the algorithm channel-wise to an image are shown in Fig. 3.16.

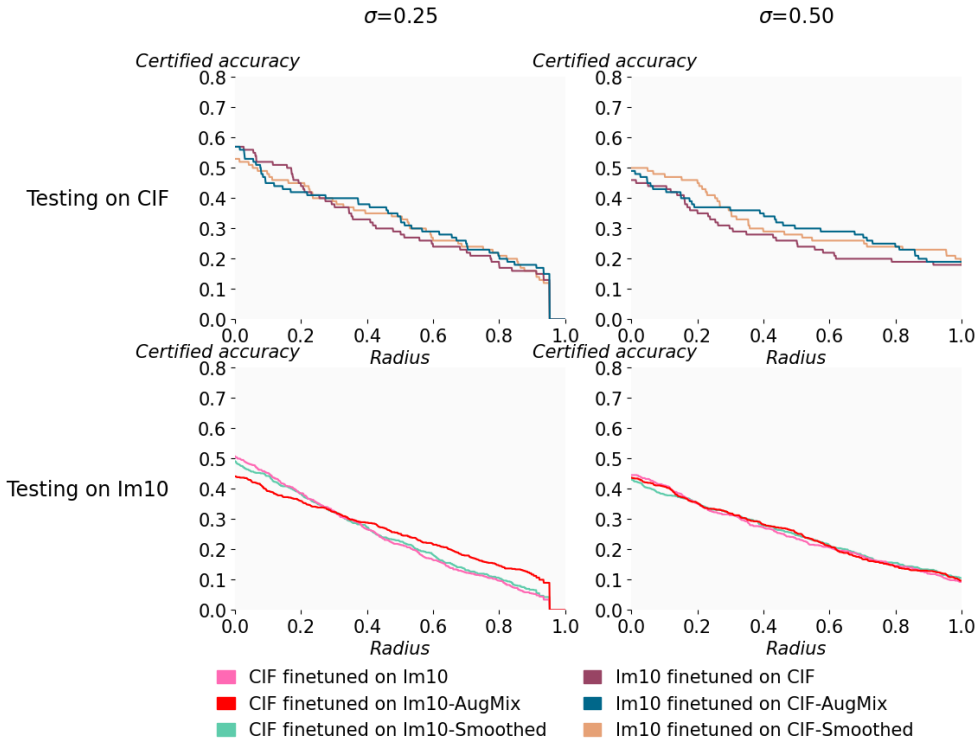



**Figure 3.15:** The effect of fine-tuning on smoothed data - i.e. images the histograms of which have been flattened.

**Algorithm 3:** Flattening

```
def FLATTENING(image : matrix):
    sorted_indices ← SORT_PIXEL_INDICES_BY_INTENSITY(image)
    num_pixels ← LEN(sorted_indices)
    num_indices_per_intensity ← ⌊num_pixels / 256⌋
    remaining_indices ← MOD(num_pixels, 256)
    for i in [0,255]:
        start_index ← i · num_indices_per_intensity
        end_index ← MIN((i + 1)·num_indices_per_intensity,
         num_pixels)
        if remaining_indices > 0:
            start_index ← start_index + i
            end_index ← end_index + i + 1
            remaining_indices ← remaining_indices − 1
        else:
            start_index ← start_index + i
            end_index ← end_index + i
        change_indices ← sorted_indices[start_index:end_index]
        for j in change_indices:
            list_image ← IMAGE_TO_LIST(image)
            list_image[j] ← i
    flat_image ← LIST_TO_IMAGE(list_image)
    return flat_image
```

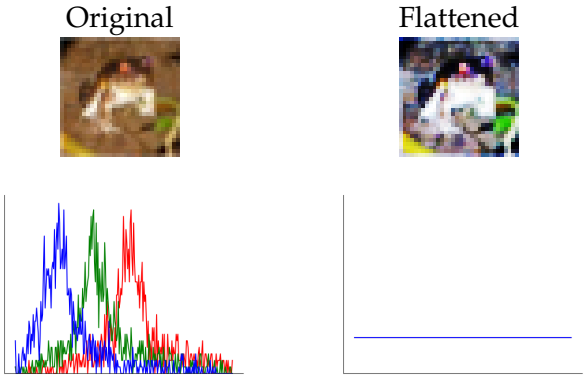





**Figure 3.16:** The effect of the flattening algorithm on an image and its histogram.

In Fig. 3.15 we can see the result of fine-tuning on images which had the histogram-flattening augment applied. While using this augmentation does have a noticeable impact on results, especially in the case of fine-tuning and testing on the CIFAR-10 dataset, it does not outperform the models fine-tuned on `AugMix`-augmented data. This suggests that it is *not* the histogram-flattening aspect of `autocontrast` and `equalize` that are contributing most to the improved performance of models fine-tuned on `AugMix`-augmented data.

### Dataset particularities

Finally, since testing on Images-10 resulted in models consistently more indistinguishable from each other, we also considered the hypothesis that the improvements stemming from `AugMix`-augmentation are a dataset-specific phenomenon that occurs only when pre-training on Images-10 and fine-tuning on CIFAR-10. To test this, we used a third dataset, OpenImages-10 (see section 3.1.1) which has a further distribution shift: instead of pre-training using CIFAR-10 and Images-10, we pre-trained on OpenImages-10 ('OI' in the key), then fine-tuned on CIFAR-10 and Images-10 using `AugMix`-augmented data.

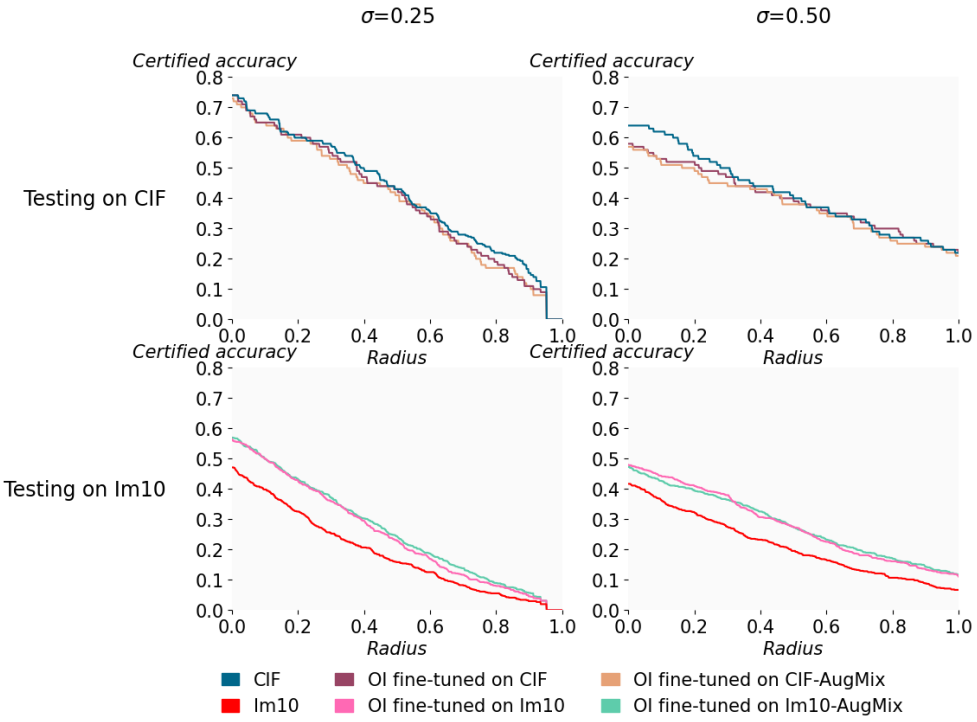



**Figure 3.17:** The figure compares OI pre-trained models fine-tuned on both clean and `AugMix`-augmented CIFAR-10 and Images-10 datasets. As a baseline we use a model pre-trained solely on the target dataset (CIFAR-10 or Images-10).

As preamble to the results, we note that OpenImages-10 is closer to Images-10 in makeup, since it is a larger and more complex dataset.

In Fig. 3.17 see a clear improvement from fine-tuning using `AugMix`-augmented data at $\sigma = 0.50$ on the CIFAR-10 dataset for both the models pre-trained on Images-10 and on OpenImages, however this is the only place where it is seen. On Images-10, the OpenImages pre-trained model appears to do equally well whether trained on `AugMix`-augmented or clean data, and the improvements exhibited at higher radii on Images-10 are minor. The fact that there are clear improvements when fine-tuning with `AugMix`-augmentation on CIFAR-10 (no matter the pre-training dataset) suggests that `AugMix`-augmentation improves performance only on less complex datasets. On datasets of greater complexity such as Images-10 there is likely already enough variety in the images that `AugMix` does not provide sufficiently more complexity to improve the model's robustness further.

Overall, we have to conclude that although increasing the support of images' histograms has a positive affect on fine-tuned models (as seen in section 3.3.3), the greatest difference in how augmented data affects a model arises from the dataset the model is tested on, and specifically the lack of complexity in the CIFAR-10 dataset. Thus when fine-tuning is done, an augmented version of the dataset will contain a greater variation allowing for better robustness. Unfortunately, this means that data augmentation does not appear to be a viable method for improving the robustness of common-usage fine-tuned models, except when a model is required for a relatively simple dataset.

## 3.4 Applications beyond robustness transfer

While we explored various methods for improving the robustness of fine-tuned models, two additional questions naturally arose which were outside the scope of our original investigation into fine-tuning for robustness. Firstly, just as we converted a model pre-trained for accuracy into one that was robust via fine-tuning (section 3.2.2), and not only robust but at certain radii more robust that the model that had been pre-trained for robustness (see the top row of Fig. 3.4), we questioned whether it was possible to do the reverse. That is, take a model that had been pre-trained for robustness and transform it via fine-tuning into one that performed well on standard accuracy metrics. Secondly, just as we had considered the effects of using data-augmentation during fine-tuning to increase models' certified accuracy, we decided to explore whether the same data-augmentation methods could be effective during *pre-training* as well. This section covers the methodology and results of these two questions.

| Im10 fine-tuned on CIF | | CIF fine-tuned on Im10 | |
|---|---|---|---|
| **Pre-training noise level** | **Accuracy** | **Pre-training noise level** | **Accuracy** |
| 0.00 | 0.689 | 0.00 | 0.820 |
| 0.25 | 0.569 | 0.25 | 0.580 |
| 0.50 | 0.523 | 0.50 | 0.600 |

**Table 3.2:** The table shows the accuracy results for six models. The first half pre-trained on Images-10 at noise levels $\sigma = 0, 0.25$, and $0.50$ then fine-tuned on CIFAR-10 with no noise (i.e. $\sigma = 0$). The second other half pre-trained on CIFAR-10 at noise levels $\sigma = 0, 0.25$, and $0.50$ then fine-tuned on Images-10 with no noise.

### 3.4.1 Transferring robustness into accuracy via fine-tuning

Previous literature has shown that adversarially trained models work particularly well for transfer learning Deng et al. ((2021)). Intuitively, this is because by improving robustness one necessarily improves the decision boundaries between classes and hence increases accuracy. Despite this, in actuality most papers observe decreased generalisation properties as robustness increases - the so-called adversarial robustness and general trade-off as discussed in section 2.2.2. Hence, we investigated whether higher robustness levels could improve accuracy after fine-tuning. The details of how accuracy was assessed can be found in section 6.1.3. Although we see a slight (2 percentage point) increase in performance on the Im10 dataset by pre-training at $\sigma = 0.50$ instead of $\sigma = 0.25$, this is outweighed by the 12 and 24 percentage point increases by pre-training on clean data. Thus we should conclude from these preliminary results, that fine-tuning for accuracy is not assisted by robustness properties.

### 3.4.2 Pre-training with augmentation

In section 3.3.3, we saw that `AugMix` sometimes improved the robust performance of fine-tuned models. Although we saw that this was due to the simplicity of the dataset fine-tuned on, we wanted to experiment further with using `AugMix` data-augmentation in conjunction with RS-models as method to increase robust performance. Specifically, we investigated whether it is possible use `AugMix` to achieve general improvements in RS-model performance *without* fine-tuning. Hence we pre-trained four models on the CIFAR-10 and Images-10 datasets at $\sigma = 0.25$ and $\sigma = 0.50$. The results of pre-training with `AugMix`- and autocontrast-augmented data are shown in Fig. 3.18.

From these results we see that that pre-training on augmented data nearly uniformly improves results. The complexity comes into which augmentation works best on which dataset. While `AugMix` outperforms no augmentation or `autocontrast` for Images-10, on CIFAR-10 it performs worse than training on clean data. On the other hand, data augmented with `autocontrast` nearly

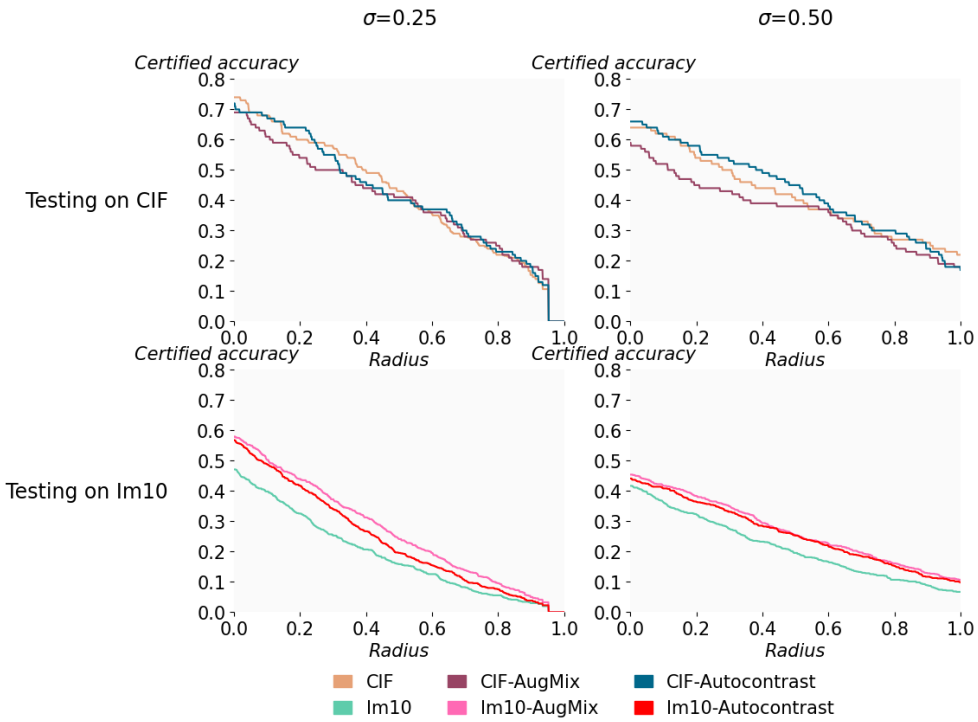



**Figure 3.18:** Models pre-trained on three versions of CIFAR-10 (top-row) and Images-10 (bottom row): clean data, `AugMix`-augmented data, and `autocontrast`-augmented data. Models were tested on the same model on which they were trained.

always performs as well as or better than training on clean data, even if it performs worse than `AugMix`-augmentation for Images-10.

Hence, the answer to the final question of this thesis is that augmenting the base training data can increase the performance of RS models.

Chapter 4

# Discussion

In this chapter we first summarise the work by recalling the original research questions we set out to answer and the conclusions we found. We then discuss potential future direction for research highlighting those we think are the most promising.

## 4.1 Summary

Firstly we set out to show that robustness transfer via fine-tuning for was feasible: We first investigated robustness transfer via fine-tuning:

1. Can a model trained for RS on one task (e.g. ImageNet), be fine-tuned for RS on another task (e.g. CIFAR-10)?

We saw in Fig. 3.2 that fine-tuning on a target dataset improves certified accuracy, especially at lower radii (by 30 percentage points in the case of OpenImages on CIFAR-10 at $r = 0$). We also saw that the more complex the pre-training dataset, the greater the performance increase from fine-tuning. CIFAR-10 increased only a few percentage points when fine-tuned on Images-10, while OpenImages increased by 15 percentage points at $r = 0$. Similarly, Images-10 increased by a maximum of 7 percentage points when fine-tuned on CIFAR-10 (at $r = .95$), while fine-tuning increased OpenImages' performance by 30 p.p. on CIFAR-10 at $r = 0$.

We then looked at introducing robustness via fine-tuning:

2. Can a model trained for standard accuracy be fine-tuned for RS on the same dataset?
3. Can a model trained for standard accuracy be fine-tuned for RS on a different dataset?

In this case, although the resulting fine-tuned models were robust, we discovered that standard pre-training on a simple dataset was the key to achieving

higher accuracies no matter the fine-tuning target dataset. Models pre-trained on Images-10 were significantly worse both when testing on Images-10 (quickly dropping to near 0 accuracy by $r = 0.4$ for both $\sigma$), and when testing on CIFAR-10 (dropping to 0.1 by $r = 0.4$ for both $\sigma$). When pre-training and fine-tuning on CIFAR-10, however, models achieved robustness levels consistently only approximately 10 p.p. lower than the model pre-trained for 90 epochs on CIFAR-10 (figures 3.3 and 3.4). When a CIFAR-10 pre-trained model was applied to Images-10, it achieved results nearly equal to and in some cases exceeding the model robustly pre-trained on Images-10 for both $\sigma$ levels.

Overall we saw that the answer to questions (1), (2), and (3) was that fine-tuning for robustness using randomised smoothing was feasible. Once we knew this, we then explored ways to adapt the fine-tuning method for the robustness context.

4. Does using a different learning rate increase the performance of fine-tuning with randomised smoothing?

The results of testing for this question were inconclusive. In Fig. 3.5 we see that the results of different learning rates varied greatly when testing on CIFAR-10 with both a lower and higher learning rate resulting in improvements, but had no effect when testing on Images-10. Thus we leave this for further investigation.

Another hyper-parameter that we experimented with was the noise level, $\sigma$.

5. Does using a training regime increase the performance of fine-tuning with randomised smoothing?

This was the most promising adaption to the fine-tuning process. While a regime increasing the noise level from 0 to the chosen $\sigma$ did not improve performance except when testing on CIFAR-10 for $\sigma = 0.50$ (see Fig. 3.6), a decreasing-$\sigma$ regime did. Indeed, introducing a training regime that decreased the noise level from $4\sigma$ to $\sigma$ resulted in uniform model improvement for both $\sigma = 0.25$ and 0.50, and on both CIFAR-10 and Images-10 (Fig. 3.7).

After adapting the hyper-parameters of learning rate and noise level, $\sigma$, we then explored augmenting the data used during the fine-tuning process:

6. Can the fine-tuning process be adapted via further data-augmentation to improve certified accuracy?

7. What aspect of augmentation leads to the increased accuracy of the resulting RS model?

Despite positive results at the beginning, as seen in Fig. 3.8, where `AugMix`-augmented data improved the fine-tuning process in three out of four cases, through an investigation into the causes of this improvement, we discovered

it was mostly due to the dataset used for testing. Although the size of the augmentation's support (in the pixel intensity space) did have an impact (see Fig. 3.13), it was not sufficient to account for the results of Fig. 3.8. Thus using data-augmentation during the fine-tuning process does not appear to be a feasible method to improve performance in a robustness context.

Finally, we explored two questions that naturally arose throughout the course of our investigation:

8. Can a model trained for RS on one task be fine-tuned for a non-RS applications on another task?
9. Can augmenting the base training data increase the performance of RS models?

Of these, we see in section 3.4.1 that the first proved fruitless, with standard pre-training instead of robust pre-training resulting in significantly (12 to 24 percentage points) higher accuracy. Pre-training with data-augmentation, however, provided near-uniform improvements in Fig. 3.18, especially when testing on the more complex Images-10 dataset, suggesting that more complex datasets could benefit more from this method. We recommend further investigation into the feasibility of this method to improve all randomised smoothing models.

Overall, this thesis has shown that it is feasible to use fine-tuning in conjunction with randomised smoothing to create robust models using little compute and that the performance of these models can be increased by using a decreasing-$\sigma$ regime. Further, by pre-training with `AugMix`-augmented data, the certified accuracies achieved by a model can be improved at nearly all radii, even without fine-tuning. The combination of these results paves the way for the practical implementation of robust models into common usage.

## 4.2 Further work

While we have shown that fine-tuning for robustness using RS is feasible, there is much work still to be done on improving the certified accuracy of these methods as well as improving the RS technique in general. Nevertheless, there are some areas in particular which we will point out as ripe for future work.

This thesis only addresses the adversarial robustness of a model as measured by an $\ell_2$ perturbation budget, as is common in current literature. However many other approaches use an $\ell_\infty$-norm Goodfellow et al. ((2015)); Kurakin et al. ((2018)), and it has already been shown that randomised smoothing can be extended to any $\ell_p$-norm Hayes ((2020)). We assume the observed trends to hold irrespective of the underlying norm, nevertheless, it would be natural to extend this work to other metrics.

Secondly, this thesis, used only one method of fine-tuning, where the weights throughout the whole model are adjusted for the new dataset. It is also common, however, to fine-tune by freezing earlier layers of the model and only adapt the weights of later layers to the new dataset in order to preserve more general features. This could potentially allow for greater conservation of robustness properties across the fine-tuning process and hence is also worth investigating.

Thirdly, one direction that randomised smoothing research has taken is to prepend a denoiser onto the original model, meaning that where originally a multitude of noisy images would be fed in to the model, they are now denoised before entering the model Salman et al. ((2020b)). This would distinctly change the training mechanism, with the original (non-RS) model needing much less Gaussian-augmented training data, and hence putting in doubt the effectiveness of the pre-training data-augmentation method we saw to work in section 3.4.2.

Finally, we hope to further the positive results we had in this thesis. Namely, that a decreasing-$\sigma$ regime during fine-tuning can increase the robustness of the resulting model and that pre-training with augmented data can improve the robustness of the resulting RS-model. For the former and obvious first step is to check whether a decreasing-$\sigma$ regime also improves performance for the introduction of robustness properties, rather than only the transferal. Beyond this, we only tested a very simple linearly decreasing regime. Other options, such as a quadratic or logarithmic decrease could result in more robust models, just as starting from a higher multiplier of $\sigma$ could also have an effect and both these options should be explored. In terms of using augmented data during pre-training, the root cause(s) should be investigated. Since we found an increase in performance when testing on Images-10 as well as CIFAR-10, this is unlikely to be a dataset-specific phenomenon as it was for the improvements seen when augmenting the data during fine-tuning (see *Dataset particularities*). However this also means that the results from the investigation into applying `AugMix`-augmentation during fine-tuning do not necessary carry over. Thus, such an investigation into the root cause(s) of the improved certified accuracy should be carried out in order to enable the creation of augmentations specifically tailored to the randomised smoothing context.

Chapter 5

# Conclusion

In this work we showed that creating robust models via RS-based fine-tuning is feasible. The most effective method was via robustness transfer, rather than introduction, where models which were pre-trained for robustness via RS on a source dataset then fine-tuned on a different target dataset achieved equal or higher (by up to 30 percentage points) robustness levels compared to those trained solely on the target dataset. Once we knew that it was feasible to transfer or introduce robustness via fine-tuning, we then explored ways to improve the performance of the fine-tuned models. We found that introducing a decreasing sigma training regime nearly uniformly increased models' performance. Finally, we also went beyond the original scope of fine-tuning for robustness and found that pre-training with `AugMix` data-augmentation provided near-uniform improvements in certified accuracy, especially when testing on the more complex Images-10 dataset. Overall, this thesis has shown that it is not only feasible to use fine-tuning in conjunction with randomised smoothing to create robust models using little compute, but that this method can increase the robustness of models, paving the road for the practical implementation of robust models into common usage.

Chapter 6

# Appendix

## 6.1 Training and testing

### 6.1.1 Training

Base training was done using 90 epochs, a learning rate of 0.1 decreasing by a factor of 10 every 30 epochs. Momentum was 0.9, and weight-decay was 0.0001.

Fine-tuning was done by running the training code for a further 30 epochs on the new dataset, with a learning rate of 0.001 and other settings identical.

Where AugMix or other augmentations were applied to data, they were applied anew each epoch to allow for the greatest variety of training images.

### 6.1.2 Certification

Certification was done using the code from the Cohen model (see section 2.1.4), with 100,000 samples used for Monte Carlo estimation, 100 Monte Carlo samples used for selection and an allowed possibility of failure of 0.001.

### 6.1.3 Prediction

Prediction was used for accuracy calculations, with the algorithm coming from the Cohen model. 100,000 samples were used for each accuracy estimation and an allowed possibility of failure of 0.001 (otherwise the model would abstain).

## 6.2 Augmentation algorithms

The full list of augmentations used in AugMix is: `autocontrast`, `equalize`, posterize, `rotate`, solarize, `shear x`, `shear y`, `translate x`, and `translate`

y. The details for autocontrast, `equalize`, `posterize`, and `solarize` can be found in the Pillow ImageOps Module[1], and the details of rotate, `shear x`, `shear y`, `translate x`, and `translate y` in the AugMix repo.[2]

For accessibility, we include the pseudocode for the most referenced AugMix augmentation, `autocontrast`:

**Algorithm 4:** Autocontrast

```
def MAPPING(image : matrix, map : list):
    new_image ← image
    for i, j in SIZE(image):
        new_image[i,j] ← map[image[i,j]]
    return new_image
def AUTOCONTRAST(image : array):
    map ← []
    hi ← HIGHEST_INTENSITY(image)
    lo ← LOWEST_INTENSITY(image)
    scale ← 255/(hi−lo)  offset ← −lo · scale
    for i ← 0 in [0, 255]:
        i ← INT(scale · i)
        if i < 0:
            i ← 0
        elif i > 255:
            i ← 255
        map ← APPEND(map, i)
    return MAPPING(image, map)
```

Note that the pseudocode does not include the code for cutoff that is in the ImageOps Module. We leave it out as throughout the experiments performed in this thesis no cutoff was used.

[1] https://pillow.readthedocs.io/en/stable/reference/ImageOps.html
[2] https://github.com/google-research/augmix/tree/master

# Bibliography


A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 2017. doi: 10.1109/SP.2017.49.

N. Carlini, F. Tramer, K. D. Dvijotham, L. Rice, M. Sun, and J. Z. Kolter. (certified!!) adversarial robustness for free! In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JLg5aHHv7j.

Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. C. Duchi. *Unlabeled Data Improves Adversarial Robustness*. 2019.

J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR, 2019. URL https://proceedings.mlr.press/v97/cohen19c.html.

L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey. Cinic-10 is not imagenet or cifar-10. Technical report, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Z. Deng, L. Zhang, K. Vodrahalli, K. Kawaguchi, and J. Y. Zou. Adversarial training helps transfer learning via better representations. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/d3aeec875c479e55d1cdeea161842ec6-Paper.pdf.

Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00957.

I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

Z. Hao, C. Ying, Y. Dong, H. Su, J. Song, and J. Zhu. GSmooth: Certified robustness against semantic transformations via generalized randomized smoothing. In *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2022. URL https://proceedings.mlr.press/v162/hao22c.html.

J. Hayes. Extensions and limitations of randomized smoothing for robustness guarantees. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3413–3421, 2020. doi: 10.1109/CVPRW50498.2020.00401.

M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/e077e1a544eec4f0307cf5c3c721d944-Paper.pdf.

D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1gmrxHFvB.

A. Javanmard, M. Soltanolkotabi, and H. Hassani. Precise tradeoffs in adversarial training for linear regression. In *Proceedings of Thirty Third Conference on Learning Theory*, Proceedings of Machine Learning Research, 2020. URL https://proceedings.mlr.press/v125/javanmard20a.html.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

A. Kumar, A. Raghunathan, R. M. Jones, T. Ma, and P. Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In

*International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

A. Kurakin, D. Boneh, F. Tramèr, I. Goodfellow, N. Papernot, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/pdf?id=rkZvSe-RZ.

A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, 2019. doi: 10.1109/SP.2019.00044.

H. Lee and A. Banerjee. Intelligent scheduling and motion control for household vacuum cleaning robot system using simulation based optimization. In *2015 Winter Simulation Conference (WSC)*, 2015. doi: 10.1109/WSC.2015.7408242.

K. Lång, V. Josefsson, A.-M. Larsson, S. Larsson, C. Högberg, H. Sartor, S. Hofvind, I. Andersson, and A. Rosso. Artificial intelligence-supported screen reading versus standard double reading in the mammography screening with artificial intelligence trial (masai): a clinical safety analysis of a randomised, controlled, non-inferiority, single-blinded, screening accuracy study. *The Lancet Oncology*, 2023.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

M. Mirman, T. Gehr, and M. Vechev. Differentiable abstract interpretation for provably robust neural networks. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3578–3586. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/mirman18b.html.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010. doi: 10.1109/TKDE.2009.191.

M. S. Pydi and V. Jog. The many faces of adversarial risk: An expanded study. *IEEE Transactions on Information Theory*, 2023. doi: 10.1109/TIT.2023.3303221.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings*

*of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. *CoRR*, abs/1801.09344, 2018. URL http://arxiv.org/abs/1801.09344.

A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014. URL http://arxiv.org/abs/1403.6382.

H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry. Do adversarially robust imagenet models transfer better? In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020a. ISBN 9781713829546.

H. Salman, M. Sun, G. Yang, A. Kapoor, and J. Z. Kolter. Black-box smoothing: A provable defense for pretrained classifiers. *CoRR*, abs/2003.01908, 2020b. URL https://arxiv.org/abs/2003.01908.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *The Second International Conference on Learning Representations*, 2014.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

J. Uesato, B. O'Donoghue, P. Kohli, and A. van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/uesato18a.html.

P. Vaishnavi, K. Eykholt, and A. Rahmati. Accelerating certified robustness training via knowledge transfer. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/22bf0634985f4e6dbb1fb40e247d1478-Paper-Conference.pdf.

L. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon. Towards fast computation of certified robustness for ReLU net-

works. In *Proceedings of the 35th International Conference on Machine Learning*, 2018. URL https://proceedings.mlr.press/v80/weng18a.html.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014. URL https://api.semanticscholar.org/CorpusID:362467.

H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/zhang19p.html.

F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 2021. doi: 10.1109/JPROC.2020.3004555.