Can we have it all? On the Trade-off between Spatial and Adversarial Robustness of Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

(Non-)robustness of neural networks to small, adversarial pixel-wise perturbations, and as more recently shown, to even random spatial transformations (e.g., translations, rotations) entreats both theoretical and empirical understanding. Spatial robustness to random translations and rotations is commonly attained via equivariant models (e.g., StdCNNs, GCNNs) and training augmentation, whereas adversarial robustness is typically achieved by adversarial training. In this paper, we prove a quantitative trade-off between spatial and adversarial robustness in a simple statistical setting. We complement this empirically by showing that: (a) as the spatial robustness of equivariant models improves by training augmentation with progressively larger transformations, their adversarial robustness worsens progressively, and (b) as the state-of-the-art robust models are adversarially trained with progressively larger pixel-wise perturbations, their spatial robustness drops progressively. Towards achieving pareto-optimality in this trade-off, we propose a method based on curriculum learning that trains gradually on more difficult perturbations (both spatial and adversarial) to improve spatial and adversarial robustness simultaneously.

1 Introduction

Neural network (NN) models have achieved state-of-the-art performance on several image tasks over the last few years. However, they have been known to be vulnerable to adversarial attacks that make small, imperceptible pixel-wise perturbations to images [39, 5]. Most pixel-wise adversarial perturbations studied in previous work are carefully constructed, model-dependent and, in most cases also input-dependent, perturbations of small ℓ_∞ norm (and sometimes ℓ_1 or ℓ_2 norm). To achieve adversarial robustness, a popular, principled, and effective defense against ℓ_∞ adversarial attacks known at present is adversarial training [17, 29, 37, 49].

From another perspective, recent work has also shown natural, physical-world attacks on neural network models using 2D and 3D spatial transformations such as translations, rotations and pose [47, 14, 42, 1]. Even random spatial transformations such as random translations and random rotations degrade the accuracy of neural network models considerably [12]. Recent benchmarks also measure the robustness of NN models to the average-case, model-agnostic perturbations (e.g., noise, blur, brightness, contrast) or natural distribution shifts, and not only to the worst-case adversarial perturbations [21, 40]. Spatial robustness or robustness to spatial transformations (e.g., translations, rotations) has been long studied as a problem of model invariance, and is generally addressed using equivariant models or training augmentation. While standard Convolutional Neural Networks (StdCNNs) are translation-equivariant, recent efforts have resulted in equivariant NN models for other transformations such as rotation, flip [16, 8, 10, 45, 31, 26, 27, 13], and scaling [30, 46, 38]. Such group-equivariant NN models (GCNNs) are designed to be invariant to a specific group of

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

transformations, so they need training augmentation to attain spatial robustness to random or average-case transformations. However, aided by clever weight-sharing, these recent works have achieved good spatial robustness performance with lesser training augmentation than their non-equivariant counterparts.

In this paper, we address an inevitable trade-off between the average-case spatial robustness and the worst-case ℓ_∞ adversarial robustness in neural network models, and show results theoretically as well as empirically. On the one hand, as we attain better spatial robustness via equivariant models and larger training augmentation, the adversarial robustness worsens. On the other hand, as we attain better adversarial robustness via adversarial training against larger pixel-wise perturbations, the spatial robustness worsens. We then provide a curriculum learning-based approach to mitigate this issue and achieve a certain degree of pareto-optimality in this trade-off.

Our work extends recent progress on two important questions, namely, the robustness to multiple attacks and the robustness vs accuracy trade-off. The robustness to translations (up to ± 3 pixels) and rotations (up to $\pm 30^\circ$) has been shown to have a fundamentally different loss landscape than the robustness to adversarial ℓ_p perturbations [12]. A trade-off between robustness to adversarial ℓ_∞ perturbations and adversarial translations and rotations up to ± 3 pixels and $\pm 40^\circ$, respectively, was also recently noted [42]. Adversarial attacks on ℓ_p adversarially robust models by exploiting invariance at intermediate layers have also been of interest in recent work [23, 41]. A key difference of our work with these efforts is that: (i) we study spatial robustness to *random* translations and rotations (which is more closely related to practice when training NN models and has not been studied in these earlier efforts), instead of adversarial translations or rotations; (ii) we study spatial robustness w.r.t transformations over their entire range, and do not restrict ourselves to a $\pm 30 - 40^\circ$ range, for instance; and (iii) we consider the use of group-equivariant models with training augmentation (which are designed for spatial robustness) in our studies, beyond just StdCNN models. Our results show a *progressive decline* in spatial robustness as adversarial robustness improves, and vice versa – a trend that was not explicitly noted in previous results. Our result can also be interpreted as a progressive trade-off between adversarial robustness and generalization accuracy itself [43, 49, 48], where generalization performance measures accuracy on a test distribution that includes natural data augmentation such as random translations, rotations, etc.

Given the nature of this trade-off, we then ask the question of how one can obtain a model close to the pareto-optimal frontier for adversarial and random spatial robustness. Are there training strategies that can help achieve a balance between both? We observe that the obvious solutions of training in two stages – first, adversarial training and then training augmentation by spatial transformations, or vice versa – does not help address this issue. This results in a version of catastrophic forgetting, where as the model learns robustness on one of these fronts, it loses robustness on the other. We instead show that a curriculum learning-based strategy, where the model is trained by applying an adversarial perturbation to a randomly transformed input with gradually increased difficulty levels, gives solutions consistently closer to the pareto-optimal frontier across well-known datasets and models. This strategy reflects the progressive nature noted of the trade-off itself, and highlights the importance of the observation of this trend.

Our key contributions can be summarized as follows:

- We study a trade-off between adversarial robustness and spatial robustness (with a focus on *random* spatial transformations) both theoretically and empirically, on a setting closer to natural neural network training, which to the best of our knowledge, has not been considered before. We explain the trade-off between spatial and adversarial robustness with a well-motivated theoretical construction.

- We conduct a comprehensive suite of experiments across popularly used models and datasets, and find that models trained for spatial robustness (StdCNNs and GCNNs) progressively improve spatial robustness when their training is augmented with larger random spatial transformations but while doing so, their ℓ_∞ adversarial robustness drops progressively. Similarly, models adversarially trained using progressively larger ℓ_∞-norm attacks improve their adversarial robustness but while doing so, their spatial robustness drops progressively.

- We provide a simple yet effective approach based on curriculum learning that gives better-balanced, pareto-optimal spatial and adversarial robustness compared to several natural baselines derived from combining adversarial training and training augmentation by random spatial transformations.

2 Problem Setting and Formulation

We begin by describing our problem formally. Let (X, Y) denote a random input-label pair from a given data distribution over the input-label pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Let f be a neural network classifier. The accuracy of f is given by $\Pr(f(X) = Y)$, the fraction of inputs where the prediction matches the true label. For $\epsilon > 0$, an ℓ_∞ adversarial attack $\mathcal{A}$ maps each input x to $x + \mathcal{A}(x)$ such that $\|\mathcal{A}(x)\|_\infty \leq \epsilon$, for all $x \in \mathcal{X}$. The *fooling rate* of $\mathcal{A}$ is given by $\Pr(f(X + \mathcal{A}(X)) \neq f(X))$, the fraction of inputs where the prediction changes after adversarial perturbation. For a given spatial transformation T, the classifier f is said to be T-invariant if the predicted label remains unchanged after the transformation T for all inputs; in other words, $f(Tx) = f(x)$, for all $x \in \mathcal{X}$. The spatial transformations such as translation, rotation and flip preserve the ℓ_∞ norm, so $T\mathcal{A}(x)$ is an adversarial perturbation of small ℓ_∞ norm for the input Tx for a T-invariant classifier. Therefore, for a given perturbation bound $\epsilon > 0$, the maximum fooling rate for any T-invariant classifier f on the transformed data $\{Tx \ : \ x \in \mathcal{X}\}$ must be equal to the maximum fooling rate for f on the original data $\mathcal{X}$. It is a bit more subtle when f is not truly invariant, that is, $f(Tx) = f(x)$ for most inputs x but not all x. We define the rate of invariance of a classifier f to a transformation T as $\Pr(f(TX) = f(X))$ or the fraction of test images whose predicted labels remain unchanged under the transformation T. For a class of transformations, e.g., random rotation r from the range $[-\theta^\circ, +\theta^\circ]$, we define the *rate of invariance* as the average rate of invariance over transformations T in this class, i.e., $\Pr(f(r(X)) = f(X))$, where the probability is over the random input X as well as the random transformation r. The rate of invariance is 100% if the model f is truly invariant. When f is not truly invariant, the interplay between the invariance under transformations and robustness under adversarial perturbations of small ℓ_∞-norm is subtle. *This interplay is exactly what we investigate.*

In this paper, we study neural network models and the simultaneous interplay between their spatial robustness, viz, rate of invariance for spatial transformations (such as random rotations between $[-\theta^\circ, +\theta^\circ]$ for θ varying in the range $[0, 180]$), and their adversarial robustness to pixel-wise perturbations of ℓ_∞ norm at most ϵ. Measuring the robustness of a model to adversarial perturbations of ℓ_p norm at most ϵ is NP-hard [24, 37]. Previous work on the comparison of different adversarial attacks has shown the Projected Gradient Descent (PGD) attack to be among the strongest [2]. Hence, we use PGD-based adversarial training as the methodology for adversarial robustness in a model, as commonly done. We show that our conclusions also hold for other forms of adversarial training such as TRADES [49], which gives a trade-off between adversarial robustness and natural accuracy (on the original unperturbed data).

Theoretically speaking, we prove a quantitative trade-off between spatial and adversarial robustness in a simple, natural statistical setting proposed in previous work [43]. Our theoretical results (described in Sec 3) show that there is a trade-off between spatial and adversarial robustness, and that it is not always possible to have both in a given model. Empirically, we study the following: (a) change in ℓ_∞ adversarial robustness as we improve only the rate of spatial robustness using training augmentation with progressively larger transformations; (b) change in spatial invariance as we improve only adversarial robustness using PGD adversarial training with progressively larger ℓ_∞-norm of pixel-wise perturbations. We study StdCNNs, group-equivariant GCNNs, as well as popular architectures used to assess adversarial robustness [29, 49]. Our empirical studies are conducted on MNIST, CIFAR10, CIFAR100 as well as Tiny ImageNet datasets, thus showing the general nature of these results (described in Sec 4). Importantly, as stated earlier, we consider random spatial transformations that are more commonplace than adversarial rotations in previous work [42]. We look at the entire possible range for spatial transformations. Similarly, we normalize the underlying dataset, and compute the accuracy of a given model on test inputs adversarially perturbed using PGD attack of ℓ_∞ norm at most ϵ, for ϵ varying over $[0, 1]$. In contrast with the certifiable lower bounds on robustness [3, 15, 35, 28, 32], we study upper bounds on spatial and adversarial robustness, respectively.

3 Spatial-Adversarial Robustness Trade-off

In this section, we prove the trade-off between spatial and adversarial robustness theoretically, and support this result with experiments in Sec 4. We use $\mathcal{A}(x)$ to denote an adversarial ℓ_∞ perturbation and $r(x)$ to denote a random spatial transformation. Equivariant model constructions often consider a group of transformations and construct a neural network model invariant to this group. For simplicity, we consider a cyclic group that can model a rotation group (e.g., integer multiples of 30°), or horizontal/vertical translations (e.g., horizontal translations by multiples of, say, ± 4 pixels).

We take a construction proposed in previous work for a robustness vs accuracy trade-off [43], and infuse it with a simple idea from the theory of error correcting codes [34]. Consider a binary cyclic code of length d, where each codeword is a binary string in $\{-1, 1\}^d$ and all the codewords can be obtained by applying successive cyclic shift of coordinates to a generator codeword; see Chapter 8 in [34]. The cyclic code is said to have relative distance δ, if any two codewords differ in at least δd coordinates. Let $c = (c_1, c_2, \ldots, c_d)$ be the generator codeword. Consider a random input-label pair (X, Y), with $X = (X_0, X_1, \ldots, X_d)$ taking values in $\mathbb{R}^{d+1}$ and Y taking values in $\{-1, 1\}$, generated as follows. The class label Y takes value ± 1 with probability $1/2$ each. $X_0 \mid Y = y$ takes value y with probability p and $-y$ with probability $1 - p$, for some $p \geq 1/2$. The remaining coordinates are independent and normally distributed with $X_t \mid Y = y$ as $N(2c_t y/\sqrt{d}, 1)$, for $1 \leq t \leq d$. Let m be an integer divisor of d, and let $r_j(x)$ denote the cyclic shift of mj coordinates applied to $(x_1, x_2, \ldots, x_d)$ while keeping x_0 unchanged, i.e., $r_j(x) = (x_0, x_{mj+1}, x_{mj+2}, \ldots, x_d, x_1, \ldots, x_{mj})$. For example, a 90° rotation can be considered as a cyclic permutation of pixels with $m = d/4$ and the center pixel x_0 being fixed. Now let $r(x)$ denote a random permutation that takes value $r_j(x)$, uniformly at random over $j \in \{1, 2, \ldots, d/m\}$. For example, using $m = d/4$ as above, $r(x)$ can model a random multiple of 90° rotation.

First, we show that spatial robustness on the above distribution is non-trivial even if we have high accuracy on the original data without any transformations. So our subsequent trade-off between spatial and adversarial robustness cannot be derived from previously known trade-offs between accuracy and adversarial robustness [43, 49].

Proposition 1. *There exists $p \geq 1/2$ and a cyclic code with relative distance $\delta \geq 3/8$ such that given the input distribution defined as above, the classifier of maximum accuracy on input (X, Y) has accuracy at least 97%. Similarly, the classifier of maximum accuracy on the transformed input $(r_j(X), Y)$ also has accuracy at least 97%. However, when the classifier of maximum accuracy on (X, Y) is applied to $(r_j(X), Y)$, for any j, it has accuracy at most 85%.*

Next we show that if we have a classifier of high adversarial robustness on the above data distribution, then it must have low spatial robustness. Moreover, the spatial robustness degrades for larger range of spatial transformations. In this context, m signifies the range of spatial transformations, e.g., $m = d/4$ corresponds to 90° rotation whereas $m = d/2$ corresponds to 180° rotation.

Theorem 2. *Given the input distribution defined as above and a classifier $f : \mathbb{R}^{d+1} \to \{-1, 1\}$, suppose $\Pr\left(f(X + \mathcal{A}(X)) = Y\right) \geq 1 - \eta$, for all adversarial attacks $\mathcal{A}$ of ℓ_∞ norm bounded by $4/\sqrt{d}$. Then*

$$\left(1 - \frac{m}{d}\right)\,(1 - \eta) \leq \Pr\left(f(r(X)) = Y\right) \leq 1 - \frac{m}{d}\,\left(1 - \frac{\eta\, p}{1 - p}\right).$$

Note that as $p \to 1/2$ and $\eta \to 0$, both the bounds approach $1 - m/d$, which decreases as m increases. For $m = d/2$, this accuracy is as bad as that of a random classifier.

The above results show that adversarial and spatial robustness can not just be high simultaneously; rather, increasing one progressively declines the other. We restate and prove Proposition 1 and Theorem 2 in the Supplementary. We do not explicitly model the effect of adversarial training with larger ℓ_∞ perturbations but, as a reasonable proxy, consider the increase in the adversarial accuracy for a fixed perturbation bound. The other direction that high spatial robustness implies low adversarial robustness is easier and follows directly from the previous accuracy vs robustness trade-off result [43]. Their distribution is invariant to any permutation of the coordinates $x_1, x_2, \ldots, x_d$, so the accuracy and the spatial robustness are equal for their distribution.

4 Empirical Analysis

Experimental Setup. In order to study the effect of spatial transformations (e.g. rotations, translations) and adversarial perturbations (e.g. Projected Gradient Descent or PGD attacks), we use well-known popularly used NN architectures as well as state-of-the-art networks known to have strong invariance [8], as well as networks which are known to have strong robustness [29, 49]. We study the spatial vs adversarial robustness trade-off using well-known architectures (described below) on MNIST, CIFAR10, CIFAR100 and Tiny ImageNet datasets. Our code is made available for reproducibility.

Spatially Robust Model Architectures: StdCNNs are known to be translation-equivariant by design, and GCNNs [8] are rotation-equivariant by design through clever weight sharing [25]. Equivariant

models, especially GCNNs, when trained with random rotation augmentations have been observed to come very close to being truly rotation-invariant [8, 9, 7] (or spatially robust in our context). We hence use both StdCNNs and equivalent GCNNs trained with suitable data augmentations for our studies with spatially robust architectures. In particular, for each StdCNN we use, the corresponding GCNN architecture is obtained by replacing the layer operations with equivalent GCNN operations as in [8][1]. For the StdCNN, we use a Conv-Conv-Pool-Conv-Conv-Pool-FC-FC architecture for MNIST (more details in the supplementary); VGG16 [36] and ResNet18 [19] for CIFAR10 and CIFAR100; and ResNet18 for the Tiny ImageNet dataset.

Adversarially Robust Model Architectures: For adversarial training, we use a LeNet-based architecture for MNIST[2] and a ResNet-based architecture for CIFAR10[3]. Both these models are exactly as given in [29]. For CIFAR100, we use the popularly used WideResNet-34[4] architecture also used in [49]. We use ResNet18 [19] for the Tiny ImageNet dataset.

Training Data Augmentation: Spatial: (a) **Aug - R** : Data is augmented with random rotations in the range $\pm\theta^\circ$ given θ, along with random crops and random horizontal flips (for MNIST alone, we do not apply crop and horizontal flips); (b) **Aug - T**: Data is augmented with random translations within $[-i, +i]$ range of pixels in the image, given i (eg. for CIFAR10 with $i = 0.1$ is $32 * 0.1 \approx \pm 3px$) in both horizontal and vertical directions; (c) **Aug - RT**: Data is augmented with random rotations in $\pm i * 180^\circ$ and random translations within $[-i, +i]$ range of pixels in the image (eg. for CIFAR10 with $i = 0.1$ is $0.1 * 180^\circ = \pm 18^\circ$ rotation and $32 * 0.1 \approx \pm 3px$ translation), here no cropping and no horizontal flip is used. We use nearest neighbour interpolation and black filling to obtain the transformed image. *Adversarial* : **Adv - PGD**: Adversarial training using PGD-perturbed adversarial samples using an ϵ-budget of given ϵ. Our experiments with PGD use a random start, 40 iterations, and step size 0.01 on MNIST, and a random start, 10 iterations, and step size $2/255$ on CIFAR10, CIFAR100 and Tiny ImageNet. Our results, presented in the next section, are best understood by noting the augmentation method mentioned in the figure caption. For example, in Fig 1(a), the augmentation scheme used is **Aug-R**. The red line (annotated as 60 in the legend) corresponds to the model trained with random rotation augmentations in the range $\pm 60^\circ$.

Hardware Configuration: We used a server with 4 Nvidia GeForce GTX 1080i GPU to run all the experiments in the paper.

Evaluation Metrics: We quantify performance using a spatial invariance profile and an adversarial robustness profile.

Spatial invariance: We quantify rotation invariance by measuring the rate of invariance or the fraction of test images whose predicted label remains the same after rotation by a random angle between $[-\theta^\circ, \theta^\circ]$. As θ varies from 0 to 180, we plot the rate of invariance. We call this the *rotation invariance profile* of a given model. Similarly, by training a model with other augmentation schemes given above, we obtain a *translation invariance profile* and a *rotation-translation invariance profile* for a given model.

Adversarial robustness: Similarly, we quantify the ℓ_∞ adversarial robustness of a given model to a fixed adversarial attack (e.g., PGD) and a fixed ℓ_∞ norm $\epsilon \in [0, 1]$ by (1 - fooling rate), i.e., the fraction of test inputs for which their predicted label does not change after adversarial perturbation. We plot this for ϵ varying from 0 to 1. We call this the *robustness profile* of a given model. (One can also choose accuracy instead of 1 - fooling rate and we observe similar trends, as shown in the supplementary. We use 1 - fooling rate for clearer visual presentation of our main results.)

Results. We now present our results on spatially robust model architectures followed by adversarially robust ones.

Results on Spatially Robust Model Architectures: Fig 1 present our results on the trade-off on adversarial robustness for VGG16, trained with spatial augmentations, on CIFAR10. (More results including ResNet18 on CIFAR10, VGG16/ResNet18 on CIFAR100 and the StdCNN/GCNN-based model on MNIST are deferred to the supplementary owing to space constraints.)

For any fixed $\theta \in [0, 180]$, we take an equivariant model, namely, StdCNN or GCNN, and augment its training data by random rotations from $[-\theta^\circ, +\theta^\circ]$. For example, Fig 1(b) show how the robustness

[1] https://github.com/adambielski/pytorch-gconv-experiments

[2] https://github.com/MadryLab/mnist_challenge/

[3] https://github.com/MadryLab/cifar10_challenge

[4] https://github.com/yaodongyu/TRADES/models/

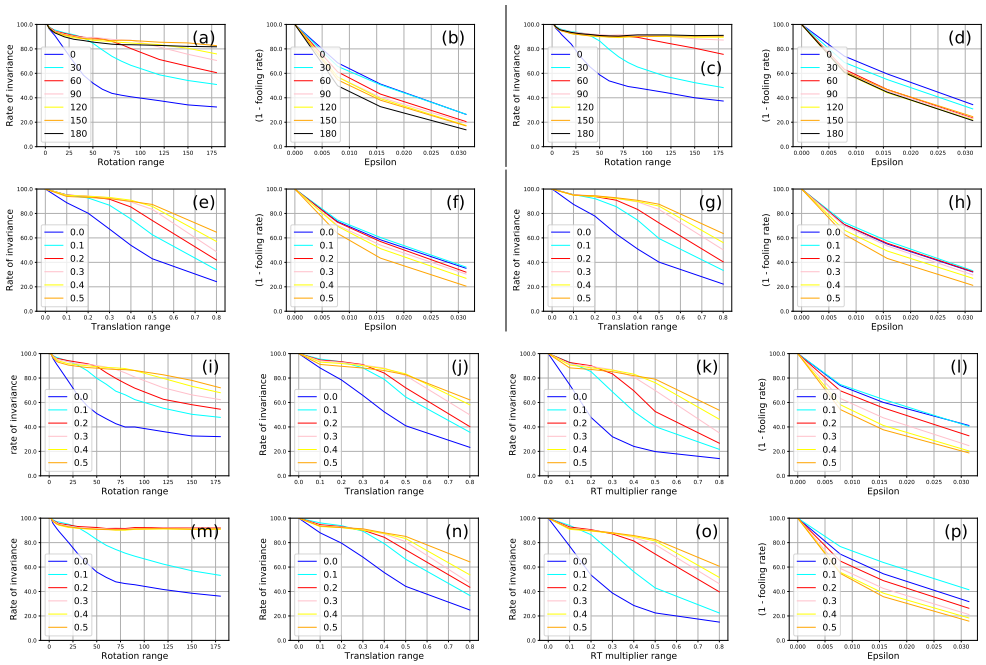

Figure 1: On CIFAR10, For VGG16 model (a-b) **Aug - R** StdCNN, (c-d) **Aug - R** GCNN, (e-f) **Aug - T** StdCNN, (g-h) **Aug - T** GCNN, (i-l) **Aug - RT** StdCNN, (m-p) **Aug - RT**, invariance profiles of StdCNN/GCNN models and corresponding robustness profiles.

profile of StdCNN changes on CIFAR10, as we increase the degree θ used in training augmentation of the model. We use PGD attack provided in versions of AdverTorch [11] to obtain the robustness profiles. Similarly, Figs 1(a) show the rotation invariance profile of the same models on CIFAR10. The black line in Fig 1(b) shows that the adversarial robustness of a StdCNN which is trained to handle rotations up to ± 180 degrees on CIFAR10, drops by more than 50%, even when the ϵ budget for PGD attack is only $2/255$. Similarly, the black line in Fig 1(a) shows this model's rotation invariance profile - this model is invariant to larger rotations on test data. This can be contrasted with the model depicted by the red line - this StdCNN is trained to handle rotations up to 60 degrees. The rotation invariance profile of this model is below that of the model depicted by the black line and hence is lesser invariant to large rotations. However, this model can handle adversarial ℓ_∞-perturbations up to $2/255$ on unrotated data, with an accuracy more than 10%, as seen from the red line in Fig 1(b). The remaining plots in Figs 1 show similar trends for translation, as well as a combination of rotation and translation (e.g. plots (e-p) in Figs 1 for CIFAR10) for both StdCNN and GCNN models (e.g. Figs 1c,d).

The above results show that *the spatial robustness of these models improves by training augmentation but at the cost of their adversarial robustness, indicating a trade-off between spatial and adversarial robustness.* This trade-off exists in both StdCNN and GCNN models.

Results on Adversarially Robust Model Architectures: Fig 2 presents our results on the trade-off on spatial robustness for models trained adversarially using PGD [29] on MNIST, CIFAR10 and CIFAR100 datasets. Each row corresponds to experiments for a single dataset with a plot of robustness profile (for which they are trained) followed by the spatial invariance profiles. Each colored line in the plots corresponds to a model adversarially trained with a different value of an ϵ-budget.

On MNIST, adversarial training with PGD with larger ϵ results in a drop in the invariance profile of the LeNet-based model; in Fig 2(b-d), the red line (PGD with $\epsilon = 0.3$) is below the light blue line (PGD with $\epsilon = 0.1$). Similar observations hold for the ResNet model on CIFAR10 (see Fig 2 (f-h)), as well as for WideResNet-34 on CIFAR100 (see Fig 2(j-l)). To complete this picture, the robustness profile curves confirm that as these models are trained with PGD using larger ϵ budget, their adversarial robustness increases. The robustness profile curves of the LeNet model trained with a larger PGD budget dominates the robustness profile curve of the same model trained with a smaller

PGD budget; the red line in Fig 2(a) dominates the light blue line. This is true of the ResNet model too, as in Figs 2(e) and 2(i).

In other words, *adversarial training with progressively larger ϵ leads to a drop in the rate of spatial invariance on test data.* Fig 3 shows an additional result with StdCNN (VGG-16) with PGD-based adversarial training on CIFAR10. These results too support the same findings.

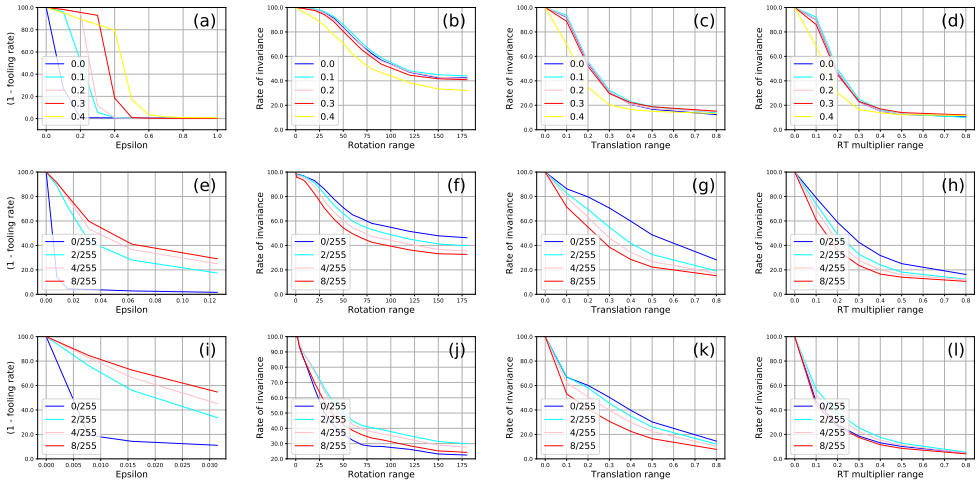

Figure 2: **Adv - PGD**, (1) Robustness, (2) Rotation, (3) Translation, (4) Rotation-Translation invariance profiles for PGD adversarially trained models. (a-d) LeNet based model from [29] on MNIST (e-h) ResNet based model from [29] on CIFAR10, (i-l) WideResNet34 model on CIFAR100. Different colored lines represent models adversarially trained with different ℓ_∞ budgets $\epsilon \in [0, 1]$.

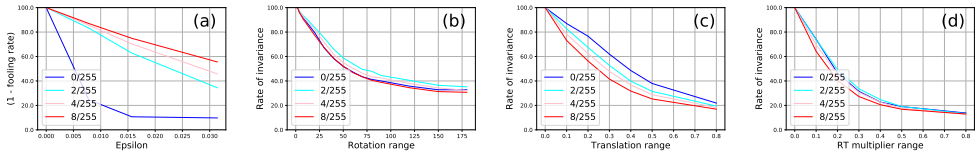

Figure 3: On CIFAR10, **Adv - PGD**, For StdCNN/VGG16 model (a) Robustness profile, (b) Rotation invariance profile, (c) Translation invariance profile (d) Rotation-Translation invariance profile. Different colored lines represent models adversarially trained with different ℓ_∞ budgets $\epsilon \in [0, 1]$.

Results on Tiny ImageNet: Appendix I studies both adversarial robustness of spatially robust models, as well as spatial invariance of adversarially robust models on the Tiny ImageNet dataset. Our observations mentioned so far holds here too, corroborating our claim that this trade-off is not merely theoretical but noticed in real-world datasets too.

Going beyond PGD-based Adversarial Training: While we have used the popular PGD-based adversarial training for the study of robust models in all the results so far, we conduct experimental studies with another recent popular adversarial method, TRADES [49]. Fig 4 shows the results of these studies on CIFAR10, where we observe the same qualitative behavior of the models.

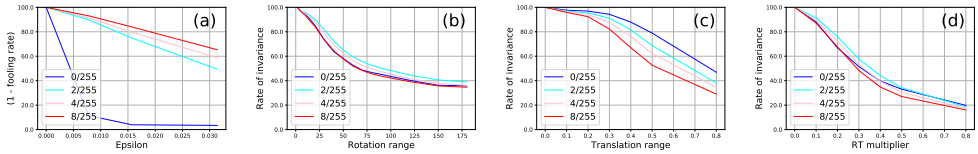

Figure 4: For adversarially trained WideResNet34 model with TRADES on CIFAR10: (a) Robustness, (b) Rotation invariance, (c) Translation invariance, (d) Rotation-Translation invariance profiles. Different colored lines represent models adversarially trained with different ℓ_∞ budgets $\epsilon \in [0, 1]$.

Additional Empirical Evidence (Average Perturbation Distance to Boundary) : While more results and implementation details are provided in the supplementary section, we briefly provide an interesting evidence to our claims based on the average distance to the decision boundary (details in

Appendix K). We notice that there is a distinct reduction in the distance to decision boundary as model achieves better spatial invariance(Figure 5). This reduction in distance can be linked to the drop in adversarial robustness. Such a study also has the potential to be used to study trade-offs in the presence of other common corruptions [21, 22, 20].

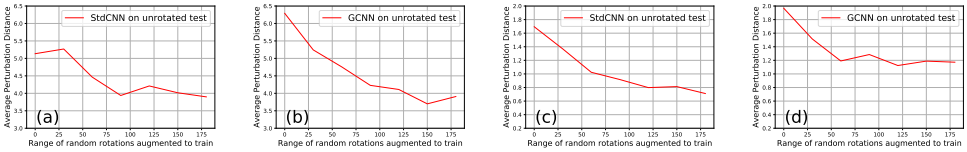

Figure 5: Average perturbation distance of StdCNN/GCNN on (a-b) MNIST(based on appendix Table 2), (c-d) CIFAR10(VGG16) wrt unrotated test data.

5 Curriculum Learning-based Approach for Pareto-Optimality

The progressive nature of the trade-off between spatial invariance and adversarial robustness elicits the need to study training methods that can simultaneously achieve robustness on both of these fronts. In this section, we propose a curriculum learning-based training strategy that consistently provides models close to the pareto-optimal frontier of the trade-off. An obvious strategy to obtain both kinds of robustness would be to sequentially combine adversarial training and augmentation by spatial transformations, i.e., take an adversarially trained network and re-train it with spatial (e.g., rotation) augmentation, or vice versa (Appendix J). We however observe that such a strategy presents a version of catastrophic forgetting where the robustness learned earlier is forgotten when the other robustness is enforced. A model thus trained has either good spatial robustness or good adversarial robustness but not both. We hence explore a curriculum learning strategy to this end.

Curriculum learning is inherently inspired by how humans learn; first learning simple tasks and then followed by gradually difficult tasks [4]. Previous work has demonstrated its effectiveness in training DNN models [44, 18]. A curriculum-based adversarial training has also been proposed [6], which however focuses only on adversarial robustness. For simultaneous spatial and adversarial robustness, we propose CuSP (**Cu**rriculum based **S**patial-Adversarial Robustness training for **P**areto-Optimality), where the difficulty of spatial augmentation and adversarial strength is gradually increased each time the learning rate is updated as part of a schedule. In particular, each time the learning rate is updated, we increase the spatial augmentation range $[-\theta^\circ, \theta^\circ]$ as well as the adversarial perturbation bound ϵ and adversarial train on a randomly perturbed input within that range. This strategy reflects an optimization procedure where the difficulty of robustness handled by the model is increased when the model treads more slowly (lower learning rate) along areas of the loss surface. The outline of the proposed CuSP approach (**Cu**rriculum based **S**patial-Adversarial Robustness training for **P**areto-Optimality) is given in Algorithm 1.

Algorithm 1: CuSP: **Cu**rriculum-based **S**patial-Adversarial Robustness Training for **P**areto-Optimality

Input : Training data (X, Y)
Parameters : No. of epochs: K, Learning rate: η, Adversarial training (AT) method: PGD or TRADES, Range of spatial transformations: $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_l$, Range of perturbation intensity: $\epsilon_1 \leq \epsilon_2 \leq \ldots \leq \epsilon_s$; Learning rate schedule: $\eta_1 > \eta_2 > \ldots > \eta_t$
Output : model parameters ϕ

Initialize model parameters ϕ; $i, j, k \leftarrow 1$
$\theta \leftarrow \theta_i, \epsilon \leftarrow \epsilon_j, \eta \leftarrow \eta_k$
for *epochs* $t = 1, \ldots, E$ **do**
  if $k \leftarrow k+1$; $\eta \leftarrow \eta_k$ `/* Change in learning rate schedule */`
  then $i \leftarrow i+1$; $j \leftarrow j+1$; Update $\theta \leftarrow \theta_i$ and $\epsilon \leftarrow \epsilon_j$;
  $\phi \leftarrow AdversarialTraining(r_\theta(X) + A_\epsilon(r_\theta(X), \phi, \theta, \eta, \epsilon)$
  `/* Minimize loss on input randomly transformed within` $[-\theta, \theta]$ `and an adversarial attack of norm at most` ϵ `*/`
end

To study the goodness of the proposed strategy, we look at three metrics: (a) spatial accuracy on large random transformations $\Pr\left(f(r_{180}(X)) = Y\right)$, where r_{180} denotes a random rotation in the range $[-180^\circ, 180^\circ]$, (b) adversarial accuracy at large perturbations $\Pr\left(f(X + A_{8/255}(X)) = Y\right)$, where

	Training Method	Adv (PGD) Accuracy(%)	Std Accuracy(%)	Spatial Accuracy(%)
1	PGD (Aug 0)	44.88 ± 0.22	80.08 ± 0.58	33.07 ± 0.81
2	PGD (Aug 180)	32.79±0.73	54.60±0.51	53.69±0.48
3	PGD (Aug 0) $\rightarrow$ Aug 180	00.00±0.00	85.46 ± 0.25	83.99 ± 0.34
4	Aug 180 $\rightarrow$ PGD (Aug 0)	45.80 ± 1.05	81.03 ± 0.14	33.19 ± 0.33
5	CuSP (PGD, 30-60-180, $\{\frac{2}{255}, \frac{4}{255}, \frac{8}{255}\}$)	38.63 ± 0.11	65.92 ± 0.35	53.31 ± 0.13
6	CuSP (PGD, 60-120-180, $\{\frac{2}{255}, \frac{4}{255}, \frac{8}{255}\}$)	37.18 ± 0.14	65.88 ± 0.11	59.09 ± 0.18
7	CuSP (PGD, 120-150-180, $\{\frac{2}{255}, \frac{4}{255}, \frac{8}{255}\}$)	**36.01±0.36**	**64.96±0.10**	**61.41±0.61**
8	CuSP (PGD, 180, $\{\frac{2}{255}, \frac{4}{255}, \frac{8}{255}\}$)	33.63±0.38	62.17±0.50	61.07±0.43
9	CuSP (PGD, 120-150-180,$\{\frac{8}{255}\}$)	35.57±0.16	58.08±0.26	55.54±0.19

Table 1: Comparison of performance of proposed CuSP for CIFAR10 on StdCNN/VGG16 with other baseline strategies. (Aug θ): denotes training data augmented with random rotations in the range $[-\theta, +\theta]$; $a \rightarrow b$: denotes a sequentially followed by b during training.

$A_{8/255}$ denotes an adversarial attack (PGD or TRADES) of ℓ_∞-norm bounded by $\epsilon = 8/255$, and finally, (c) natural accuracy on the unperturbed data $\Pr(f(X) = Y)$. The last one is included for completeness because we want better spatial and adversarial robustness simultaneously but not lose on natural accuracy in the bargain [43, 49].

Results: We studied CuSP across models and datasets in our experiments, and report the results of VGG16 on CIFAR10 in Table 1. The remainder of these results are presented in Appendix L. We follow the learning rate schedule 75-90-100 as used in TRADES [49] where the learning rate starting with 0.1 decays by a factor of 0.1 successively at the 75-th, 90-th and 100-th epochs. Our algorithm works with given sets of values for θ and ϵ and does not optimize for the choice of these values. We empirically studied various combinations of (θ, ϵ) values as θ varies over $\{0, 30, 60^\circ, \ldots, 180\}$ and ϵ varies over $\{0, 2/255, 4/255, 8/255\}$. While each of these options improves performance over the baselines, we found the list of $\theta = \{120^\circ, 150^\circ, 180^\circ\}$ to give the best results on this particular setting. Figure 6 presents a visualization of the performance of CuSP against other strategies to see its relevance to the pareto-optimal frontier.

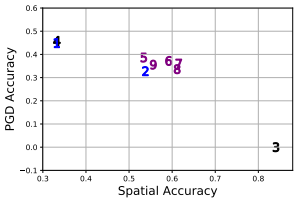

Figure 6: Visualization of performance of CuSP against other baseline strategies (each index corresponds to a row in Table 1).

6 Conclusions and Future Work

In this work, we present a comprehensive discussion on the trade-off between spatial and adversarial robustness in neural network models. While a few recent efforts have studied similar trade-offs, we study random spatial transformations in this work, which are more popularly used in practice than any adversarial spatial transformations. We study the trade-off both theoretically and empirically, and prove a quantitative trade-off between spatial and adversarial robustness in a simple statistical setting used in earlier work. Our experiments show that as equivariant models (StdCNNs and GCNNs) are trained with progressively larger spatial transformations, their spatial invariance improves but at the cost of their adversarial robustness. Similarly, adversarial training with perturbations of progressively increasing norms improves the robustness of well-known neural network models, but with a resulting drop in their spatial invariance. We also provide a curriculum learning-based training method to obtain a solution close to the pareto-optimal in comparison to other baseline adversarial training methods. Recently, [21, 22, 20] have shown that study of other common corruptions is important in the study of state-of-the-art NN models. Similar studies between spatial invariance and such common corruptions is an interesting direction of future work.

Broader Impact. Adversarial robustness is one of the most important problems in better understanding the functioning of neural networks. Another important problem is how networks can efficiently handle various spatial transformations like rotation, translation, etc. The trade-off studied and the algorithm proposed to find a better pareto-optimal in this paper tries to understand the interaction between the two which is a natural question. Hence, we believe this study is towards better understanding of the networks learning capability. There is no known detrimental social impact of our work.

References

[1] Michael A. Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *In Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[3] Mislav Balunovic, Maximilian Baader, Gagandeep Singh, Timon Gehr, and Martin Vechev. Certifying geometric robustness of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *In Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

[5] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. *Lecture Notes in Computer Science*, 2013.

[6] Qi-Zhi Cai, Chang Liu, and Dawn Song. Curriculum adversarial training. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3740–3747. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[7] Mario Cohen, Taco S. Geiger and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *arXiv preprint arXiv:1811.02017*, 2018.

[8] Taco S. Cohen and Max Welling. Group equivariant convolutional networks. *In Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[9] Taco S. Cohen and Max Welling. Steerable CNNs. *In International Conference on Learning Representations (ICLR)*, 2017.

[10] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. *In Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[11] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

[12] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. *In Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[13] Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar transformer networks. *In International Conference on Learning Representations (ICLR)*, 2018.

[14] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] Marc Fischer, Maximilian Baader, and Martin Vechev. Certified defense to image transformations via randomized smoothing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[16] Robert Gens and Pedro M Domingos. Deep symmetry networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *In International Conference on Learning Representations (ICLR)*, 2015.

[18] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *In Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

[22] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

[23] Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations (ICLR)*, 2019.

[24] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *CAV*, 2017.

[25] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. *In Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[26] Dmitry Laptev, Nikolay Savinov, Joachim M. Buhmann, and Marc Pollefeys. TI-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] Junying Li, Zichen Yang, Haifeng Liu, and Deng Cai. Deep rotation equivariant network. *arXiv preprint arXiv:1705.08623*, 2017.

[28] Linyi Li, Maurice Weber, Xiaojun Xu, Luka Rimanic, Tao Xie, Ce Zhang, and Bo Li. Provable robust learning based on transformation-specific smoothing, 2020.

[29] Aleksander Madry, Aleksandar A Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *In International Conference on Learning Representations (ICLR)*, 2018.

[30] Diego Marcos, Benjamin Kellenberger, Sylvain Lobry, and Devis Tuia. Scale equivariance in cnns with vector fields. *arXiv preprint arXiv:1807.11783*, 2018.

[31] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. *In International Conference on Computer Vision (ICCV)*, 2017.

[32] Jeet Mohapatra, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Towards verifying robustness of neural networks against a family of semantic perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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

[34] Ron Roth. *Introduction to Coding Theory*. Cambridge University Press, 2006.

[35] Anian Ruoss, Maximilian Baader, Mislav Balunović, and Martin Vechev. Efficient certification of spatial robustness, 2021.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Aman Sinha, Hongseok Namkoong, and John C. Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations (ICLR)*, 2018.

[38] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[40] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[41] Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *In Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[42] Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

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

[44] Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *In Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[45] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Harmonic networks: Deep translation and rotation equivariance. *arXiv preprint arXiv:1612.04642*, 2016.

[46] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. In *Advances in Neural Information Processing Systems*, 2019.

[47] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

[48] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R. Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[49] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *In Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In the supplementary.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]