Probing Predictions on OOD Images via Nearest Categories

Anonymous authors
Paper under double-blind review

Abstract

We study out-of-distribution (OOD) prediction behavior of neural networks when they classify images from unseen classes or corrupted images. To probe the OOD behavior, we introduce a new measure, *nearest category generalization* (NCG), where we compute the fraction of OOD inputs that are classified with the same label as their nearest neighbor in the training set. Our motivation stems from understanding the prediction patterns of adversarially robust networks, since previous work has identified unexpected consequences of training to be robust to norm-bounded perturbations. We find that robust networks have consistently higher NCG accuracy than natural training, even when the OOD data is much farther away than the robustness radius. This implies that the local regularization of robust training has a significant impact on the network's decision regions. We replicate our findings using many datasets, comparing new and existing training methods. Overall, adversarially robust networks resemble a nearest neighbor classifier when it comes to OOD data.

1 Introduction

A large amount of machine learning research focuses on improving classifier accuracy under the premise that test data is similar enough to training data to warrant good generalization. Recently, however, there has been much interest in the behavior of deep neural networks on out-of-distribution (OOD) data. In this context, OOD could mean that the inputs come from previously unseen categories (Salehi et al., 2021; Yang et al., 2021), or that the inputs have been adversarially perturbed (Madry et al., 2017) or corrupted (Hendrycks & Dietterich, 2019). Studying OOD generalization may improve the reliability of machine learning systems. Another goal comes from semi-supervised and self-supervised learning, where the model propagates labels to unlabeled data (Van Engelen & Hoos, 2020). Identifying clear patterns in OOD behavior can aid engineers in choosing among many methods.

Unfortunately, predictions on OOD data can be mysterious. For example, adversarially robust training methods often involve regularizing the network so that it predicts the same label on both a training example and on all points in a small ϵ-ball around the example (Madry et al., 2017; Zhang et al., 2019). Such methods only specify a *local* constraint on prediction behavior. At first glance, it may be tempting to guess that training to be robust in a small ϵ-ball should not affect predictions on other parts of the input space. However, it has become clear that robust training can cause substantial differences in *global* behavior. For example, it may lead to excessive invariances (Jacobsen et al., 2018; Tramèr et al., 2020), improve transfer learning (Salman et al., 2020), or change confidence on OOD data (Hein et al., 2019).

1.1 Nearest Category Generalization

To investigate the global behavior of robust networks, we explore patterns in how such networks predict on OOD data. Recent studies observe that robust training encourages the network to predict the same label not just in an ϵ-ball but also much further away in the pixel space (Carlini et al., 2019). However, the scope of these prior experiments is limited to certain OOD data, such as random noise or adversarial perturbations (Hein et al., 2019). While it is hard to analyze the entirety of the high-dimensional input space, we can probe predictions on OOD data in another way.

We introduce a new metric that we call Nearest Category Generalization (NCG). In short, we explore whether robust networks are more likely to classify OOD data with the class label of the nearest training input. Concretely, we can fix the same metric for both adversarially robust training and for the nearest neighbor prediction (e.g., ℓ_2 distance in pixel space). Then, we can calculate how often the predicted label on OOD data matches the 1-nearest-neighbor (1-NN) label. We refer to this measure as the *NCG accuracy*.

Following prior work (Salehi et al., 2021), we consider two canonical types of OOD data (i) **Unseen Classes:** during training, we hold out all images from one of the classes, but during testing, we predict labels for images from this unseen class, (ii) **Corrupted Data:** we train on all classes, but we test on images that have been corrupted. Importantly for us, both sources of OOD data have the property that the distance (e.g., ℓ_2) is quite large between the OOD data and the standard train/test images. In particular, the distance is much larger than the robustness radius used during robust training, and therefore, the training procedure does not explicitly dictate the predictions on such OOD data. Hence, NCG accuracy measures global resemblance to the 1-NN classifier for unseen or corrupted images.

Understanding NCG performance can shed new light on many research questions. First, if changes in the training method lead to significant changes in NCG accuracy, then this suggests the network's decision boundaries have shifted to extrapolate very differently on OOD data. Even for ReLU networks, understanding the decision regions far away from training data is an active and important area of research (Arora et al., 2016; Hanin & Rolnick, 2019; Williams et al., 2019). Second, NCG accuracy provides a new way to evaluate training methods on unlabeled data that cannot be labeled using other information. This is in contrast to transfer learning and few-shot learning that require auxiliary data Raghu et al. (2019); Yosinski et al. (2014); Wang et al. (2020). Overall, we emphasize that the NCG framework is not rooted in a standalone task, but instead, it highlights new prediction patterns of networks on OOD data.

1.2 Contributions

Our first finding is that NCG accuracy is consistently higher than chance levels for many training methods and datasets. This means that the behavior on OOD data is far from random. On both natural and corrupted OOD images, the network favors the 1-NN label. Surprisingly, this correlation with 1-NN happens in pixel space with ℓ_2 distance. The network converges to a classifier that depends heavily on the geometry of the input space, as opposed to making predictions based on semantic or higher-level structures.

Next, we show that robust networks indeed have much higher NCG accuracy than natural training methods. This holds for a variety of held out classes, corruption types, and robust training methods. OOD inputs that are classified with the NCG label (1-NN in training set) are considerably further than from their closest training examples. These training examples also have adversarial examples that are closer than the robustness radius r (see Figure 1). This implies that the decision regions of adversarially robust methods extend in certain directions, but not others, and that the local training constraints lead to globally different behavior. We can identify these types of prediction patterns precisely because we examine the network with OOD data.

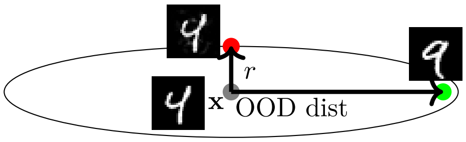


Figure 1: Robust networks tend to predict the same at a large distance in some directions, e.g., toward natural OOD examples (green), but are susceptible to adversarial examples that are closer in the worst-case directions (red).

Besides unseen classes, we also look at corrupted data, such as CIFAR10-C, CIFAR100-C, and ImgNet100-C Hendrycks & Dietterich (2019). The NCG accuracies for all networks (including natural and robust networks) are above the chance levels, and robust networks often have much higher NCG accuracy than natural training. We also uncover an interesting correlation between NCG accuracy and prediction accuracy for corrupted data. Corrupted examples that are correct in terms of NCG accuracy have a higher chance of being classified correctly in terms of the semantic label as well. In other words, if the network matches the 1-NN prediction, then it is more likely to predict the correct label for a corrupted image. This indicates that having higher NCG accuracy may be a desirable property, as it can encourage better predictions on corrupted data.

Our work uncovers an intriguing OOD generalization property of neural networks, and we find that robust networks have higher NCG accuracy than naturally trained counterparts. However, robust training does not inherently impose constraints on the OOD data that we consider (OOD data are far from perturbed examples). We posit that the NCG behavior is a consequence of the inductive bias produced by neural networks (especially for adversarially robust networks). Also, different forms of robustness (ℓ_2 and corruptions) may be interconnected with NCG at a deeper level.

1.3 Related Work

There has been much research on properties of robust networks beyond robustness. Santurkar et al. (2020) study the performance of adversarially trained models on subpopulation shift. Salman et al. (2020) consider transfer learning for robust models. Kireev et al. (2021) look at accuracy on corrupted data for robust training methods.

Our results on NCG strengthen and complement existing efforts in understanding the excessive invariances that are induced by adversarial training (Jacobsen et al., 2018; Ortiz-Jimenez et al., 2020; Tramèr et al., 2020). Another related area is extrapolation (Balestriero et al., 2021; Xu et al., 2020), where we provide a theoretical result (Theorem 1) that corroborates claims that higher diversity of the training distribution helps extrapolation to linear target functions (Hein et al., 2019; Xu et al., 2020).

Data augmentation is an effective way to improve generalization and robustness (Cubuk et al., 2020; Shorten & Khoshgoftaar, 2019). The success of this approach is consistent with our findings. Local regularization can lead to unexpected behavior, and hence, training on far-away images helps control the decision regions (e.g., for robustness and generalization, cf. Herrmann et al. (2021)).

Connecting NCG with OOD detection is a nice direction for future work (Manevitz & Yousef, 2001; Liang et al., 2018; Ren et al., 2019). OOD detectors already work well for some tasks, such as detecting data from another dataset (Sehwag et al., 2021; Tack et al., 2020). Holding out a single class from the same dataset provides a harder instance of OOD detection. Predictions on unseen data are also studied in the area of open set recognition (Dhamija et al., 2020). Another approach to OOD generalization involves the confidence/uncertainty on OOD data (Kristiadi et al., 2020; Meinke & Hein, 2019; Van Amersfoort et al., 2020). However, much of this work takes a Bayesian perspective and calibrates OOD predictions. We focus on a geometric framework, looking at distances and 1-NN labels in the input space. The perceptual organization of neural networks is another way to probe OOD behavior (Kim et al., 2021).

2 Preliminaries

OOD Data. We consider two standard types of OOD data. **Unseen Classes:** We hold out all examples from one class during training time (e.g., MNIST without 9s). During test time, we evaluate test accuracy on the remaining classes, and we evaluate NCG accuracy on the held out class (e.g., we predict 0–8 for the 9s). For each dataset, we hold out different classes, and we use the shorthand `dataset`-wo# to mean that this class # is unseen. For example, MNIST-wo9 is MNIST with unseen digit 9, CIFAR10-wo0 is CIFAR10 with unseen *airplane* and CIFAR100-wo0 is CIFAR100 with unseen *aquatic mammals.* We shorten this as M-9, C10-0, C100-0, etc. We use coarse labels for CIFAR100. For ImageNet, we subsample to 100 classes to form ImgNet100. **Corruptions:** We train on all classes and evaluate standard corruptions, which includes CIFAR10-C and CIFAR100-C (Hendrycks & Dietterich, 2019). Again, we measure NCG accuracy by classifying corrupted data and checking whether the label matches the 1-NN training label.

Adversarially Robust Training. Let $\mathcal{B}(\mathbf{x}, r)$ be a ball of radius $r > 0$ around $\mathbf{x}$ in a metric space. A classifier f is said to be *robust* at $\mathbf{x}$ with radius r if for all $\mathbf{x}' \in \mathcal{B}(\mathbf{x}, r)$, we have $f(\mathbf{x}') = f(\mathbf{x})$. Standard adversarially robust training methods such as TRADES (Zhang et al., 2019) work by minimizing a loss function that is the sum of the cross-entropy loss plus a regularization term; this regularization term encourages that the network is smooth in a ball of radius r around each training point $\mathbf{x}_i$, ensuring robustness in this ball.

Concretely, the TRADES loss is:

$$\ell(f_\theta(\mathbf{x}_i), y_i) + \beta \max_{\mathbf{x}'_i \in P_i} D_{\mathrm{KL}}(f_\theta(\mathbf{x}'_i), f_\theta(\mathbf{x}_i)), \tag{1}$$

where β is a tradeoff parameter, ℓ is the cross-entropy loss, and $P_i = \mathcal{B}(\mathbf{x}_i, r)$ is the ball of radius r around $\mathbf{x}_i$.

Distance Metrics for NCG. We use the ℓ_2 distance for two representations. **Pixel Space:** Robust methods aim to have invariant predictions within a small norm ball in pixel space. Hence, we evaluate the distance in pixel space (we believe the ℓ_∞ results would be similar). **Feature Space:** For another representation, we first train a different neural network (fully connected MLP) on the in-distribution data (we omit the unseen class). Then we compute the last layer embedding for all images, including those in the unseen class, giving us a learned, latent embedding. This provides vectors for both in-distribution and OOD images, and we use these vectors as our "feature space" version of the datasets. Note that we *do not* claim that these representations capture human-level or semantic similarity for the OOD data. Nonetheless, they both suffice to provide insight into the OOD prediction behavior of robust and normal networks.

NCG Accuracy. The NCG accuracy is the fraction of OOD inputs that are labeled as their nearest neighbor in the training set (i.e., we measure agreement with the 1-NN classifier). For corrupted data, we use the whole training set. For unseen classes, we use the training set minus the held out class. We measure the 1-NN prediction in ℓ_2 distance in either the pixel space or the above-defined feature space.

3 NCG for Unseen Classes

Set-up. For natural/robust training, on MNIST, we evaluate a CNN; on CIFAR10/100, we use Wider ResNet (WRN-40-10); on ImageNet, we use ResNet50. We consider natural training and mixup (Zhang et al., 2017) as baselines. For robust training, we consider two standard methods, Adversarial Training (Madry et al., 2017)(AT) and TRADES (Zhang et al., 2019). These methods are known to have high adversarial robustness to ℓ_2 perturbations, and hence, they serve as good baseline examples of robust networks. For TRADES, we use robustness radii $r \in \{2, 4, 8\}$ for the ℓ_2 ball in Equation (1). In the pixel space, we use $r = 2$ for AT. In the feature space, we set $r = 1$ for AT on CIFAR10/100, and $r = .5$ for AT on ImgNet100 (on CIFAR10/100, AT failed to converge with $r = 2$ and on ImgNet100 with $r \in \{1, 2\}$). We denote TRADES with $r = 2$ and AT with $r = 1$ as TRADES(2) and AT(1), respectively. Prior work observes that AT and TRADES give similar results with parameter tuning (Yang et al., 2020; Carmon et al., 2019), and hence we expect them to behave similarly. Appendix C has more details.

Datasets. We consider all 10 classes as the unseen class for MNIST and three unseen classes for each of CIFAR10, CIFAR100, and ImgNet100. CIFAR10, we consider removing the *airplane*, *deer*, and *truck* classes; for CIFAR100, we remove the *aquatic mammals*, *fruit and vegetables*, and *large man-made outdoor things* classes; for ImgNet100, we remove the *American robin*, *Gila monster*, and *eastern hog-nosed snake* classes. These are denoted as CIFAR10-wo{0, 4, 9}, CIFAR100-wo{0, 4, 9}, ImgNet100-wo{0, 1, 2}.

Results. Table 1 shows the NCG accuracy of natural and robust models averaged over unseen classes (in both pixel and feature space). We perform a chi-squared test against the null hypothesis that the distribution of the labels is uniform with the p-value threshold set to 0.01. We find that for **all** 80 models trained, there is a significantly higher than chance level NCG accuracy. Then, we perform a t-test between each robust model vs. natural training, with the null hypothesis being that the robust model has lower NCG accuracy. In Table 2, we show the number of cases that pass this t-test. Adversarially robust models, TRADES and AT, almost always have a higher NCG accuracy than natural training with verified statistical significance. Meanwhile, mixup have a higher NCG accuracy than natural training in only less than half of the cases.

Adversarial robustness increases NCG accuracy. The unseen class is absent at training, and this property has been obtained simply by making the model adversarially robust. This is interesting because it suggests that robust models extrapolate to OOD data in a way that is more likely to match 1-NN predictions. Prior work on extrapolation (Xu et al., 2020) has shown that MLPs tend to extrapolate as linear functions on far OOD data. The 1-NN classifier is not a linear function in their sense. Specifically, the 1-NN label is determined by the Voronoi decomposition of the training data, and the decision boundaries separate far away points using hyperplanes. Hence, the higher NCG accuracy of robust classifiers uncovers a new phenomenon of OOD behavior. We next investigate whether we should expect higher NCG accuracy for robust models by measuring the distance to OOD examples.

Table 1: The average and standard deviation of the NCG accuracies across different held out class of each dataset. There are 10 unseen classes for MNIST and 3 unseen classes for CIFAR10, CIFAR100, and ImgNet100. In general robust methods have a higher average NCG than natural training. The chance level is 1/9 for MNIST and CIFAR10, 1/19 for CIFAR100, and 1/99 for ImgNet100.

	MINST	CIFAR10	CIFAR100	ImgNet100
		pixel		
natural	$.49 \pm .14$	$.24 \pm .09$	$.18 \pm .03$	$.04 \pm .01$
mixup	$.47 \pm .14$	$.23 \pm .09$	$.20 \pm .06$	$.04 \pm .01$
TRADES(2)	$.59 \pm .12$	$.34 \pm .12$	$.29 \pm .09$	$.04 \pm .01$
TRADES(4)	$.59 \pm .10$	$.37 \pm .11$	$.29 \pm .10$	$.05 \pm .01$
TRADES(8)	$.52 \pm .12$	$.34 \pm .10$	$.29 \pm .13$	$.06 \pm .01$
AT(2)	$.60 \pm .10$	$.36 \pm .12$	$.26 \pm .07$	$.04 \pm .01$
		feature		
natural	$.50 \pm .18$	$.82 \pm .02$	$.66 \pm .03$	$.12 \pm .01$
mixup	$.56 \pm .16$	$.79 \pm .02$	$.66 \pm .03$	$.14 \pm .01$
TRADES(2)	$.58 \pm .15$	$.82 \pm .01$	$.72 \pm .02$	$.16 \pm .01$
TRADES(4)	$.64 \pm .10$	$.85 \pm .02$	$.71 \pm .02$	$.13 \pm .01$
TRADES(8)	$.67 \pm .11$	$.85 \pm .02$	$.71 \pm .02$	$.14 \pm .01$
AT(2)/(1)/(.5)	$.54 \pm .17$	$.85 \pm .03$	$.73 \pm .02$	$.16 \pm .01$

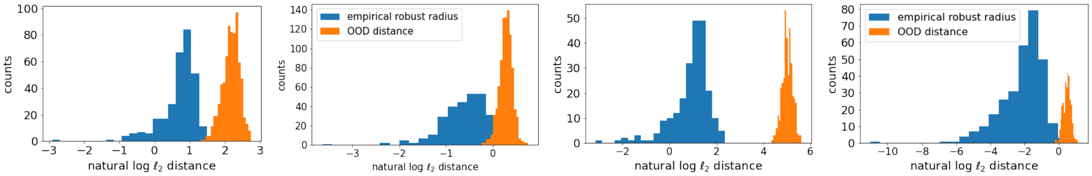
Table 2: The number of robust models with higher NCG accuracy than natural training. For MNIST we check 10 unseen classes, and for CIFAR10, CIFAR100, and ImgNet100, we use 3 unseen classes. 10/10 means that out of the 10 unseen classes, all 10 models have higher NCG.

	pixel				feature			
	M	C10	C100	I	M	C10	C100	I
mixup	5/10	0/3	2/3	0/3	0/10	0/3	2/3	3/3
TRADES(2)	10/10	3/3	3/3	3/3	9/10	2/3	3/3	3/3
TRADES(4)	8/10	3/3	3/3	3/3	10/10	3/3	3/3	3/3
TRADES(8)	7/10	3/3	3/3	3/3	10/10	3/3	3/3	3/3
AT(2)/(1)/(.5)	10/10	3/3	3/3	3/3	9/10	3/3	3/3	3/3

3.1 OOD Data are Farther than Adversarial Examples

Why do robust models have a higher NCG accuracy for unseen classes? One plausible explanation is that the robust methods enforce the neural network to be locally smooth in a ball of radius r; if the OOD inputs are closer than r from their nearest training example, then they would get classified accordingly. Next, we test if this is the case by measuring the ℓ_2 distance between each OOD input $\mathbf{x}$ and its closest training example $\tilde{\mathbf{x}}$. Then, we calculate the closest adversarial example $\mathbf{x}'$ to $\tilde{\mathbf{x}}$ using various attack algorithms. We measure (i) the distance between OOD example and its closest training example ($\|\mathbf{x} - \tilde{\mathbf{x}}\|_2$) and (ii) the empirical robustness radius ($\|\mathbf{x}' - \tilde{\mathbf{x}}\|_2$). We plot the histogram in Figure 2. We use several different attack algorithms to find the closest adversarial example. Some of these attacks are slow, so we compute adversarial examples for 300 randomly sampled training examples that are correctly predicted. Then, we restrict to OOD examples that have one of these 300 training examples as their closest neighbor.

Figure 2 (a) reports a typical distance histogram in the pixel space (for C10-0). We find that the histograms of OOD distances and the empirical robust radii have little to no overlap in the pixel space, while in the feature space, there is some overlap but not much. To better understand what is happening, we measure the percentage of OOD examples that are covered in the ball centered around the closest training example with a radius of the empirical robust radius. We find that in both the pixel and feature space, for 186 out of 190 models, this percentage is less than 2%, which is significantly smaller than the difference between the NCG accuracy of robust and naturally trained models in most cases (190 comes from having two metric spaces, five models, and 19 datasets). This result shows that almost all OOD examples from unseen classes are significantly further away from their closest training example than the empirical robust radius of these training examples.

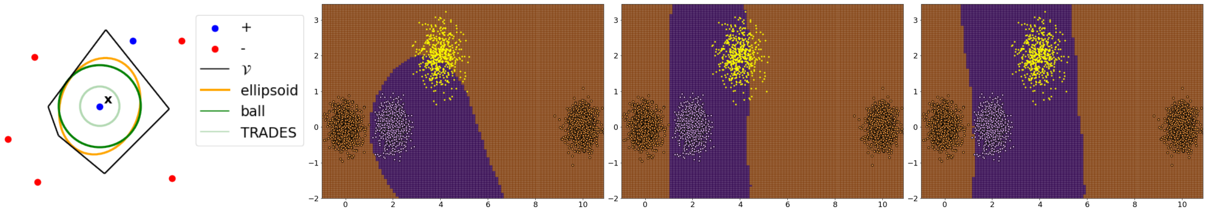


(a) C10-wo0, pixel space (b) C10-wo0, feature space (c) I-wo0, pixel space (d) I-wo0, feature space

Figure 2: Histograms: log of the empirical robust radius and OOD distance for TRADES(2) on CIFAR10-wo0 and ImgNet100-wo0. Adversarial examples are much closer than OOD examples.

3.2 The Role of the Training Procedure

We next ask whether changing the robustness regions P_i when optimizing the loss in Equation (1) can change NCG accuracy. TRADES enforces smoothness in a region P_i that is to be a fixed radius r norm ball around each training example. Enforcing smoothness on fixed radius norm balls may not ensure good NCG accuracy. Figure 3 (c) shows an example – here, the purple points are closer to the orange cluster on the left and further from the orange cluster on the right. If we only enforce smoothness on a uniform ball (TRADES), the decision boundary for the purple points does not extend right enough. We explore making the classifier smooth in regions P_i that *adapt to the geometry of the dataset.*



(a) region diagram (b) natural (c) uniform ball (d) sub-Voronoi region

Figure 3: (a) A diagram showing the difference between the sub-Voronoi region ($\mathcal{V}$) and the ball $\mathcal{B}$ used to approximate it. In figure (b), (c), and (d), we plot the decision boundary of neural networks trained with natural training, TRADES, and enforcement on the smoothness in $\mathcal{V}$. The yellow examples are the OOD examples, and they are closer to the purple examples. In (b) and (c), we see that the predictions on the yellow examples are not consistent to the nearest neighbor; on the other hand, in (d), the yellow examples are predicted as purple.

Sub-Voronoi Region. Can we use the 1-NN classifier as a guide to actively improve NCG accuracy? While we do not see OOD inputs at training, we can encourage all points $\mathbf{x}$ that are closer to a training point $(\mathbf{x}_i, y_i)$ than to any other training point with a different label to be assigned to label y_i. In other words, we could set P_i to be the Voronoi region of $\mathbf{x}_i$ in the union of $(\mathbf{x}_i, y_i)$ and all other training points whose labels are different from y_i. We call this the *sub-Voronoi region* of $(\mathbf{x}_i, y_i)$. Figure 3 (a) shows an example.

Figure 3 (b), (c), (d) show how different training methods change the decision regions. Note that the yellow points are never seen during training or testing (they are just for illustration of hypothetical OOD inputs).

We draw the decision boundary for three methods – natural, TRADES, and training with P_i set to the sub-Voronoi region in Equation (1). All three perform well on in-distribution examples; however, unlike TRADES, optimizing over the sub-Voronoi region classifies all of the OOD examples with their nearest categories by putting the purple boundary in the correct location.

3.3 Approximations to the sub-Voronoi region

The loss in Equation (1) is minimized by an iterative procedure. At each iteration, we find the input in P_i that maximizes the regularization term. Running this requires being able to project onto P_i efficiently. While this can be done relatively fast when P_i is a constant radius ball, it is considerably more challenging for the sub-Voronoi region – which is a polytope with close to n constraints (one for each training point with label $\neq y_i$). Therefore, we consider three alternative approximations that are faster to project on and can be efficiently implemented during training.

Sub-sampled sub-Voronoi. Here, instead of the sub-Voronoi region, of $\mathbf{x}_i$ in the full training data, we use the Voronoi cell of $\mathbf{x}_i$ in the union of $(\mathbf{x}_i, y_i)$ and a subsample of the training data with labels not equal to y_i. Since the sub-sampled sub-Voronoi region can be large, which can cause the network to underfit, we introduce a shrinkage parameter $\lambda \in [0, 1]$ to scale down the size of the region.

Ellipsoid. An alternative method is to use an ellipsoidal approximation. Computing the maximum volume ellipsoid inside the region is again challenging. To improve efficiency, we use the following approximation. We pick k differently-labeled training examples that are closest to $\mathbf{x}_i$, learn a PCA on these k examples, and pick the ellipsoid centered at $\mathbf{x}_i$ and described by the top $k/2$ principal components. We use a shrinkage parameter λ to help generalization.

Non-uniform ball. A final approximation is to use an ℓ_2 ball of radius r_i, where r_i is set to half the distance between $\mathbf{x}_i$ and its closest training point with a different label; this is the largest ball centered at $\mathbf{x}_i$ that is contained in the sub-Voronoi region. As with the previous methods, with finite training data, this may overestimate the region where we should predict y_i and hence lead to underfitting; to address this, we again introduce a shrinkage parameter λ, setting P_i to $\mathcal{B}(\mathbf{x}_i, \lambda r_i)$. More details about the minimization procedure and each of these alternatives are given in Appendix B.

3.4 Experiments on how the robust region affects NCG

We now empirically measure how the role of changing P_i affects NCG accuracy on real data. For this purpose, we consider enforcing smoothness in the three types of regions discussed above – non-uniform ball, ellipsoid, and sub-sampled sub-Voronoi. We also look at TRADES with three different radii and natural training as baselines. A detailed discussion of the experimental setup is in Appendix C. Note that we do not aim to achieve the best performance in any given measure, so we mostly use standard parameter settings, and we do not use data augmentation.

Results and Discussion. Table 3 shows the train, test, NCG accuracy for MNIST, CIFAR10, CIFAR100 with different unseen categories. All robust methods – TRADES and three approximations to sub-Voronoi – have higher NCG accuracy than the natural training. This agrees with our previous observations. Surprisingly, there is a lot of variation in the results. This warrants investigation, but we do not have a clear conjecture as to why certain classes have higher NCG accuracy. For instance, in M-4, the NCG accuracy can be up to 83%, while in M-1, the best is 53%. Perhaps there is a visual similarity between some classes of images (e.g., 4s and 9s look alike), or spurious correlations (Veitch et al., 2021), or only some datasets have sufficient diversity in examples to "cover" the OOD class (Hein et al., 2019; Xu et al., 2020). In high dimensional space, it is hard to measure whether some of the unseen classes are more like the training data than others. We later explore the NCG accuracy as a function of the distance to the training set. At least in pixel space and for color images, we find that closer images are more likely to receive the NCG label. We next consider corrupted images as another source of OOD data.

Table 3: The training, testing and NCG accuracy of networks trained by enforcing smoothness on different regions (pixel space). We use MNIST with digits 0, 1, 4, and 9 as the unseen classes, CIFAR10 with *airplane* and *deer* as the unseen classes, and CIFAR100 with *aquatic mammals* and *fruit and vegetables* as the unseen classes.

	trn acc.	tst acc.	NCG acc.	trn acc.	tst acc.	NCG acc.
	MNIST-wo0 (M-0)			MNIST-wo1 (M-1)		
sub-voronoi	0.981	0.981	0.474	0.982	0.981	0.376
ellipsoid	0.981	0.981	0.476	0.982	0.980	0.425
ball	0.975	0.973	**0.510**	0.976	0.973	0.338
TRADES	0.954	0.956	0.485	0.975	0.974	**0.528**
nat	1.000	0.995	0.390	1.000	0.995	0.273
	MNIST-wo4 (M-4)			MNIST-wo9 (M-9)		
sub-voronoi	0.982	0.983	**0.827**	0.988	0.988	0.703
ellipsoid	0.982	0.982	0.820	0.988	0.988	**0.725**
ball	0.977	0.976	0.795	0.982	0.981	0.711
TRADES	0.988	0.987	0.810	0.962	0.964	0.703
nat	1.000	0.995	0.760	1.000	0.996	0.577
	CIFAR10-wo0 (C10-0)			CIFAR10-wo4 (C10-4)		
sub-voronoi	0.735	0.658	0.452	0.486	0.482	**0.417**
ellipsoid	0.671	0.613	0.472	0.483	0.481	0.409
ball	0.794	0.618	**0.530**	0.871	0.664	0.317
TRADES	0.870	0.660	0.520	0.862	0.643	0.355
nat	1.000	0.900	0.362	1.000	0.886	0.222
	CIFAR100-wo0 (C100-0)			CIFAR100-wo4 (C100-4)		
sub-voronoi	0.308	0.289	0.241	0.706	0.499	**0.207**
ellipsoid	0.633	0.478	0.255	0.466	0.385	0.198
ball	0.936	0.517	0.236	0.930	0.489	0.176
TRADES	0.891	0.534	**0.264**	0.995	0.534	0.193
nat	1.000	0.757	0.169	1.000	0.694	0.140

4 NCG for Corrupted Data

Do the trends that we have seen for NCG also hold for other OOD data besides unseen classes? We consider images with Gaussian noise, blur, JPEG artifacts, snow, speckle, etc Hendrycks & Dietterich (2019). We use "-C" to denoted the corrupted version of a dataset, e.g., CIFAR10-C (C10-C), CIFAR100-C (C100-C), and ImgNet100-C (I-C), which consists of corrupted images from the CIFAR10 (C10), CIFAR100 (C100), and ImgNet100 (I) datasets. C10-C and C100-C have 18 kinds of corruption, each with 5 corruption levels. I-C has 15 kinds of corruption, each with 5 levels. We consider models trained on regular datasets, C10, C100, and I (instead of removing the unseen class). We use *corrupted set* to refer to a corruption type and intensity level. For C10 and C100, there are 90 corrupted sets for each dataset; for I, there are 75 corrupted sets. We consider NCG under ℓ_2 distance for both the pixel and learned feature spaces. We train on C10, C100, and I and measure NCG accuracy on C10-C, C100-C, and I-C, respectively; each training method is measured on 255 corruption sets.

Results. In both pixel and feature space, we find that **all** the 255 corruption sets have an NCG accuracy above chance level. For robust models, we find that in the pixel space, TRADES(2) has an NCG accuracy higher than naturally trained models on **all** 255 corrupted sets. Quantitatively, on average (over the 90 and 75 corrupted sets), TRADES has an NCG accuracy that is $1.35 \pm .02$, $1.36 \pm .03$, and $1.66 \pm .04$ times higher than naturally trained models for CIFAR10, CIFAR100, and ImgNet100 respectively. Hence, in pixel space, the TRADES training procedure leads to much higher NCG accuracy for corrupted data. On the other hand, in the feature space, the results are much less conclusive. In particular, the robust models still have higher NCG accuracy on average, but this does not happen consistently across corruption types or datasets (see Appendix D.6).

Discussion. Our findings in Section 3 extend to these corruptions as the OOD data. In particular, the NCG accuracy of adversarially robust networks is higher on both unseen, natural images, and on corrupted versions of seen classes. On the other hand, in the feature space, the robust models do not have much difference in NCG accuracy from the naturally trained models. The fact that we see less variation in the feature space compared to the pixel space is consistent with our results on the unseen classes. The TRADES robust training does not affect the decision regions as much when the smoothness is enforced in the feature space. This is likely because the learned embedding has less variation in elements of the same class, and hence, the TRADES training does not contribute as much.

4.1 NCG accuracy vs. test accuracy

Next, we look at the interaction between the NCG and test accuracies, so we also measure the test accuracy on the NCG correct data and NCG incorrect data. We first observe that NCG correct examples are more likely to be correctly classified. To verify that this phenomenon is statistically significant across the board, we perform the one-sided Welch's t-test (which does not assume equal variance) with the null hypothesis being that the accuracy of NCG correct example is not greater than the accuracy of NCG incorrect example. We set the p-value threshold to 0.05, and the test results are in Table 4. For more details, please refer to Appendix D.3.

Table 4: Number of cases where the NCG correct examples have a **significantly** higher test accuracy than the NCG incorrect examples. 87/90 means that out of the 90 corrupted sets, 87 of them pass the t-test.

	pixel			feature		
	C10	C100	I	C10	C100	I
natural	87/90	87/90	57/75	88/90	90/90	73/75
TRADES(2)	84/90	88/90	60/75	89/90	90/90	73/75

5 Discussion and Connections

5.1 Sample complexity of NCG vs. OOD detection

We first observe that NCG is an easier problem in theory than OOD detection. Indeed, any time we can detect an OOD example, then we can use the 1-NN classifier to label it. Thus, the NCG accuracy should always be at least as high as the true positive rate for OOD detection.

We next theoretically show that the converse is false in general. We prove that there exist cases where maximizing NCG accuracy can be significantly more sample efficient than solving detection problems. OOD detection is hard when certain regions of space have low mass under the input distribution. In this case, it takes many samples to see a representative covering of the support. Prior work has made similar high-level observations in the context of data diversity for robustness and extrapolation (Hein et al., 2019; Xu et al., 2020). In contrast, when the NCG label is the same for nearby regions, then it suffices to see samples from fewer regions and generalize accordingly. We formalize this claim in the following theorem, which identifies simple distributions where detection requires many more samples than NCG. While the proof is not complicated, our result shows that NCG can be much easier than detection.

Theorem 1 *For any $\epsilon \in (0, 1/2)$, $d \geqslant 1$, and $C \geqslant 2$, there exists distributions μ on training examples from C classes in $\mathbb{R}^d$ and ν on OOD test examples from outside of $\mathsf{supp}(\mu)$ such that (i) detecting whether an example is from μ or ν requires $\Omega(C/\epsilon)$ samples from μ, while (ii) classifying examples from ν with their nearest neighbor label from the support of μ requires only $O(C \log C)$ samples.*

Figure 4 shows intuition for Theorem 1 in the binary case. OOD examples come from outside of the colored cubes. Appendix A has the proof for C classes in $\mathbb{R}^d$.

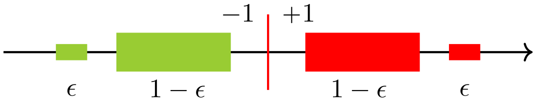


Figure 4: An example for Theorem 1. The sample frequency is the size of the red/green shapes. With a few samples from each large probability region, we determine the NCG label via a large margin solution, but OOD detection requires samples from the small probability regions.

We sketch how to generalize Figure 4 for more classes and higher dimensional data. The idea is that we translate and replicate the binary dataset and increase the regions to d-dimensional cubes. For the distribution μ, we have $4C$ cubes with side length $1/\sqrt{d}$. There will be 2 cubes that have labels from each of the C classes. The high probability cubes emit samples with probability $\approx (1-\epsilon)/C$ and the lower probability with $\approx \epsilon/C$. Due to the side lengths being $1/\sqrt{d}$, the 1-nearest neighbor (1-NN) in ℓ_2 of the low probability region is paired with an adjacent high probability box, and hence, it is easy to predict given samples from the high probability region. By a coupon collector argument, we see all high probability regions after $O(C \log C)$ samples. On the other hand, by the construction of the probability distributions, we need $\Omega(C/\epsilon)$ samples for OOD detection, where ϵ is the sample probability from a low probability region. For the OOD distribution ν, we strategically sample points from outside of all of these cubes (while guaranteeing that the nearest neighbor labels are still correct). Thus, $O(C \log C)$ samples are sufficient for NCG, but $\Omega(C/\epsilon)$ are needed for OOD detection.

This result suggests that there are cases where OOD detection is extremely sample inefficient and cannot be done well. In these cases, the model will have to give a prediction on examples that it does not expect. Thus, it would be crucial to understand how the model predicts these examples.

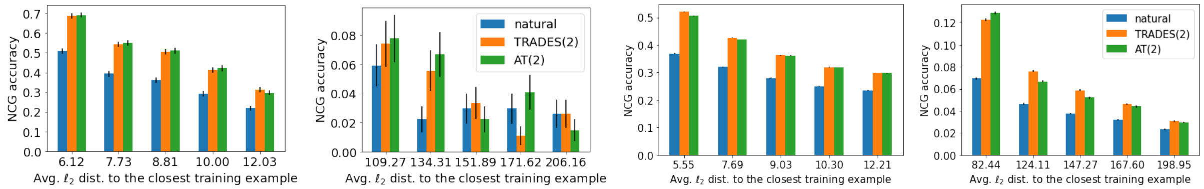
As a concrete example, we train a model on MNIST images of 0-8 and use the model for prediction on images of 9s. We also train an OOD detector – ODIN Liang et al. (2018) – which has a .951 true positive rate and .875 false negative rate. In this example, many images of 9s cannot be easily picked out by OOD detectors and will be treated as in-distribution examples. Thus, it is important to know what kind of prediction will be given to these 9s. From our previous result, we find that many 9s are predicted as 4s, and this can be explained by nearest category generalization.

5.2 When do we have higher NCG accuracies?

One hypothesis is that OOD examples that are further from the training set are less likely to be predicted with the NCG label. To check this hypothesis, we conduct the following experiment. We bin the OOD examples based on their distance to the closest training example into 5 equal size bins, and we evaluate the NCG accuracy in each bin. A typical result is shown in Figure 5 (additional results are in Appendix D.5.3). We find that the NCG accuracy is generally higher when OOD examples are closer to the training examples. This is true both for an unseen class and for corrupted data (in aggregate). While this is not surprising, it does give more insight into the patterns of OOD data that are labeled with the 1-NN label. We also looked at MNIST, but there was no clear connection between distance and NCG accuracy.

It is known that OOD detectors perform well when in- and out-of-distribution data are far away from each other (Liang et al., 2018). This, along with our result, gives us an interesting dynamic, which is that neural networks behave more like the nearest neighbor classifier when detectors perform worse. This means that many of the OOD examples that are misclassified as in-distribution examples could follow the NCG property. It would allow the user to know that even when the OOD detector fails, the model would still output something reasonable. Thus, if one wants a robust and predictable prediction on OOD data, it can be desirable to have a high NCG accuracy.

Limitation: Choice of the distance metric. We only evaluate ℓ_2 distance in the pixel and feature space. With ℓ_2 distance, we already discovered interesting OOD behavior. However, an important direction is to explore other distance measures to understand the prediction patterns. Some options include ℓ_∞ or cosine



(a) CIFAR10-wo0 (b) ImgNet100-wo0 (c) CIFAR10 (corrupted) (d) ImgNet100 (corrupted)

Figure 5: We group OOD examples into five bins based on ℓ_2 distance to the closest training example in pixel space. (a) and (b) show the NCG accuracy of each bin on the unseen class of CIFAR10-wo0 and ImgNet100-wo0. (c) and (d) the NCG accuracy of each bin on the aggregate corrupted data of CIFAR10 and ImgNet100. The downward trend seen here is not as apparent in the feature space (see Appendix D.5.3).

distance or measuring distance in a embedding space of an auto-encoder. Ideally, the distance measure would capture perceptual similarity of the images. This would imply that NCG accuracy corresponds to how humans may predict an unseen class. However, it is not clear if such a perceptual metric exists for images.

6 Conclusion

We examine out-of-distribution (OOD) properties of neural networks and uncover intriguing generalization properties. Neural networks have a tendency of predicting OOD examples with the labels of their closest training examples. We measure this via a new metric called NCG accuracy. Robust networks consistently have higher NCG accuracy than naturally trained models. We replicate this result for two sources of OOD data (unseen classes and corrupted images), and we experiment with a variety of new and existing robust training methods. This is surprising because the OOD data are much further away in ℓ_2 distance than both the robustness radius and the nearest adversarial examples. Therefore, the robust training procedure is changing the decision regions on parts of space that are not directly considered in the loss function. We posit that this behavior and the higher NCG accuracy is a consequence of the inductive bias of robust networks. In the future, it would be interesting to evaluate NCG for other training methods, architectures, and sources of OOD data like distribution shift or spurious correlations. Overall, NCG can be a valuable and scalable addition to the toolbox of evaluation metrics for OOD generalization.

References

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

Randall Balestriero, Jerome Pesenti, and Yann LeCun. Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*, 2021.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. *arXiv preprint arXiv:1907.01003*, 2019.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

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

Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773*, 2019.

Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness. *arXiv preprint arXiv:2002.06789*, 2020.

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. *Advances in Neural Information Processing Systems*, 33, 2020.

Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1021–1030, 2020.

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

Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *International Conference on Machine Learning*, pp. 2596–2604. PMLR, 2019.

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

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41–50, 2019.

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

Charles Herrmann, Kyle Sargent, Lu Jiang, Ramin Zabih, Huiwen Chang, Ce Liu, Dilip Krishnan, and Deqing Sun. Pyramid adversarial training improves vit performance. *arXiv preprint arXiv:2111.15121*, 2021.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Jörn-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. *arXiv preprint arXiv:1811.00401*, 2018.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

Been Kim, Emily Reif, Martin Wattenberg, Samy Bengio, and Michael C Mozer. Neural networks trained on natural scenes exhibit gestalt closure. *Computational Brain & Behavior*, pp. 1–13, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. *arXiv preprint arXiv:2103.02325*, 2021.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International Conference on Machine Learning*, pp. 5436–5446. PMLR, 2020.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Hyun Kwon, Yongchul Kim, Ki-Woong Park, Hyunsoo Yoon, and Daeseon Choi. Multi-targeted adversarial example in evasion attack on deep neural network. *IEEE Access*, 6:46084–46096, 2018.

Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Larry M Manevitz and Malik Yousef. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154, 2001.

Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. *arXiv preprint arXiv:1909.12180*, 2019.

Guillermo Ortiz-Jimenez, Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Hold me tight! influence of discriminative features on deep network boundaries. *arXiv preprint arXiv:2002.06349*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. 2019.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Advances in neural information processing systems*, pp. 3347–3357, 2019.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pp. 14707–14718, 2019.

Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv preprint arXiv:2110.14051*, 2021.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *arXiv preprint arXiv:2007.08489*, 2020.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*, 2020.

Vikash Sehwag, Mung Chiang, and Prateek Mittal. Ssd: A unified framework for self-supervised outlier detection. *arXiv preprint arXiv:2103.12051*, 2021.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Chawin Sitawarin, Supriyo Chakraborty, and David Wagner. Improving adversarial robustness through progressive hardening. *arXiv preprint arXiv:2003.09347*, 2020.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *arXiv preprint arXiv:2007.08176*, 2020.

Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *International Conference on Machine Learning*, pp. 9561–9571. PMLR, 2020.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pp. 9690–9700. PMLR, 2020.

Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2): 373–440, 2020.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*, 2021.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

Francis Williams, Matthew Trager, Claudio Silva, Daniele Panozzo, Denis Zorin, and Joan Bruna. Gradient dynamics of shallow univariate relu networks. *arXiv preprint arXiv:1906.07842*, 2019.

Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848*, 2020.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *arXiv preprint arXiv:2003.02460*, 2020.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

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

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.