Robustness and Generalization to Nearest Categories

Yao-Yuan Yang
yay005@eng.ucsd.edu
UCSD

Cyrus Rashtchian
crashtchian@eng.ucsd.edu
UCSD

Ruslan Salakhutdinov
rsalakhu@cs.cmu.edu
Carnegie Mellon University

Kamalika Chaudhuri
kamalika@cs.ucsd.edu
UCSD

Abstract

There has been much recent interest in out-of-distribution (OOD) generalization, where inputs from unseen classes are presented to a neural network. In this work, we take a closer look at how neural networks predict for these problems. Psychologists suggest that humans tend to categorize inputs from unseen classes to the category of the closest seen example; we investigate whether neural networks also behave similarly. We formalize this behavior as the *nearest category generalization* (NCG) problem and design experiments to explore whether it is happening in neural networks. We find that neural networks do tend to follow NCG for unseen classes in pixel as well as feature spaces, and the accuracy for NCG is typically higher in feature space. This suggests that for unseen classes, neural networks often predict the class of the closest training input in the feature space. Additionally, we see that adversarially robust neural networks have more enhanced NCG properties. Finally, we investigate whether this also happens for other kinds of OOD inputs beyond unseen classes, such as data with natural corruptions.

1 Introduction

Recently, there has been much interest in various aspects of out-of-distribution (OOD) generalization, such as transfer learning (Salman et al., 2020), outlier detection (Meinke and Hein, 2019), and few-shot learning (Koch et al., 2015). We want to understand the output of neural networks on OOD inputs, and whether there are patterns in the predicted values. By observing the outputs of a neural network with OOD inputs in Table 1, we find that there are indeed some patterns. The question is, what is this pattern? A line of work in the psychology literature posits that humans categorize unseen examples into the most similar category they have seen before (Nosofsky, 1986; Rouder and Ratcliff, 2004; Austerweil et al., 2019; Sanborn et al., 2021). For example, when a child sees an orange for the first time, he may categorize an orange as a type of similar fruit he has seen before, such as a tangerine. Inspired by this unique tendency of humans, we investigate whether neural networks show similar behavior.

	removed class	top most predicted class	second most predicted class
CIFAR10	airplane	ship	bird
CIFAR100	aquatic mammals	fish	small mammals

Table 1: We remove images of a class from training set of CIFAR10 and CIFAR100, and train a neural network on the modified trianing set. We then look at the predictions of the neural network on these removed images and record their top two most predicted classes. From the result, we can see that the outputs that these two networks produce follow some patterns. "airplanes" are predicted as a ship or bird possibly because they have similar background of the sky. "aquatic mammals" are predicted as fish possibly because they are both in the water.

We test whether neural networks also tend to predict OOD examples as the nearest category in the training set, and we call this property Nearest Category Generalization (NCG). We begin with setting up a framework for testing whether neural networks show signs of NCG. We use images from an unseen class as the OOD examples. We take existing datasets, remove examples of a certain class from the training set, and treat the removed examples as OOD examples. We then train a neural network on the training set and examine its prediction of these OOD examples. If a significant amount of OOD examples are classified as the same category as their nearest training example, then it shows that there are some particular structures

Preprint. Under review.

in the prediction of the unseen class. We define the *NCG accuracy* as the portion of OOD examples that are predicted as the same label as their nearest training example (while measuring in-distribution accuracy as usual).

Building on this testing framework, we measure the NCG property of neural networks. We consider four datasets and select ten different unseen classes for each dataset. We train a neural network on each of these 40 different combinations of datasets and unseen classes. We find that the NCG accuracies of **all** networks are significantly above the chance levels. This shows concrete evidence that neural networks follow NCG property to predict the unseen class (instead of predicting randomly).

Adversarial robust neural networks are trained to produce smooth predictions when the input is slightly altered. This is another important ability that humans also possess (Yang et al., 2020; Zhang et al., 2019; Madry et al., 2017). Does making the network more smooth (or robust) affect their NCG property? We repeat the previous experiments with robust networks and find that robust networks not only have an NCG accuracy above chance, but also generally have a higher NCG accuracy than the naturally trained models. This indicates improving adversarial robustness may make the model follow NCG more rigorously, and there are certain connections between adversarial robustness and the NCG property.

Why do robust networks generally have higher NCG accuracies? A plausible explanation for why this may happen is that robust training algorithms like TRADES (Zhang et al., 2019) enforce the network to be locally smooth in a ball of radius r around training data (Yang et al., 2020); if the OOD inputs are closer than r from their nearest training example, then they would get classified in the same class. Surprisingly, we find that this is not the case. Balls of radius r around most training examples are so small that they cover almost none of the OOD inputs. Moreover, OOD inputs that are classified with their nearest categories are considerably further than from their closest training examples, which, in turn, continues to have adversarial examples that are closer than the robustness radius r (see Figure 1). This suggests that the robust neural networks may be smoother in some directions than others, and perhaps smoother than they were trained to be along the natural image manifold.

A natural question to ask is – does NCG extend to other types of OOD data beyond the unseen classes? To answer the question, we look at the corrupted data including CIFAR10-C, CIFAR100-C, and ImgNet100-C proposed by Hendrycks and Dietterich (2019). We have

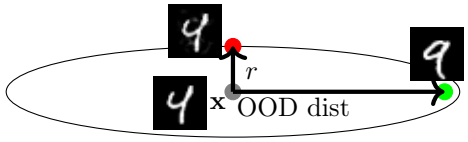


Figure 1: Robust networks tend to predict smoothly at a larger distance in some directions, e.g., toward natural OOD examples (green point), but are susceptible to adversarial examples that are closer in the worst-case directions (red point).

Figure 2: OOD examples (b) and (d) are far in pixel space from their nearest training examples (a) and (c). Surprisingly, (b) is predicted as a 4 and (d) as a 7, indicating the network is smooth in these directions.

three observations. First, the NCG accuracies for **all** networks (including natural and robust networks) are above the chance levels, and the NCG accuracies of robust networks are also generally higher than natural networks. This result allows us to extend our previous findings to many kinds of OOD data. Second, corrupted examples that are correct in terms of NCG accuracy have a higher chance of being classified correctly. Third, in general, the test and NCG accuracies of a network decreases as the intensity of corruption increases. We find that robust models have a slower rate of decrease comparing with naturally trained models. The second and third observations suggest that different forms of robustness, including adversarial robustness, the robustness to corruptions, and the NCG accuracy, may be inherently interconnected.

In summary, our work uncovers an intriguing out-of-distribution generalization property of neural networks called the nearest category generalization and investigates it in detail. We have identified that the NCG property exists for many neural networks and OOD types. We also show a connection among the NCG property, adversarial robustness, and robustness to corrupted data. We posit that the NCG property is a consequence of the inductive bias produced by neural networks (especially for adversarially robust networks). It is interesting that this inductive bias happens to be similar to some human behaviors and enforcing adversarial robustness, which is another feature that humans possess, can make the NCG property more salient. Many scholars conjecture that the effectiveness of deep learning may be coming from its similar structure to the human brain (Lake et al., 2017; Hassabis

et al., 2017; Sejnowski, 2020), which allows the neural networks to share some of the inductive biases from the brain. This work can be an additional piece of evidence supporting this theory. In addition, how neural networks generalize so well is still an open question (Liu et al., 2020). Our work provides some insights into how networks generalize, and we expect future work to build upon this knowledge.

2 Preliminaries

Nearest Category Generalization. At training time, we are given a set of examples $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$ from one of C categories. At test time, we evaluate on examples drawn from a combination of the training distribution and from a new $(C+1)$-st category. Examples from categories $\{1, 2, \ldots, C\}$ are considered in-distribution and those from class $C+1$ out-of-distribution. For example, we may see the MNIST classes $0-8$ at training time, and all MNIST classes at test time. We call the set of in-distribution test examples the *test set*, and the set of OOD test examples the *OOD set*. In addition to test accuracy, we also look at the NCG accuracy, which is the fraction of inputs from the $(C+1)$-st category that is assigned the same label as its nearest neighbor in the training data. Throughout, we use the shorthand `dataset`-wo# to mean that this class number is the unseen class (category). For example, we let MNIST-wo0 and MNIST-wo9 are MNIST with unseen digits 0 and 9, CIFAR10-wo0 is CIFAR10 with unseen *airplane* and CIFAR100-wo0 is CIFAR100 with unseen *aquatic mammals*. We sometimes shorten this as M-0, C10-0, C100-0, etc.

Distance metric. We need to specify a distance metric for the nearest neighbor. We use ℓ_2 distance in the pixel space, which is a commonly used distance metric; however, it may not provide much semantic information, which is important in some cases. Therefore, in the experiment, we also consider ℓ_2 distance in the feature space. In the pixel space, we use the original image as the input to the neural network. In the feature space, we first train a neural network on the training set (without the unseen class), and then we use this network to extract the features of each image in the training, testing, and OOD set (forming a new training, testing and OOD set). Finally, we train a fully connected multi-layer perceptron on the new training set and evaluate the test and NCG accuracy on the new testing and OOD set.

Adversarial Robustness. Let $\mathcal{B}(\mathbf{x}, r)$ denote a ball of radius $r > 0$ around $\mathbf{x}$ in a metric space $(\mathrm{X}, \mathsf{dist})$. 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 $f(\mathbf{x}') = f(\mathbf{x})$. Typically, we require classifiers to be robust at points x that are drawn from the underlying data distribution. Popular solutions for training robust classifiers are adversarial training (AT) (Madry et al., 2017) and TRADES (Zhang et al., 2019). These methods ensure robustness by encouraging the network to be more locally Lipschitz (smooth) on a ball of radius r around each training point, where r is usually small.

3 Nearest Category Generalization

We begin with experiments to test out whether neural networks generalize to the nearest category.

Datasets. We experiment with four datasets: MNIST (LeCun et al., 2010), CIFAR10 (Krizhevsky, 2009), CIFAR100 (Krizhevsky, 2009), and ImgNet100 (an ImgNet (Deng et al., 2009) subset with 100 classes[1]). For MNIST and CIFAR10, there are 10 distinct classes; for CIFAR100, we use the coarse labeling, which has 20 classes; for ImgNet100, there is 100 distinct classes. For all four datasets, we consider 10 different unseen category combinations for each dataset (i.e., MNIST-wo0, ..., MNIST-wo9, CIFAR10-wo0, ..., CIFAR10-wo9, CIFAR100-wo0, ..., ImgNet100-wo9), which gives us a total of 40 dataset.

Results. We train neural networks on the training set of these 40 datasets with a standard training method (natural) and measure their NCG accuracies. We perform a chi-square test against the null hypothesis that the distribution of the labels is uniform, which gives a chance-level NCG accuracy. With the p-value smaller than 0.01, **all** networks trained on these 40 datasets have an NCG accuracy significantly higher than the chance level. We repeat the experiment in the feature space and observed similar results. In addition, we see many of the networks trained in the feature space have their NCG accuracies being higher than the networks trained in the pixel space. Partial results are shown in the "natural" row of Table 2, and the full table can be found in Appendix B.1.

3.1 Adversarial robust networks

Adversarial robustness is another feature that human possesses, and the machine learning models are still trying to acquire this feature (Goodfellow et al., 2014). Here, we investigate whether there is a connection between the adversarial robustness and NCG property.

Training methods. We consider two of the most commonly used methods for making networks robust to adversarial examples: Adversarial Training (Madry et al., 2017)(AT) and TRADES (Zhang et al., 2019)

[1]Following https://github.com/HobbitLong/CMC/blob/master/imagenet100.txt

with perturbation distance metric set to the ℓ_2. For TRADES, we use robustness radii $r \in \{2, 4, 8\}$. We find that the training process in AT becomes unstable at larger values of r; hence we only use $r = 2$ for AT. In the feature space, we set $r = 1$ for AT on CIFAR10 and CIFAR100, and $r = .5$ for AT on ImgNet100 since CIFAR10 and CIFAR100 failed to converge with $r = 2$ and ImgNet100 failed to converge with $r = \{2, 1\}$. We denote TRADES with $r = 2$ and AT with $r = 1$ as TRADES(2) and AT(1), respectively. Prior work has observed that AT and TRADES provide roughly similar results with proper parameter tuning (Yang et al., 2020; Carmon et al., 2019), and hence we expect them to behave similarly. Appendix A has more details for the experimental setup.

Datasets. Since training adversarial robust networks are time-consuming, we only use consider 3 datasets from each of CIFAR10, CIFAR100, and ImgNet100 (we still consider all 10 datasets for MNIST). 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. We measure the NCG accuracy of the models trained on these datasets. Table 2 shows some typical results, for full details, please refer to Appendix B. As an aggregated result, we find that for **all** models trained, both in pixel and feature space, we have a higher than chance level NCG accuracy. In Table 3, we show a comparison of the NCG accuracy between robust models and naturally trained models. We see that in most cases, TRADES and AT have a higher NCG accuracy than natural training, thus showing that robust models tend to predict images of the unseen class with the same class as their nearest training example. We emphasize that since the unseen class was absent at training, this property has been obtained simply by making the model adversarially robust and not by optimizing for NCG accuracy.

Discussion. There are two particularly interesting observations. First, we see that models in the feature space generally have higher NCG accuracies than pixel space. One plausible explanation is that the nearest neighbor works better in the feature space. To support this, we measure the test accuracy of a 1-nearest neighbor classifier in the feature space (Appendix B.1). We find that in many cases, this test accuracy is very close to the test accuracy of neural networks trained in the feature space. This indicates that 1-nearest neighbor works well with the feature space distance metric, thus, we may get neural networks with higher

	M-0	M-9	C10-0	C100-0	I-0
			pixel		
natural	.39	.58	.35	.17	.03
TRADES(2)	.46	.69	.49	.25	.04
TRADES(4)	.48	.70	.52	.25	.05
TRADES(8)	.40	.66	.48	.21	.07
AT(2)	.46	.71	.49	.24	.04
			feature		
natural	.28	.66	.80	.63	.11
TRADES(2)	.39	.71	.81	.69	.15
TRADES(4)	.45	.73	.83	.68	.12
TRADES(8)	.58	.78	.83	.68	.13
AT(2)/(1)/(.5)	.32	.70	.83	.70	.16

Table 2: NCG accuracy for different algorithms on five datasets. M-0, M-9, C10-0, C100-0, I-0 mean MNIST-wo0, MNIST-wo9, CIFAR10-wo0, CIFAR100-wo0, ImageNet100-wo0 respectively. The chance level is $\frac{1}{9}$ for MNIST and CIFAR10, $\frac{1}{19}$ for CIFAR100, and $\frac{1}{99}$ for ImgNet100.

	pixel				feature			
	M	C10	C100	I	M	C10	C100	I
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

Table 3: The number of models that have a higher NCG accuracy than the naturally trained model. For MNIST, there are 10 different unseen classes, and for CIFAR10, CIFAR100, and ImgNet100, there are 3 different unseen classes. 10/10 means that out of the 10 datasets with different unseen classes, all 10 models have a higher NCG accuracy than the naturally trained model.

NCG accuracies than in the pixel space. Second, we observe that even within the same dataset, different unseen classes can have very different NCG accuracy. For example, the M-0 and M-9 datasets in the feature space of Table 2 has .28 and .66 NCG accuracy for naturally trained models. One plausible explanation is that an image of 9 can be similar to images of 7s or 1s, but an image of 0 is not particularly similar to other digits. This suggests that NCG accuracies can be significantly affected by the geometry of the dataset.

3.2 Robustness improves NCG

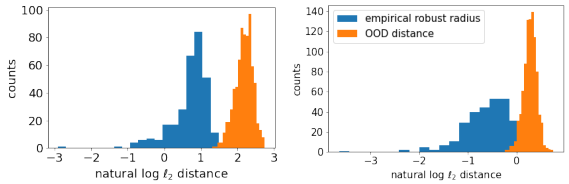
A natural question to ask is why robust models have a higher NCG accuracy for unseen classes. One plausible explanation is that the robust methods enforce the neural network to be locally smooth in a ball of radius r; if the OOD inputs are closer than r from their nearest training example, then they would get classified accordingly. Next, we test if this is the case

by measuring the distances between the OOD inputs and their closest training examples.

We again look at four datasets and four robust models. For each OOD $\mathbf{x}$ that is predicted with the same label as its closest training example $\tilde{\mathbf{x}}$, we calculate the distance $\mathsf{dist}(\mathbf{x}, \tilde{\mathbf{x}})$. Additionally, we calculate the closest adversarial example to $\tilde{\mathbf{x}}$ using various attack algorithms and take the closest adversarial example among them and denote it as $\mathbf{x}'$. We measure the OOD distances ($\mathsf{dist}(\mathbf{x}, \tilde{\mathbf{x}})$) and empirical robust radius ($\mathsf{dist}(\mathbf{x}', \tilde{\mathbf{x}})$) and then plot them in a histogram (Figure 3). Because some attack methods are computation-intensive, we only compute the adversarial examples for 300 randomly sampled correctly predicted training examples and consider OOD examples with one of these 300 training examples as their closest neighbor.

Figure 3(a) reports typical distance histograms in the pixel space (for CIFAR10-wo0); full result appears in Appendix B.3. 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 are some overlaps 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).

Discussion. This result shows that almost all OOD examples are significantly further away from their closest training example than the empirical robust radius of these training examples. This indicates that this property of adversarially robust models is not simply because the OOD inputs are close. Rather, even though they were not directly trained to do so, the robust models are generalizing better along unseen directions on the natural image manifold than arbitrary unseen directions.



(a) pixel space (b) feature space

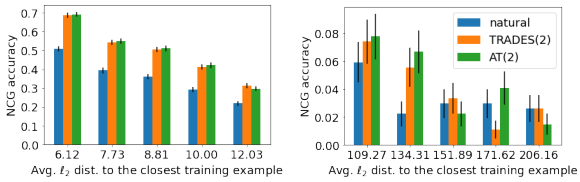
Figure 3: Here, we show the histograms of the empirical robust radius and log OOD distance for TRADES(2) trained on CIFAR10-wo0.

3.3 When do we have higher NCG accuracy?

A child who has never seen an orange before may be able to guess it is a tangerine. What if you show him an image taken from the surface of the moon, which is something completely out of his normal life, what might have he guess this time? It appears to us that when OOD examples are too far away from other training examples, it may be hard for neural networks to predict it as the label of the nearest training example.

To verify 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 4 (more details are in Appendix B.4). We find that the NCG accuracy is generally higher when OOD examples are closer to the training examples.

Discussion. An out-of-distribution detection algorithm is a common approach for dealing with OOD examples. However, Liang et al. (2018) point out that OOD detection can perform poorly when in- and out-of-distribution examples closer to each other. On the other hand, in the same situation, our result shows the networks follow NCG more strictly. The NCG property can be seen as the network being "robust" in terms of giving a reasonable output when the input is OOD. In the future, one can use the NCG property of neural networks to develop methods for tackling OOD examples that are close to in-distribution examples.



(a) CIFAR10-wo0 (b) ImgNet100-wo0

Figure 4: The NCG accuracy and the distance to the closest training example on CIFAR10-wo0 and ImgNet100-wo0 in the pixel space. The distance metric is in the pixel space, and the NCG accuracy is evaluated on TRADES(2). Similar phenomenon can also be found in the feature space (see Appendix B.4).

4 NCG with Corrupted Data

Does NCG apply to other kinds of OOD data besides unseen classes? In this section, we look at the case of corrupted data. We consider the corrupted data generated by Hendrycks and Dietterich (2019) and look at whether the NCG property holds on them. In addition, we also look at what kinds of relationships

there are between NCG, adversarial robustness, and the robustness towards these corrupted data.

The corrupted datasets that we consider here include CIFAR10-C, CIFAR100-C, and ImgNet100-C, which consists of corrupted images from the CIFAR-10, CIFAR-100, and ImgNet100 datasets. These datasets include images corrupted by effects such as Gaussian noise, JPEG artifacts, etc. Figure 5 shows an example and its corrupted counterpart from ImgNet100 and ImgNet100-C. CIFAR10-C and CIFAR100-C each have 18 different kinds of corruption, and each kind has 5 corruption levels. For ImgNet, due to computational constraints, we subsample it to 100 classes and constructed the ImgNet100-C dataset. ImgNet100-C has 15 kinds of corruption, and each corruption has 5 corruption levels. We consider models trained on regular datasets, CIFAR10, CIFAR100, and ImgNet100 (instead of removing the unseen class). For each corruption type and intensity level pair, we call it a *corrupted set*. For CIFAR10 and CIFAR100, there are 90 corrupted sets; for ImgNet100, there are 75 corrupted sets.

(a) original (b) Gaussian (c) fog

Figure 5: The original image of an American robin and images with two of the level 5 corruptions.

We want to verify whether the observations observed in Section 3 still hold for corrupted data, which is a different kind of OOD example. We evaluate the models trained on CIFAR10, CIFAR100, and ImgNet100 on each of the corrupted sets and measures their NCG accuracy. In other words, each training method will be measure on 255 different 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 see that in the pixel space, **all** robust models (TRADES(2)) have an NCG accuracy higher than naturally trained models. In the feature space, in general, robust models still have an NCG accuracy higher than the naturally trained models, however, not by a lot (see Appendix B.5).

Discussion. These results demonstrate that our findings in Section 3 extend to these corruptions as the OOD data. We also see that in the feature space, the robust models do not have much difference in NCG accuracy from the naturally trained models. One hypothesis is that there is no evidence showing that adversarial robustness in the neural network feature space is a feature that humans possess. Therefore, enforcing smoothness (or robustness) in such space may not give us much change over the NCG accuracy as did in the pixel space.

4.1 NCG accuracy vs. test accuracy

The original design of the corrupted datasets is to measure whether the models keep the same prediction after the corruption, thus, they measure the test accuracy on corrupted data as a metric for robustness to corruptions. We follow the same procedure as in the previous section while also evaluate the test accuracy on each of the corrupted sets. We say that an example is *NCG correct* if the prediction on that example is the same as the label of its closest training example – i.e. consider correct under the NCG accuracy[2]. We want to 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. In Table 5, we show the result of Gaussian noise as the corruption type with the model trained on CIFAR10, CIFAR100, and ImgNet100. This is a typical result; for other corruption types, please refer to Appendix B. The major findings are in Section 4.1.1 and 4.1.2.

4.1.1 NCG correct examples are more likely to be correctly classified

The first thing that we observed is 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. From the result, we can say that majority of the time, this phenomenon is significant.

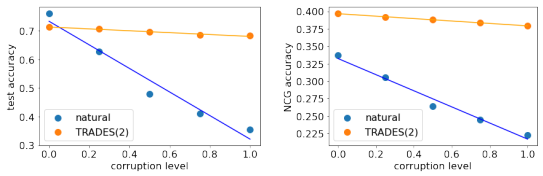
	pixel			feature		
	C10	C100	I	C10	C100	I
natural	87/90	87/90	57/75	88/90	90/90	73/75
TRADES(2)	84/90	88/90	60/75	89/90	90/90	73/75

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.

[2]Note that this is not obvious even in the feature space as neural networks are performing linear classification in the feature space instead of performing nearest neighbor classification.

		model	natural				TRADES(2)			
	dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
pixel	C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
		3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
		5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
	C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
		3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
		5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29
	I	1	0.42	0.41	0.68	0.04	0.36	0.35	0.51	0.06
		3	0.22	0.21	0.49	0.03	0.34	0.33	0.49	0.05
		5	0.04	0.04	0.07	0.02	0.22	0.22	0.34	0.04
feature	C10	1	0.74	0.39	0.78	0.89	0.72	0.32	0.77	0.89
		3	0.45	0.33	0.48	0.82	0.40	0.19	0.45	0.83
		5	0.34	0.28	0.35	0.82	0.31	0.18	0.33	0.83
	C100	1	0.60	0.25	0.72	0.74	0.62	0.29	0.71	0.78
		3	0.43	0.23	0.54	0.64	0.44	0.25	0.53	0.69
		5	0.37	0.21	0.46	0.61	0.37	0.21	0.46	0.65
	I	1	0.22	0.18	0.44	0.15	0.21	0.18	0.41	0.16
		3	0.14	0.12	0.26	0.14	0.13	0.11	0.21	0.17
		5	0.05	0.04	0.08	0.14	0.04	0.03	0.08	0.14

Table 5: Here, we show models trained on CIFAR10 and CIFAR100 and evaluate on the Gaussian noise corrupted data. The NCG accuracy, test accuracy, the test accuracy on the NCG correct examples, the test accuracy on the NCG incorrect examples. Here, we have corruption level 1, 3, and 5 (full table is in Appendix B.7).



(a) test accuracy (b) NCG accuracy

Figure 6: We show the test and NCG accuracies on the model trained on CIFAR10 and evaluated with the Gaussian noise corrupted data. From the figure, we see that as the corruption level increases, the decrease in both NCG and test accuracies are much slower for robust models. In this example, the slope for test accuracy is -0.44 and -0.03 for natural and TRADES(2), respectively; the slope for NCG accuracy is -0.12 and -0.02 for natural and TRADES(2), respectively. For other kinds of corruption, please refer to Appendix B.6.

4.1.2 Robust training slows the decrease of test accuracy with more corruption

In general, with the increase of the corruption level, the NCG and test accuracies drop. However, with robust models, this drop is slower comparing with naturally trained models. For example, in Table 5 and the C10 row, the test accuracy for naturally trained model drops from 0.76 to 0.36 while the test accuracy for TRADES(2) only drops from 0.71 to 0.68. Similar effects are found in NCG accuracy as well as other datasets (C100 and I).

To evaluate this quantitatively, we calculate the slope of the test and NCG accuracies from level 1 to 5 of each corruption type with linear least-squares regression. For example, for the naturally trained model on C10, the slope of the test accuracy is the linear least-squares regression trained on the following 5 points: $((0., 0.76), (0.25, 0.63), (0.5, 0.48), (0.75, 0.41), (1.0, 0.36))$ (0.76, 0.48, and 0.36 corresponds to the test accuarcies of the "C10" row and "natural" column in Table 20). Figure 6 (a) shows the scatter plot and the regression line for test accuracy on CIFAR10 with gaussian blur as the corruption.

We can calculate the slope of this regression line for both the robust and naturally trained models, and then we compare these two slopes. In the pixel space, we find that majority of the slopes for robust models are smaller than the slope for naturally trained models. We perform Welch's t-test (p-value threshold set to 0.05) with the null hypothesis being that the slope of a robust model is less than the slope of a naturally trained model. For CIFAR10 and CIFAR100, 15 and 14 (out of 18) of the corruption types pass this test; for ImgNet100, 11 out of 15 corruption types pass the test However, things in the features space tell a different story. We find that the slopes here do not differ significantly between robust and naturally trained models. We perform Welch's t-test with the null hypothesis being that the slopes of a robust and a naturally trained model are different. We find that for CIFAR10 and CIFAR100, 18 and 17 (out

of 18) are not significant. For ImgNet100, all 15 out of 15 corruption types are not significantly. This result resonates with some earlier observations, where we find that in pixel space, the robust and naturally trained models differ a lot in NCG accuracy, but in feature space, this difference is much smaller. We see similar phenomenon with NCG accuracy (see Appendix B.6).

4.2 Implications

The findings Section 4.1.1 and 4.1.2 show that different forms of robustness, including adversarial robustness, robustness to corruption, and the NCG, are interconnected. Through analyzing the NCG property, we may have a future direction for better understanding the underlying mechanism of the interplay of different robustness. All these three robustness properties are related to some properties that humans possess, and it seems enforcing adversarial robustness increases the robustness of the other two robustness. There are several interesting questions that are yet not answered in this work and are good future directions. What other distance metric does NCG also applies to? Does enforcing NCG or robustness to corruption increase adversarial robustness? Do these three forms of robustness also have a similar connection with other forms of robustness, such as the robustness to background changes Xiao et al. (2020) and sub-population shift (Santurkar et al., 2020)? Does enforcing other human-like behavior on neural networks increase the "humanness" of the model?

Ablation study. In addition to the results presented here, we also repeat the experiments with models trained by other scholars and different architectures. The results can be found in Appendix B.2, and these results also have come to similar conclusions.

5 Related Work

Some prior works have looked at out-of-distribution generalization benefits of adversarially robust neural networks. For transfer learning, Salman et al. (2020) and Utrera et al. (2020) report that when adapting pre-trained models to new domains, using adversarially trained models as the pre-trained models transfer better than naturally trained ones. Shafahi et al. (2019) show that robust models also have better adversarial robustness after transferring to new domains. Dong et al. (2020) and Huang et al. (2021) find that robust language models transfer better to a different language. In other related work, Stutz et al. (2019a) develop confidence calibrated adversarial training to reject examples with low confidence. All these works focus on transferring a robust pre-trained model to completely new datasets and they evaluate test accuracy or adversarial test accuracy. In contrast, we look at understanding a different phenomenon – generalizing to nearby categories from a similar dataset.

Understanding adversarially robust generalization for in-distribution inputs has also been the topic of some study – particularly since most adversarially robust neural networks models suffer from a loss in test accuracy. Rice et al. (2020) show that adversarial training can overfit on in-distribution examples, leading to worse test accuracy. Yang et al. (2020) suggest that the robustness-accuracy tradeoff in neural networks may be due to poor generalization, since common benchmark datasets have well-separated classes. Stutz et al. (2019b) show that robustness on the in-distribution data manifold leads to better generalization on the in-distribution test examples. Our work expands on this thread by showing that robust neural networks resemble the nearest neighbor classifier in their generalization behavior, which may have some connection to their lack of accuracy on (in-distribution) test inputs.

Ford et al. (2019); Kang et al. (2019); Taori et al. (2020) show that robust models often demonstrated improved robustness to data corruptions, and Salman et al. (2020); Utrera et al. (2020) show that robust models transfer better to downstream tasks. However, the underlying mechanism is not yet well understood. The NCG can be seen as a form of robustness as it provides a structure on the network's outputs.

6 Conclusion

We examine out-of-distribution (OOD) properties of neural networks and uncover intriguing generalization properties. We show that neural networks have a tendency of predicting OOD examples with the labels of their closest training examples. We call this property the nearest category generalization (NCG). We also show that robust networks follow NCG more strictly than naturally trained models. Through a thorough empirical investigation, we posit that NCG happens most likely due to the inductive bias of robust networks. Next, we continue to examine whether NCG holds for a set of different kinds of OOD examples, the corrupted data. We not only find that NCG holds for corrupted data, but also observe an interplay between adversarial robustness, robustness to corruption, and NCG. We show that these three seemingly disparate properties are interconnected. A future direction would be to explore this connection in more detail, either through experiments or through a better theoretical understanding of the inductive bias of robust networks. Another direction is to further investigate the relationship between NCG and other generalization-related tasks such as transfer learning or zero-shot learning.

References

Joseph L Austerweil, Shi Xian Liew, Nolan Conaway, and Kenneth J Kurtz. Creating something different: Similarity, contrast, and representativeness in categorization. 2019.

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)*, pages 39–57. IEEE, 2017.

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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Xin Dong, Yaxin Zhu, Yupeng Zhang, Zuohui Fu, Dongkuan Xu, Sen Yang, and Gerard De Melo. Leveraging adversarial training in self-learning for cross-lingual text classification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1541–1544, 2020.

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

Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

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*, pages 770–778, 2016.

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

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*, pages 4700–4708, 2017.

Kuan-Hao Huang, Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. Improving zero-shot cross-lingual transfer learning via robust training. *arXiv preprint arXiv:2104.08645*, 2021.

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

Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.

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

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

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.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

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.

Jinlong Liu, Guoqing Jiang, Yunzhi Bai, Ting Chen, and Huayan Wang. Understanding why neural networks generalize well through gsnr of parameters. *arXiv preprint arXiv:2001.07384*, 2020.

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.

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

Robert M Nosofsky. Attention, similarity, and the identification–categorization relationship. *Journal of experimental psychology: General*, 115(1):39, 1986.

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*, pages 8024–8035. 2019.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

Jeffrey N Rouder and Roger Ratcliff. Comparing categorization models. *Journal of Experimental Psychology: General*, 133(1):63, 2004.

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

Adam N Sanborn, Katherine Heller, Joseph L Austerweil, and Nick Chater. Refresh: A new approach to modeling dimensional biases in perceptual similarity and categorization. *Psychological Review*, 2021.

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

Terrence J Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48):30033–30038, 2020.

Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. *arXiv preprint arXiv:1905.08232*, 2019.

David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. *CoRR, abs/1910.06259*, 2019a.

David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019b.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *arXiv preprint arXiv:2007.00644*, 2020.

Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better. *arXiv preprint arXiv:2007.05869*, 2020.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *ArXiv preprint arXiv:2006.09994*, 2020.

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.

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.

A Detailed experiment setups

The experiments are performed on 6 NVIDIA GeForce RTX 2080 Ti and 2 RTX 3080 GPUs located on three servers. Two of the servers have Intel Core i9 9940X and 128GB of RAM and the other one has AMD Threadripper 3960X and 256GB of RAM. We compute nearest neighbors using FAISS[3] (Johnson et al., 2017), and all neural networks are implemented under the PyTorch framework[4] (Paszke et al., 2019)

Algorithm implementations. For C&W algorithm (Carlini and Wagner, 2017), we use the implementation by For TRADES (Zhang et al., 2019), we also use the implementation From the original author[5].

Datasets. All datasets used in our paper can be found in publicly available urls. MNIST can be found in this url[6], CIFAR10 and CIFAR100 can be found in this url[7], ImgNet can be found in this url[8].

Architechtures. We consider the convolutional neural network (CNN)[9], wider residual network (WRN-40-10) (Zagoruyko and Komodakis, 2016), ResNet50 (He et al., 2016) for our experiments in the pixel space.

Optimizers. We consider stochastic gradient descent (SGD) and Adam (Kingma and Ba, 2014) as the optimizers.

MNIST setup. We use the CNN used by Zhang et al. (2019) for training neural networks in the pixel space. The learning rate is decreased by a factor of 0.1 on the 40-th, 50-th, and 60-th epoch. We use the output of the last convolutional CNN output as the extracted feature.

CIFAR10, CIFAR100, ImgNet100 setup. For CIFAR10 and CIFAR100, we use Wider ResNet (WRN-40-10) (Zagoruyko and Komodakis, 2016) for training neural networks in the pixel space. For ImgNet100, we use ResNet50 (He et al., 2016) for training neural networks in the pixel space. The learning rate is decreased by a factor of 0.1 on the 40-th, 50-th, and 60-th epoch. For ImgNet100, we normalize the data by subtracting the mean (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225).

dataset	MNIST	CIFAR10	CIFAR100	ImgNet100
network structure	CNN	WRN-40-10	WRN-40-10	ResNet50
optimizer	SGD	Adam	Adam	Adam
batch size	128	64	64	128
momentum	0.9	-	-	-
epochs	70	70	70	70
initial learning rate	0.01	0.01	0.01	0.01
# train examples	60000	50000	50000	126689
# test examples	10000	10000	10000	5000
# classes	10	10	20	100

Table 6: Experimental setup for training in the pixel space. No weight decay is applied.

Adversarial attack algorithms. For the adversarial attack algorithms used to find the closest adversarial examples, we use a mixture of projected gradient descent (PGD) (Madry et al., 2017), Brendel Bethge attack (Brendel et al., 2019), boundary attack (Brendel et al., 2017), multi-targeted attack (Kwon et al., 2018), Sign-Opt (Cheng et al., 2019) and C&W algorithm (Carlini and Wagner, 2017).

A.0.1 Setups for experiments in the feature space

Architechtures. For MNIST, CIFAR10, and CIFAR100, we train a multi-layer-perceptron (MLP) with two hidden layers each with 256 neurons and ReLU as the activation function in the feature space. For ImgNet100, we

[3] code and license can be found in https://github.com/facebookresearch/faiss
[4] code and license can be found in https://github.com/pytorch/pytorch
[5] code and license can be found in https://github.com/yaodongyu/TRADES
[6] http://yann.lecun.com/exdb/mnist/
[7] https://www.cs.toronto.edu/~kriz/cifar.html
[8] https://www.image-net.org/
[9] CNN is retrieved from the public repository of TRADES (Zhang et al., 2019) https://github.com/yaodongyu/TRADES/blob/master/models/small_cnn.py

train an MLP with two hidden layers each with 1024 neurons and ReLU as the activation function in the feature space. For all four datasets, we use SGD as the optimizer with an initial learning rate of 0.01 and a momentum of 0.9.

B Additional experiment results

B.1 NCG accuracies.

Table 7 shows the test accuracies of the 1-NN classifiers in the feature space of 12 different datasets. Table 8, 9, 10, and 11 extends Table 2 with all datasets. We see that the 1-NN classifiers are actually performing very well (close) in the feature space.

M-0	M-4	M-9	C10-0	C10-4	C10-9	C100-0	C100-4	C100-9	I-0	I-1	I-2
0.99	0.99	0.99	0.89	0.88	0.88	0.73	0.73	0.71	0.14	0.14	0.14

Table 7: The test accuracy of a 1-nearest neighbor classifier in the feature space 12 different datasets.

		natural	AT(2)	TRADES(2)	TRADES(4)	TRADES(8)
MNIST-wo0	train acc.	1.000	0.993	0.987	0.954	0.997
	test acc.	0.995	0.990	0.985	0.956	0.995
	NCG acc.	0.390	0.457	0.457	0.485	0.402
MNIST-wo1	train acc.	1.000	0.994	0.987	0.975	0.997
	test acc.	0.995	0.991	0.987	0.974	0.994
	NCG acc.	0.273	0.451	0.355	0.528	0.259
MNIST-wo2	train acc.	1.000	0.993	0.988	0.958	0.997
	test acc.	0.994	0.990	0.987	0.962	0.994
	NCG acc.	0.402	0.532	0.529	0.520	0.452
MNIST-wo3	train acc.	1.000	0.994	0.989	0.962	0.997
	test acc.	0.995	0.992	0.988	0.964	0.994
	NCG acc.	0.564	0.659	0.667	0.592	0.538
MNIST-wo4	train acc.	1.000	0.994	0.988	0.963	0.997
	test acc.	0.995	0.991	0.987	0.966	0.995
	NCG acc.	0.760	0.766	0.810	0.758	0.749
MNIST-wo5	train acc.	1.000	0.993	0.988	0.965	0.997
	test acc.	0.995	0.990	0.987	0.965	0.995
	NCG acc.	0.505	0.611	0.618	0.616	0.537
MNIST-wo6	train acc.	1.000	0.993	0.987	0.959	0.997
	test acc.	0.995	0.991	0.987	0.962	0.995
	NCG acc.	0.515	0.551	0.556	0.505	0.538
MNIST-wo7	train acc.	1.000	0.994	0.989	0.962	0.997
	test acc.	0.995	0.992	0.990	0.967	0.994
	NCG acc.	0.507	0.672	0.703	0.713	0.594
MNIST-wo8	train acc.	1.000	0.993	0.987	0.966	0.997
	test acc.	0.994	0.990	0.987	0.966	0.995
	NCG acc.	0.416	0.493	0.497	0.491	0.446
MNIST-wo9	train acc.	1.000	0.996	0.992	0.962	0.997
	test acc.	0.996	0.994	0.992	0.964	0.995
	NCG acc.	0.577	0.714	0.691	0.703	0.660

Table 8: The train, test, and NCG accuracies of 10 MNIST datasets and 5 training methods in the pixel space.

		natural	AT(2)	TRADES(2)	TRADES(4)	TRADES(8)
CIFAR10-wo0	train acc.	1.000	0.999	0.992	0.870	0.878
	test acc.	0.898	0.729	0.716	0.660	0.761
	NCG acc.	0.355	0.494	0.492	0.520	0.483
CIFAR10-wo4	train acc.	1.000	1.000	0.990	0.874	0.508
	test acc.	0.886	0.754	0.742	0.700	0.485
	NCG acc.	0.222	0.361	0.333	0.331	0.289
CIFAR10-wo9	train acc.	1.000	1.000	0.992	0.948	0.778
	test acc.	0.885	0.725	0.712	0.732	0.641
	NCG acc.	0.145	0.212	0.192	0.247	0.245
CIFAR100-wo0	train acc.	1.000	0.998	0.995	0.943	0.902
	test acc.	0.741	0.554	0.547	0.576	0.607
	NCG acc.	0.175	0.240	0.252	0.252	0.206
CIFAR100-wo4	train acc.	1.000	0.998	0.995	0.857	0.859
	test acc.	0.743	0.544	0.543	0.492	0.553
	NCG acc.	0.137	0.192	0.191	0.187	0.185
CIFAR100-wo9	train acc.	1.000	0.996	0.995	0.950	0.527
	test acc.	0.727	0.547	0.537	0.585	0.431
	NCG acc.	0.222	0.353	0.412	0.427	0.465
ImgNet100-wo0	train acc.	1.000	0.999	0.994	0.983	0.704
	test acc.	0.529	0.417	0.393	0.354	0.320
	NCG acc.	0.033	0.044	0.041	0.054	0.067
ImgNet100-wo1	train acc.	1.000	0.999	0.995	0.972	0.783
	test acc.	0.534	0.414	0.385	0.356	0.316
	NCG acc.	0.047	0.049	0.051	0.061	0.072
ImgNet100-wo2	train acc.	1.000	0.999	0.994	0.971	0.695
	test acc.	0.537	0.394	0.388	0.353	0.320
	NCG acc.	0.027	0.028	0.033	0.044	0.049

Table 9: The train, test, and NCG accuracies of 9 different variations of CIFAR10, CIFAR100, and ImgNet100 datasets and 5 training methods in the pixel space.

B.2 Ablation study

B.2.1 Different architectures

We repeat the experiment with a different network architecture – DenseNet161 (Huang et al., 2017). Their training, testing, and NCG accuracies are shown in Table 12.

B.2.2 Pretrainde models

To verify that our observations can also be observed by models trained by others, we downloaded pretrained models from `https://github.com/MadryLab/robustness/tree/master/robustness` by Engstrom et al. (2019). Table 13 shows the training and testing accuarcies of their models.

Corrupted data. For models in the features space, we follow the same setup as in the feature space of CIFAR10, which still trains a multi-layer perceptron on the CNN feature space, but in the feature space of the pretrained model. Table 14 shows comparison of robust and naturally trained models. From the table, we can see that the robust models in general have higher NCG accuracy than the naturally trained models when the robust radius r is larger than 0.25. Table 15 shows the test accuracy, NCG accuracy and the test accuracy conditioned on whether the example is considered correct under NCG accuracy (NCG correct or not).

		natural	AT(2)	TRADES(2)	TRADES(4)	TRADES(8)
MNIST-wo0	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	0.99	0.99	0.99	0.99
	NCG acc.	0.28	0.32	0.39	0.49	0.55
MNIST-wo1	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	0.99	0.99	0.99	0.99
	NCG acc.	0.14	0.21	0.27	0.50	0.51
MNIST-wo2	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	0.99	0.99	0.99	1.00
	NCG acc.	0.41	0.46	0.53	0.59	0.62
MNIST-wo3	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	0.99	0.99	1.00	0.99
	NCG acc.	0.68	0.71	0.73	0.73	0.74
MNIST-wo4	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	0.99	0.99	1.00	1.00
	NCG acc.	0.78	0.73	0.77	0.81	0.86
MNIST-wo5	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	0.99	1.00	1.00	0.99
	NCG acc.	0.61	0.63	0.65	0.68	0.69
MNIST-wo6	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	1.00	1.00	1.00	1.00
	NCG acc.	0.54	0.58	0.60	0.65	0.66
MNIST-wo7	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	0.99	1.00	1.00	1.00
	NCG acc.	0.53	0.54	0.61	0.68	0.67
MNIST-wo8	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	0.99	0.99	1.00	0.99
	NCG acc.	0.46	0.47	0.51	0.56	0.59
MNIST-wo9	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.99	1.00	1.00	1.00	1.00
	NCG acc.	0.61	0.71	0.71	0.73	0.79

Table 10: The train, test, and NCG accuracies of 10 MNIST datasets and 5 training methods in the feature space.

B.3 Histograms of the empirical robust radius and OOD distance

Here we present the histogram of the empirical robust radius and OOD distance for other algorithms and datasets. Figure 7, 8, and 9 show the results for MNIST, CIFAR10, CIFAR100, and ImgNet100 in the pixel space. Figure 10, 11, and 12 show the results for MNIST, CIFAR10, CIFAR100, and ImgNet100 in the feature space.

For the MNIST histograms in the feature space, we see that the empirical robust radius have smaller number comparing with the counts for OOD distance. That is because there are many OOD examples that have these few training examples as the closest training example.

Table 16, 17, 18, and 19 show the average empirical robust radius, average OOD distance, portion of OOD examples covered by the robust norm ball of its closest training example and the NCG accuracy (in the pixel and feature space of M, C10, C100, and I). From the table, we can see that the portion of OOD examples covered by the robust norm ball of its closest training example are very low in general, regardless of the NCG accuracy. This rejects that the hypothesis of 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 distances between the OOD inputs and their closest training examples.

		natural	AT(.5)/(1)	TRADES(2)	TRADES(4)	TRADES(8)
CIFAR10-wo0	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.89	0.89	0.89	0.90	0.90
	NCG acc.	0.80	0.83	0.81	0.83	0.83
CIFAR10-wo4	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.88	0.88	0.88	0.89	0.88
	NCG acc.	0.82	0.84	0.82	0.85	0.85
CIFAR10-wo9	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.88	0.88	0.88	0.89	0.89
	NCG acc.	0.84	0.89	0.83	0.88	0.87
CIFAR100-wo0	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.72	0.73	0.73	0.74	0.74
	NCG acc.	0.63	0.70	0.69	0.68	0.68
CIFAR100-wo4	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.72	0.73	0.73	0.74	0.74
	NCG acc.	0.69	0.74	0.75	0.73	0.74
CIFAR100-wo9	train acc.	1.00	1.00	1.00	1.00	1.00
	test acc.	0.70	0.72	0.72	0.73	0.73
	NCG acc.	0.66	0.74	0.72	0.71	0.71
ImgNet100-wo0	train acc.	0.99	0.57	0.33	0.98	0.98
	test acc.	0.22	0.25	0.26	0.26	0.26
	NCG acc.	0.11	0.16	0.15	0.12	0.13
ImgNet100-wo1	train acc.	1.00	0.56	0.32	0.98	0.98
	test acc.	0.22	0.24	0.27	0.26	0.25
	NCG acc.	0.13	0.15	0.18	0.14	0.15
ImgNet100-wo2	train acc.	1.00	0.60	0.33	0.98	0.98
	test acc.	0.22	0.25	0.26	0.26	0.26
	NCG acc.	0.11	0.15	0.15	0.14	0.14

Table 11: The train, test, and NCG accuracies of 9 different variations of CIFAR10, CIFAR100, and ImgNet100 datasets and 5 training methods in the feature space. We use different radius for AT since not all converge well when the radius is large ($r = 2$) For CIFAR10 and CIFAR100, we use AT(1); for ImgNet100, we use AT(.5).

		natural	AT(2)	TRADES(2)
CIFAR10-wo0	train acc.	1.000	0.781	0.876
	test acc.	0.839	0.637	0.640
	NCG acc.	0.342	0.487	0.521
CIFAR100-wo0	train acc.	1.000	0.886	0.557
	test acc.	0.608	0.500	0.441
	NCG acc.	0.173	0.225	0.271

Table 12: Results with DenseNet161 on CIFAR10 and CIFAR100.

B.4 Additional figures on NCG accuracy and the distance to the closest training example

Figure 13 and 14 shows the NCG accuracy and the distance to the closest training example for MNIST, CIFAR10, and CIFAR100 in both pixel and feature space. We can see that, in general, the NCG accuracy is higher when in- and out-of-distribution examples are closer to each other.

B.5 Additional results for corrupted data

Robust models on corrupted data. On average (over the 90 and 75 corrupted sets), robust models have a 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. In the feature space, we still find that **all** the 255 corruption

	natural	AT(.25)	AT(.5)	AT(1.0)
trn acc.	1.00	0.97	0.98	0.86
tst acc.	0.95	0.93	0.91	0.82

Table 13: The training and testing accuracies of the Engstrom et al. (2019)'s pretrained models on CIFAR10.

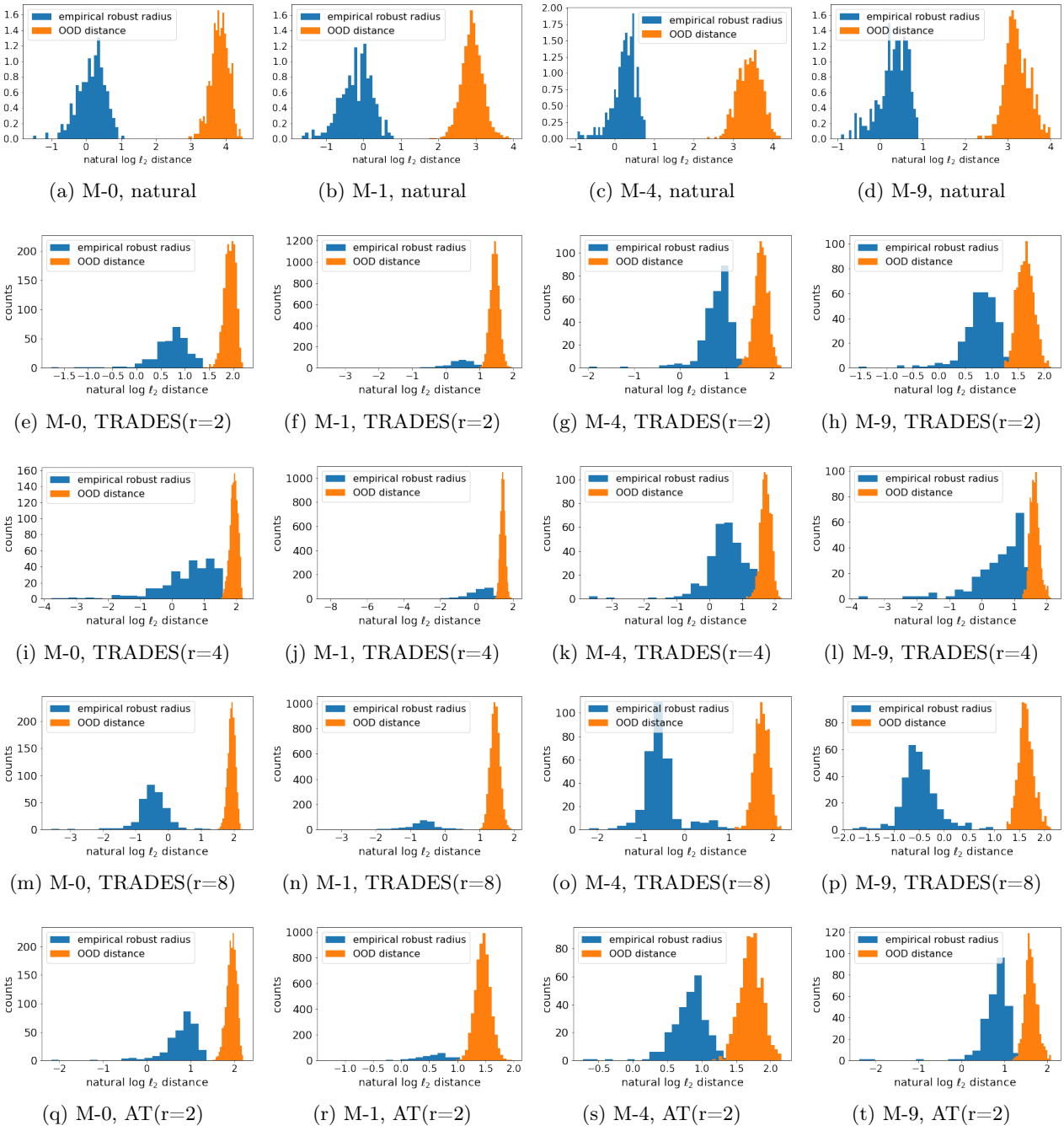
		robust > natural counts	difference	ratio
		pixel		
CIFAR10	AT(0.25)	51/90	0.00 ± 0.05	1.14 ± 0.04
	AT(0.5)	86/90	0.14 ± 0.10	3.27 ± 0.55
	AT(1.0)	88/90	0.18 ± 0.06	3.09 ± 0.22
		feature		
CIFAR10	AT(1.0)	70/90	0.00 ± 0.00	1.01 ± 0.00
	TRADES(2)	55/90	0.00 ± 0.00	1.00 ± 0.00
	TRADES(4)	52/90	0.00 ± 0.00	1.00 ± 0.00
	TRADES(8)	55/90	0.00 ± 0.00	1.00 ± 0.00

Table 14: In both pixel and feature space, among the 90 corrupted sets for cifar10, the first columns shows the number of robust models that have an NCG accuracy higher than naturally trained network. The second and third column shows the average difference and ratio of the NCG accuracy of the robust models and naturally trained networks (average over the NCG accuracies on the 90 corrupted sets).

	model	natural				AT(1)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
				pixel					
C10	1	0.52	0.50	0.68	0.13	0.21	0.17	0.31	0.30
	2	0.37	0.35	0.49	0.12	0.20	0.16	0.31	0.29
	3	0.28	0.26	0.38	0.13	0.20	0.16	0.30	0.29
	4	0.25	0.24	0.33	0.14	0.20	0.16	0.30	0.28
	5	0.23	0.21	0.32	0.14	0.20	0.16	0.30	0.28
				feature					
C10	1	0.82	0.44	0.85	0.95	0.82	0.42	0.83	0.97
	2	0.67	0.38	0.70	0.91	0.66	0.41	0.68	0.95
	3	0.49	0.31	0.52	0.86	0.48	0.30	0.49	0.93
	4	0.42	0.29	0.44	0.85	0.41	0.31	0.42	0.92
	5	0.36	0.27	0.37	0.84	0.35	0.27	0.35	0.92

Table 15: The test accuracy, NCG accuracy, and the test accuracy conditioned on the NCG correctness of Engstrom et al. (2019)'s pretrained model on the Gaussian noise corrupted data.

sets have an NCG accuracy above chance level, but the NCG accuracies of the robust models are closer to the naturally trained models. For CIFAR100, we still observe that **all** robust models have an NCG accuracy higher than the naturally trained models. But for CIFAR10, we find that on only 42 (out of 90) corrupted sets, TRADES(2) models have a higher NCG accuracy than naturally trained models. The average improvement over the naturally trained models in NCG accuracy goes down to $1.00 \pm .00$, $1.07 \pm .00$, and $1.09 \pm .01$ times for CIFAR10, CIFAR100, and ImgNet100 respectively.



(a) M-0, natural (b) M-1, natural (c) M-4, natural (d) M-9, natural

(e) M-0, TRADES(r=2) (f) M-1, TRADES(r=2) (g) M-4, TRADES(r=2) (h) M-9, TRADES(r=2)

(i) M-0, TRADES(r=4) (j) M-1, TRADES(r=4) (k) M-4, TRADES(r=4) (l) M-9, TRADES(r=4)

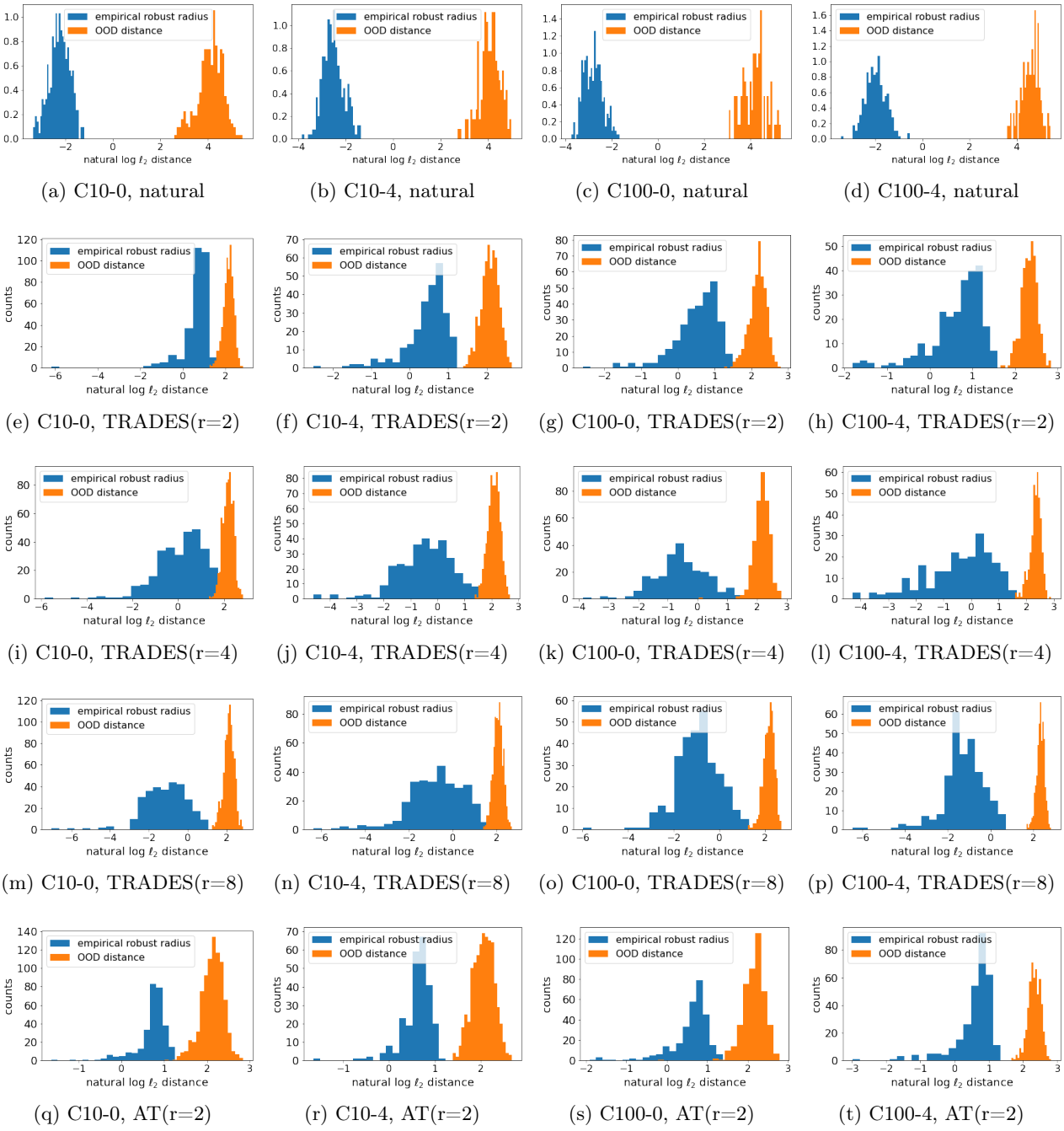
(m) M-0, TRADES(r=8) (n) M-1, TRADES(r=8) (o) M-4, TRADES(r=8) (p) M-9, TRADES(r=8)

(q) M-0, AT(r=2) (r) M-1, AT(r=2) (s) M-4, AT(r=2) (t) M-9, AT(r=2)

Figure 7: The histograms of the empirical robust radius and OOD distance for networks trained on MNIST-wo0 (M-0), MNIST-wo1 (M-1), MNIST-wo4 (M-4), and MNIST-wo9 (M-9) in the pixel space.

B.6 Additional results on the slope of corrupted test accuracy

NCG accuracy. Repeating the same experiment with NCG accuracy, we find similar results as well. In the pixel space, for CIFAR10 and CIFAR100, the slope of naturally trained models are significantly smaller than TRADES(2) on 15 and 14 (out of 18) corruption types. For ImgNet100, 6 out of 15 corruption types pass the test. The other 9 corruption types are not significant (they did not accept or reject the hypothesis). In the feature space, we also test whether the slopes of robust and naturally trained models are different. For CIFAR10 and CIFAR100, 17 and 15 (out of 18) corruption types, respectively, are not significantly different. For ImgNet100, 13



(a) C10-0, natural (b) C10-4, natural (c) C100-0, natural (d) C100-4, natural

(e) C10-0, TRADES(r=2) (f) C10-4, TRADES(r=2) (g) C100-0, TRADES(r=2) (h) C100-4, TRADES(r=2)

(i) C10-0, TRADES(r=4) (j) C10-4, TRADES(r=4) (k) C100-0, TRADES(r=4) (l) C100-4, TRADES(r=4)

(m) C10-0, TRADES(r=8) (n) C10-4, TRADES(r=8) (o) C100-0, TRADES(r=8) (p) C100-4, TRADES(r=8)

(q) C10-0, AT(r=2) (r) C10-4, AT(r=2) (s) C100-0, AT(r=2) (t) C100-4, AT(r=2)

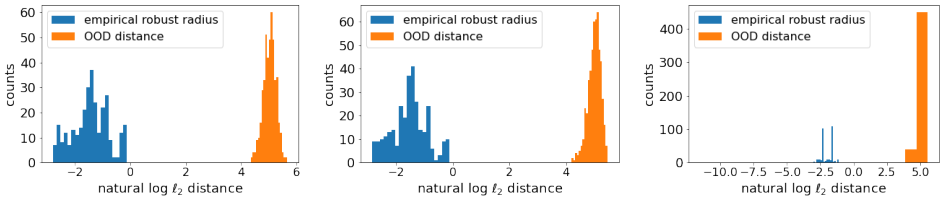
Figure 8: The histograms of the empirical robust radius and OOD distance for networks trained on CIFAR10-wo0 (C10-0), CIFAR10-wo4 (C10-4), CIFAR100-wo0 (C100-4), and CIFAR100-wo4 (C100-4) in the pixel space.

out of 15 corruption types are not significant. Figure 15 and 16, shows the slope of the corrupted test accuracy for CIFAR10 and CIFAR100. Figure 17 and 18. shows the slope of the corrupted test accuracy for CIFAR10 and CIFAR100.

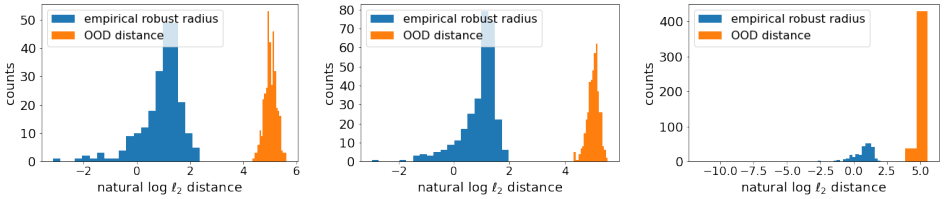
B.7 Full table of Table 5

Table 20 shows the full version of Table 5. We can derive the same conclusion from this table.

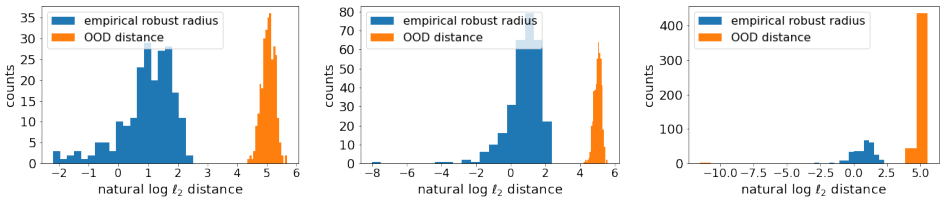
We also show the tables of other corruption type from Table 22 to Table 87 in both pixel and feature space.



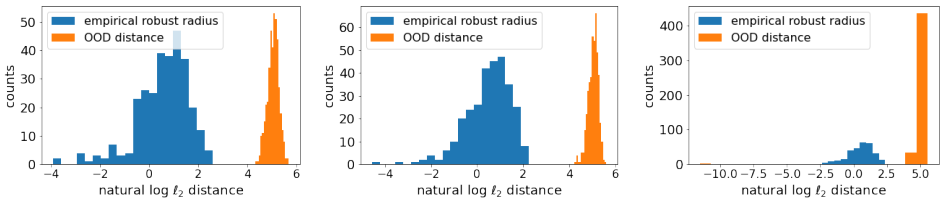
(a) I-0, natural (b) I-1, natural (c) I-2, natural



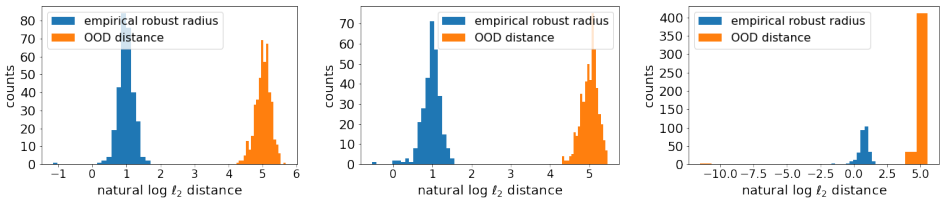
(d) I-0, TRADES(r=2) (e) I-1, TRADES(r=2) (f) I-2, TRADES(r=2)



(g) I-0, TRADES(r=4) (h) I-1, TRADES(r=4) (i) I-2, TRADES(r=4)



(j) I-0, TRADES(r=8) (k) I-1, TRADES(r=8) (l) I-2, TRADES(r=8)

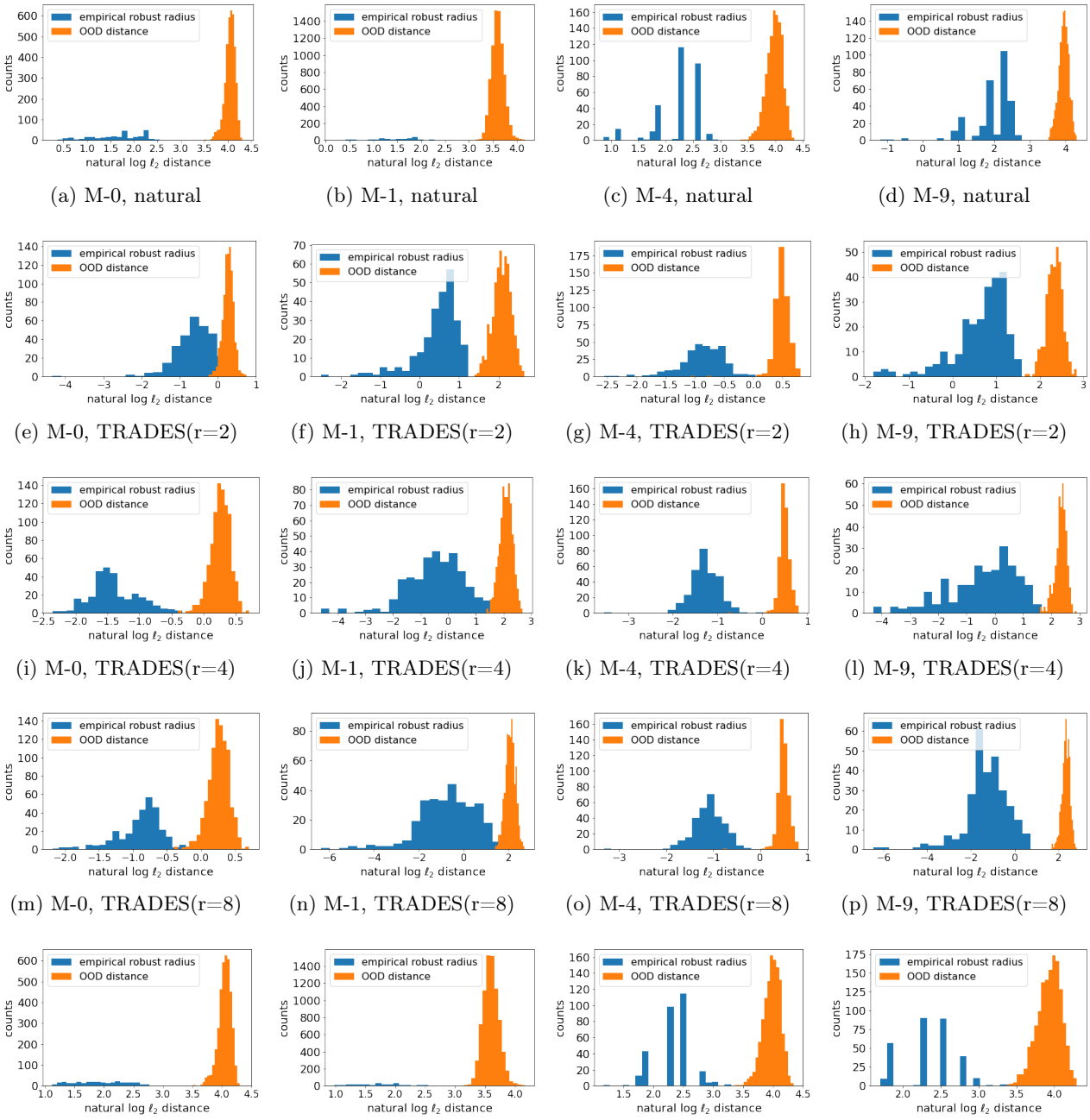


(m) I-0, AT(r=2) (n) I-1, AT(r=2) (o) I-2, AT(r=2)

Figure 9: The histograms of the empirical robust radius and OOD distance for networks trained on ImgNet100-wo0(I-0), ImgNet100-wo0(I-1), and ImgNet100-wo0(I-2) in the pixel space.

B.8 Most predicted classes

In Table 21, we first remove a class from the training set of each dataset and train a neural network on the modified trianing set. We then look at the predictions of the neural network on these removed images and record their top two most predicted classes. From the result, we see that the outputs that these two networks produce follow some patterns no matter it is in the pixel space or the feature space, robust training or not For example, for C10-0 and C100-0, we see that "airplanes" are predicted as a ship or bird possibly because they have similar background of the sky. "aquatic mammals" are predicted as fish possibly because they are both in the water.



(a) M-0, natural (b) M-1, natural (c) M-4, natural (d) M-9, natural

(e) M-0, TRADES(r=2) (f) M-1, TRADES(r=2) (g) M-4, TRADES(r=2) (h) M-9, TRADES(r=2)

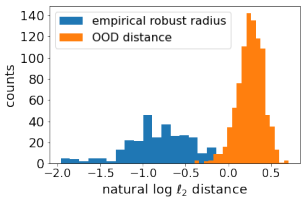
(i) M-0, TRADES(r=4) (j) M-1, TRADES(r=4) (k) M-4, TRADES(r=4) (l) M-9, TRADES(r=4)

(m) M-0, TRADES(r=8) (n) M-1, TRADES(r=8) (o) M-4, TRADES(r=8) (p) M-9, TRADES(r=8)

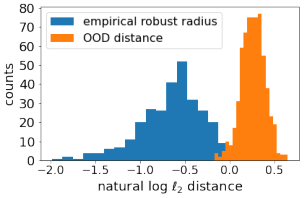
(q) M-0, AT(r=2) (r) M-1, AT(r=2) (s) M-4, AT(r=2) (t) M-9, AT(r=2)

Figure 10: The histograms of the empirical robust radius and OOD distance for networks trained on MNIST-wo0 (M-0), MNIST-wo1 (M-1), MNIST-wo4 (C100-4), and MNIST-wo9 (C100-9) in the feature space.

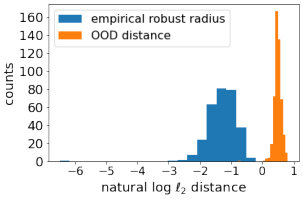
	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30



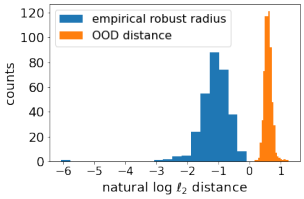
(a) C10-0, natural



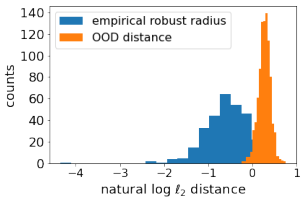
(b) C10-4, natural



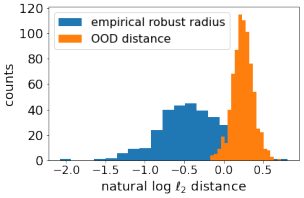
(c) C100-0, natural



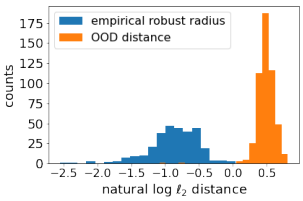
(d) C100-4, natural



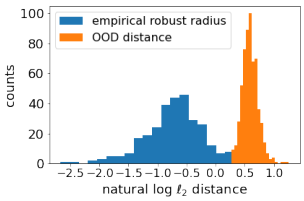
(e) C10-0, TRADES(r=2)



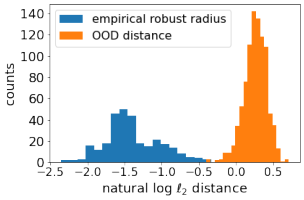
(f) C10-4, TRADES(r=2)



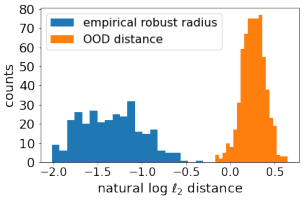
(g) C100-0, TRADES(r=2)



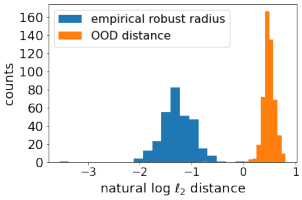
(h) C100-4, TRADES(r=2)



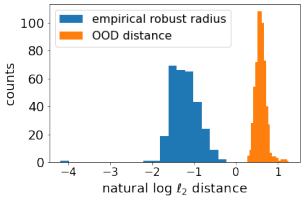
(i) C10-0, TRADES(r=4)



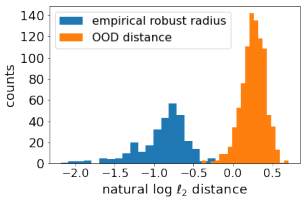
(j) C10-4, TRADES(r=4)



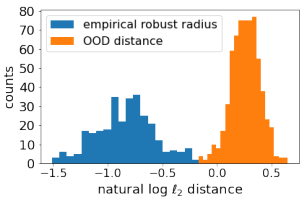
(k) C100-0, TRADES(r=4)



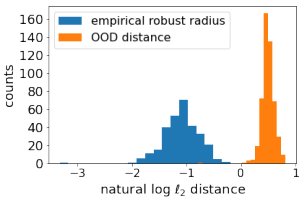
(l) C100-4, TRADES(r=4)



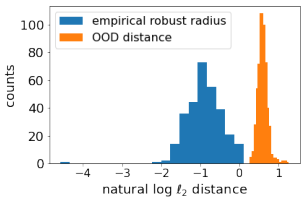
(m) C10-0, TRADES(r=8)



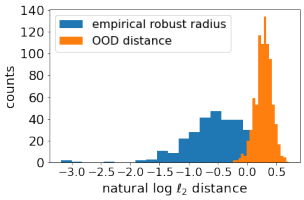
(n) C10-4, TRADES(r=8)



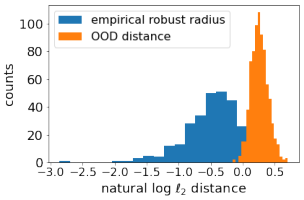
(o) C100-0, TRADES(r=8)



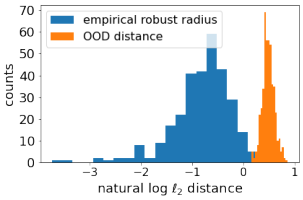
(p) C100-4, TRADES(r=8)



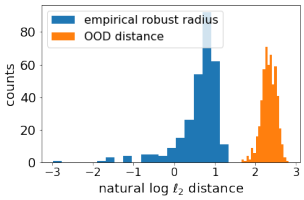
(q) C10-0, AT(r=1)



(r) C10-4, AT(r=1)

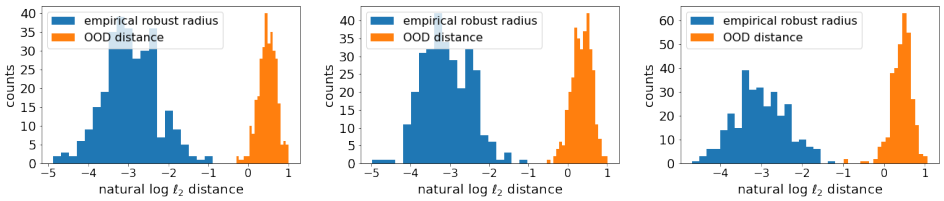


(s) C100-0, AT(r=1)

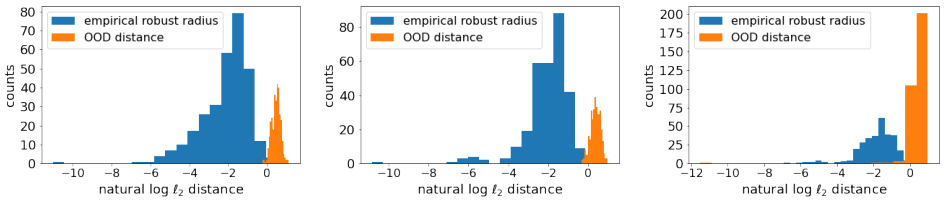


(t) C100-4, AT(r=1)

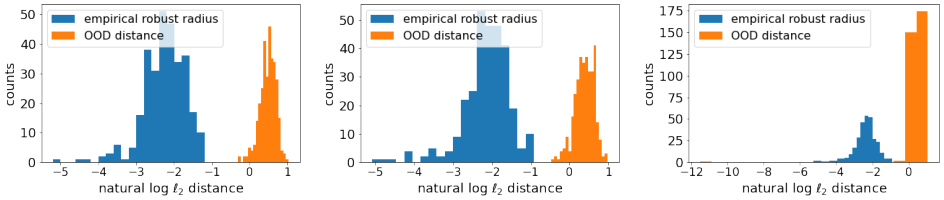
Figure 11: The histograms of the empirical robust radius and OOD distance for networks trained on CIFAR10-wo0 (C10-0), CIFAR10-wo4 (C10-4), CIFAR100-wo0 (C100-4), and CIFAR100-wo4 (C100-4) in the feature space.



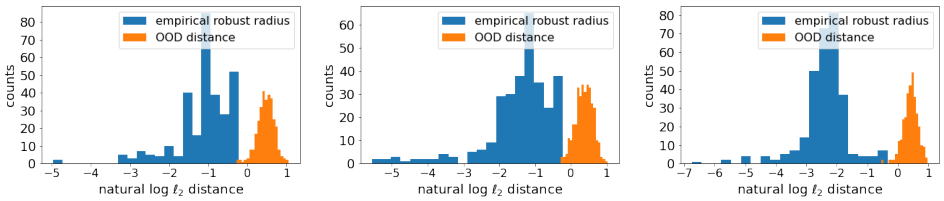
(a) I-0, natural (b) I-1, natural (c) I-2, natural



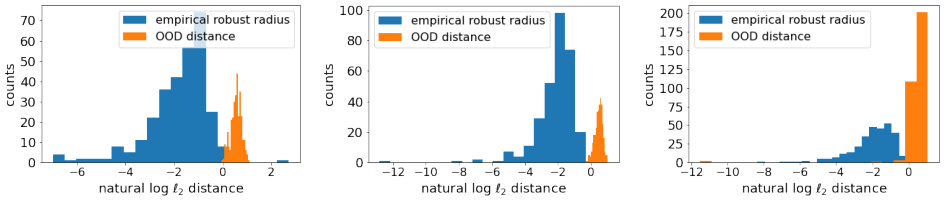
(d) I-0, TRADES(r=2) (e) I-1, TRADES(r=2) (f) I-2, TRADES(r=2)



(g) I-0, TRADES(r=4) (h) I-1, TRADES(r=4) (i) I-2, TRADES(r=4)



(j) I-0, TRADES(r=8) (k) I-1, TRADES(r=8) (l) I-2, TRADES(r=8)



(m) I-0, AT(r=.5) (n) I-1, AT(r=.5) (o) I-2, AT(r=.5)

Figure 12: The histograms of the empirical robust radius and OOD distance for networks trained on ImgNet100-wo0(I-0), ImgNet100-wo0(I-1), and ImgNet100-wo0(I-2) in the feature space.

		empirical robust radius	OOD dist.	portion covered	NCG acc.
M-0	AT(2)	2.33	7.00	0.00	0.46
	TRADES(2)	2.17	6.98	0.00	0.46
	TRADES(4)	2.07	6.94	0.00	0.48
	TRADES(8)	0.68	6.97	0.00	0.42
	natural	1.26	6.95	0.00	0.36
M-1	AT(2)	1.83	4.30	0.00	0.51
	TRADES(2)	1.57	4.31	0.00	0.35
	TRADES(4)	1.44	4.31	0.00	0.54
	TRADES(8)	0.56	4.29	0.00	0.28
	natural	0.89	4.33	0.00	0.23
M-2	AT(2)	2.27	6.92	0.00	0.54
	TRADES(2)	2.15	6.93	0.00	0.53
	TRADES(4)	2.23	6.86	0.00	0.52
	TRADES(8)	0.71	7.03	0.00	0.47
	natural	1.33	6.98	0.00	0.40
M-3	AT(2)	2.35	6.27	0.00	0.66
	TRADES(2)	2.22	6.33	0.00	0.66
	TRADES(4)	1.93	6.30	0.00	0.58
	TRADES(8)	0.70	6.20	0.01	0.55
	natural	1.52	6.34	0.00	0.59
M-4	AT(2)	2.34	5.64	0.00	0.74
	TRADES(2)	2.32	5.89	0.00	0.81
	TRADES(4)	1.84	5.65	0.00	0.76
	TRADES(8)	0.66	5.83	0.00	0.75
	natural	1.33	5.73	0.00	0.76
M-5	AT(2)	2.39	6.30	0.00	0.61
	TRADES(2)	2.17	6.34	0.00	0.62
	TRADES(4)	2.24	6.37	0.00	0.62
	TRADES(8)	0.66	6.33	0.00	0.55
	natural	1.39	6.34	0.00	0.53
M-6	AT(2)	2.41	6.46	0.00	0.55
	TRADES(2)	2.18	6.45	0.00	0.56
	TRADES(4)	2.06	6.52	0.00	0.49
	TRADES(8)	0.58	6.46	0.00	0.53
	natural	1.30	6.44	0.00	0.51
M-7	AT(2)	2.18	5.55	0.00	0.67
	TRADES(2)	2.10	5.47	0.00	0.72
	TRADES(4)	1.88	5.51	0.01	0.72
	TRADES(8)	0.77	5.53	0.01	0.59
	natural	1.27	5.51	0.00	0.53
M-8	AT(2)	2.22	6.32	0.00	0.49
	TRADES(2)	1.99	6.30	0.00	0.51
	TRADES(4)	1.92	6.35	0.00	0.47
	TRADES(8)	0.63	6.30	0.00	0.45
	natural	1.35	6.30	0.00	0.42
M-9	AT(2)	2.33	5.14	0.00	0.71
	TRADES(2)	2.28	5.25	0.00	0.68
	TRADES(4)	2.18	5.13	0.00	0.70
	TRADES(8)	0.66	5.20	0.00	0.65
	natural	1.45	5.08	0.00	0.58

Table 16: The average empirical robust radius, average OOD distance, percentage of OOD examples covered by the robust norm ball of its closest training example and the NCG accuracy (in the pixel space of MNIST datasets).

		empirical robust radius	OOD dist.	portion covered	NCG acc.
C10-0	AT(2)	2.14	8.67	0.00	0.49
	TRADES(2)	2.17	8.89	0.00	0.49
	TRADES(4)	1.62	9.13	0.00	0.52
	TRADES(8)	0.51	8.67	0.00	0.48
	natural	0.09	8.71	0.00	0.35
C10-4	AT(2)	1.92	8.04	0.00	0.36
	TRADES(2)	1.75	8.29	0.00	0.33
	TRADES(4)	0.93	8.30	0.00	0.33
	TRADES(8)	0.87	8.16	0.00	0.29
	natural	0.09	8.29	0.00	0.22
C10-9	AT(2)	2.11	10.80	0.00	0.21
	TRADES(2)	2.01	11.05	0.00	0.19
	TRADES(4)	0.92	10.83	0.00	0.25
	TRADES(8)	0.65	10.83	0.00	0.25
	natural	0.12	10.85	0.00	0.14
C100-0	AT(2)	1.96	8.89	0.00	0.24
	TRADES(2)	1.93	9.03	0.00	0.25
	TRADES(4)	0.82	9.01	0.00	0.25
	TRADES(8)	0.59	9.10	0.00	0.21
	natural	0.11	8.94	0.00	0.17
C100-4	AT(2)	2.01	10.43	0.00	0.19
	TRADES(2)	2.21	10.53	0.00	0.19
	TRADES(4)	1.12	10.57	0.00	0.19
	TRADES(8)	0.41	10.36	0.00	0.18
	natural	0.10	10.17	0.00	0.14
C100-9	AT(2)	2.10	9.18	0.00	0.35
	TRADES(2)	2.18	9.57	0.00	0.41
	TRADES(4)	0.86	9.27	0.00	0.43
	TRADES(8)	1.49	9.01	0.00	0.47
	natural	0.08	9.22	0.00	0.22
I-0	AT(2)	2.77	155.07	0.00	0.04
	TRADES(2)	3.14	155.62	0.00	0.04
	TRADES(4)	3.54	160.05	0.00	0.06
	TRADES(8)	2.67	161.10	0.00	0.07
	natural	0.29	157.42	0.00	0.03
I-1	AT(2)	2.72	153.03	0.00	0.05
	TRADES(2)	2.85	155.91	0.00	0.05
	TRADES(4)	2.92	156.97	0.00	0.06
	TRADES(8)	2.32	157.91	0.00	0.07
	natural	0.26	152.47	0.00	0.05
I-2	AT(2)	2.22	154.28	0.00	0.03
	TRADES(2)	2.49	155.76	0.00	0.03
	TRADES(4)	2.67	156.83	0.00	0.04
	TRADES(8)	2.27	156.27	0.00	0.05
	natural	0.15	154.16	0.00	0.03

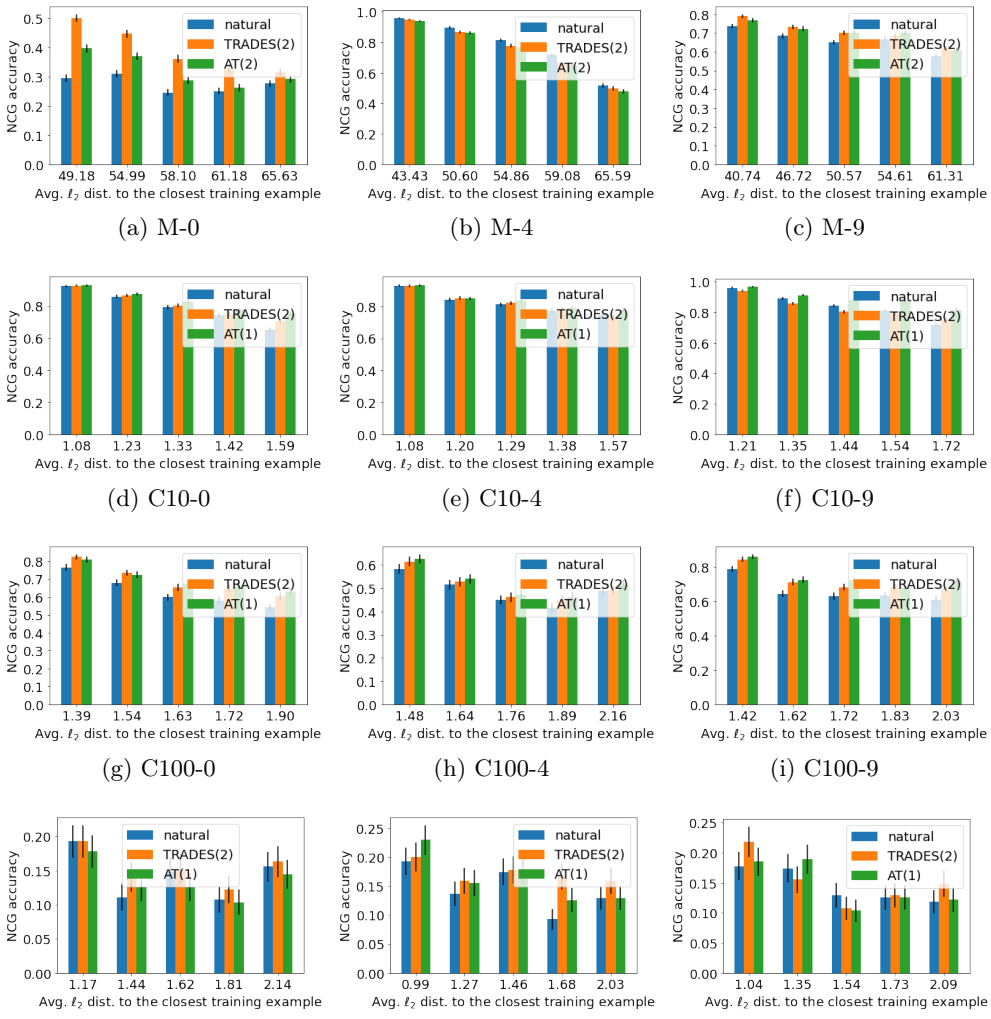
Table 17: The average empirical robust radius, average OOD distance, percentage of OOD examples covered by the robust norm ball of its closest training example and the NCG accuracy (in the pixel space of C10, C100, and I).

		empirical robust radius	OOD dist.	portion covered	NCG acc.
M-0	AT(2)	7.50	57.78	0.00	0.32
	TRADES(2)	9.08	57.16	0.00	0.39
	TRADES(4)	12.54	57.28	0.00	0.49
	TRADES(8)	15.91	57.50	0.00	0.55
	natural	5.84	57.78	0.00	0.28
M-1	AT(2)	6.12	37.06	0.00	0.20
	TRADES(2)	7.69	37.02	0.00	0.26
	TRADES(4)	10.51	37.01	0.00	0.51
	TRADES(8)	13.68	36.87	0.00	0.50
	natural	4.80	37.06	0.00	0.13
M-2	AT(2)	13.13	64.51	0.00	0.46
	TRADES(2)	10.77	63.96	0.00	0.53
	TRADES(4)	14.25	62.53	0.00	0.59
	TRADES(8)	17.60	64.77	0.00	0.62
	natural	7.25	62.41	0.00	0.41
M-3	AT(2)	15.55	70.11	0.00	0.71
	TRADES(2)	13.34	70.09	0.00	0.73
	TRADES(4)	17.64	70.68	0.00	0.73
	TRADES(8)	21.42	69.82	0.00	0.74
	natural	9.33	70.11	0.00	0.68
M-4	AT(2)	10.97	54.27	0.00	0.73
	TRADES(2)	13.43	53.89	0.00	0.77
	TRADES(4)	16.79	54.23	0.00	0.81
	TRADES(8)	20.62	53.80	0.00	0.86
	natural	9.74	54.27	0.00	0.78
M-5	AT(2)	14.92	65.51	0.00	0.63
	TRADES(2)	12.25	65.25	0.00	0.65
	TRADES(4)	15.50	64.37	0.00	0.68
	TRADES(8)	19.64	65.12	0.00	0.69
	natural	9.47	65.51	0.00	0.61
M-6	AT(2)	11.66	60.66	0.00	0.58
	TRADES(2)	10.62	60.65	0.00	0.60
	TRADES(4)	14.15	60.67	0.00	0.65
	TRADES(8)	17.44	60.28	0.00	0.66
	natural	7.42	60.66	0.00	0.54
M-7	AT(2)	12.03	51.40	0.00	0.54
	TRADES(2)	10.80	52.75	0.00	0.61
	TRADES(4)	14.09	51.78	0.00	0.68
	TRADES(8)	19.22	52.96	0.00	0.67
	natural	7.49	51.40	0.00	0.53
M-8	AT(2)	11.88	60.31	0.00	0.47
	TRADES(2)	10.88	60.31	0.00	0.51
	TRADES(4)	15.79	61.43	0.00	0.56
	TRADES(8)	17.15	59.64	0.00	0.59
	natural	7.43	60.31	0.00	0.46
M-9	AT(2)	11.28	51.54	0.00	0.71
	TRADES(2)	13.13	52.28	0.00	0.71
	TRADES(4)	16.99	50.70	0.00	0.74
	TRADES(8)	20.84	51.48	0.00	0.80
	natural	8.12	52.32	0.00	0.61

Table 18: The average empirical robust radius, average OOD distance, percentage of OOD examples covered by the robust norm ball of its closest training example and the NCG accuracy (in the feature space of MNIST datasets).

		empirical robust radius	OOD dist.	portion covered	NCG acc.
C10-0	AT(1)	0.65	1.33	0.01	0.83
	TRADES(2)	0.62	1.33	0.01	0.81
	TRADES(4)	0.27	1.31	0.00	0.83
	TRADES(8)	0.43	1.31	0.00	0.83
	natural	0.48	1.31	0.00	0.80
C10-4	AT(1)	0.69	1.31	0.01	0.84
	TRADES(2)	0.68	1.29	0.01	0.82
	TRADES(4)	0.28	1.31	0.00	0.85
	TRADES(8)	0.45	1.31	0.00	0.85
	natural	0.57	1.31	0.00	0.82
C10-9	AT(1)	0.93	1.43	0.07	0.89
	TRADES(2)	0.85	1.46	0.06	0.83
	TRADES(4)	0.30	1.43	0.00	0.88
	TRADES(8)	0.60	1.43	0.00	0.87
	natural	0.73	1.43	0.03	0.84
C100-0	AT(1)	0.51	1.64	0.00	0.70
	TRADES(2)	0.45	1.64	0.00	0.69
	TRADES(4)	0.29	1.65	0.00	0.68
	TRADES(8)	0.35	1.65	0.00	0.68
	natural	0.30	1.65	0.00	0.63
C100-4	AT(1)	0.64	1.84	0.00	0.74
	TRADES(2)	0.55	1.83	0.00	0.75
	TRADES(4)	0.31	1.83	0.00	0.73
	TRADES(8)	0.45	1.83	0.00	0.74
	natural	0.36	1.83	0.00	0.69
C100-9	AT(1)	0.63	1.75	0.02	0.74
	TRADES(2)	0.52	1.75	0.00	0.72
	TRADES(4)	0.36	1.73	0.00	0.71
	TRADES(8)	0.45	1.73	0.00	0.71
	natural	0.33	1.73	0.00	0.66
I-0	AT(.5)	0.32	1.78	0.00	0.16
	TRADES(2)	0.20	1.63	0.00	0.15
	TRADES(4)	0.12	1.66	0.00	0.12
	TRADES(8)	0.39	1.64	0.00	0.13
	natural	0.07	1.64	0.00	0.11
I-1	AT(.5)	0.23	1.65	0.00	0.15
	TRADES(2)	0.18	1.50	0.00	0.18
	TRADES(4)	0.14	1.51	0.00	0.14
	TRADES(8)	0.34	1.49	0.00	0.15
	natural	0.05	1.47	0.00	0.13
I-2	AT(.5)	0.21	1.67	0.01	0.15
	TRADES(2)	0.22	1.57	0.00	0.15
	TRADES(4)	0.11	1.57	0.00	0.14
	TRADES(8)	0.12	1.56	0.00	0.14
	natural	0.06	1.58	0.00	0.11

Table 19: The average empirical robust radius, average OOD distance, percentage of OOD examples covered by the robust norm ball of its closest training example and the NCG accuracy (in the feature space of C10, C100, and I).



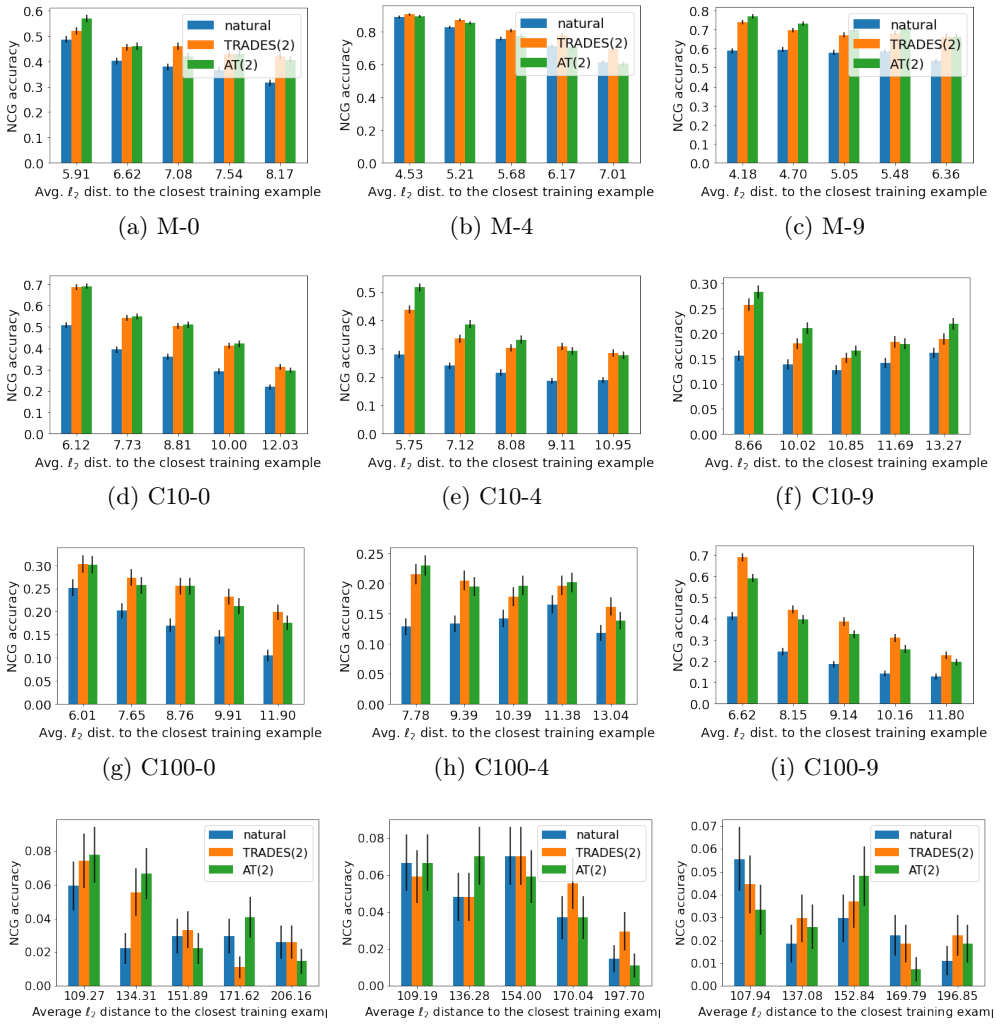
(a) M-0 (b) M-4 (c) M-9

(d) C10-0 (e) C10-4 (f) C10-9

(g) C100-0 (h) C100-4 (i) C100-9

(j) I-0 (k) I-1 (l) I-2

Figure 13: The NCG accuracy and the distance to the closest training example for MNIST, CIFAR10, CIFAR100, and ImageNet-100 in the pixel space.



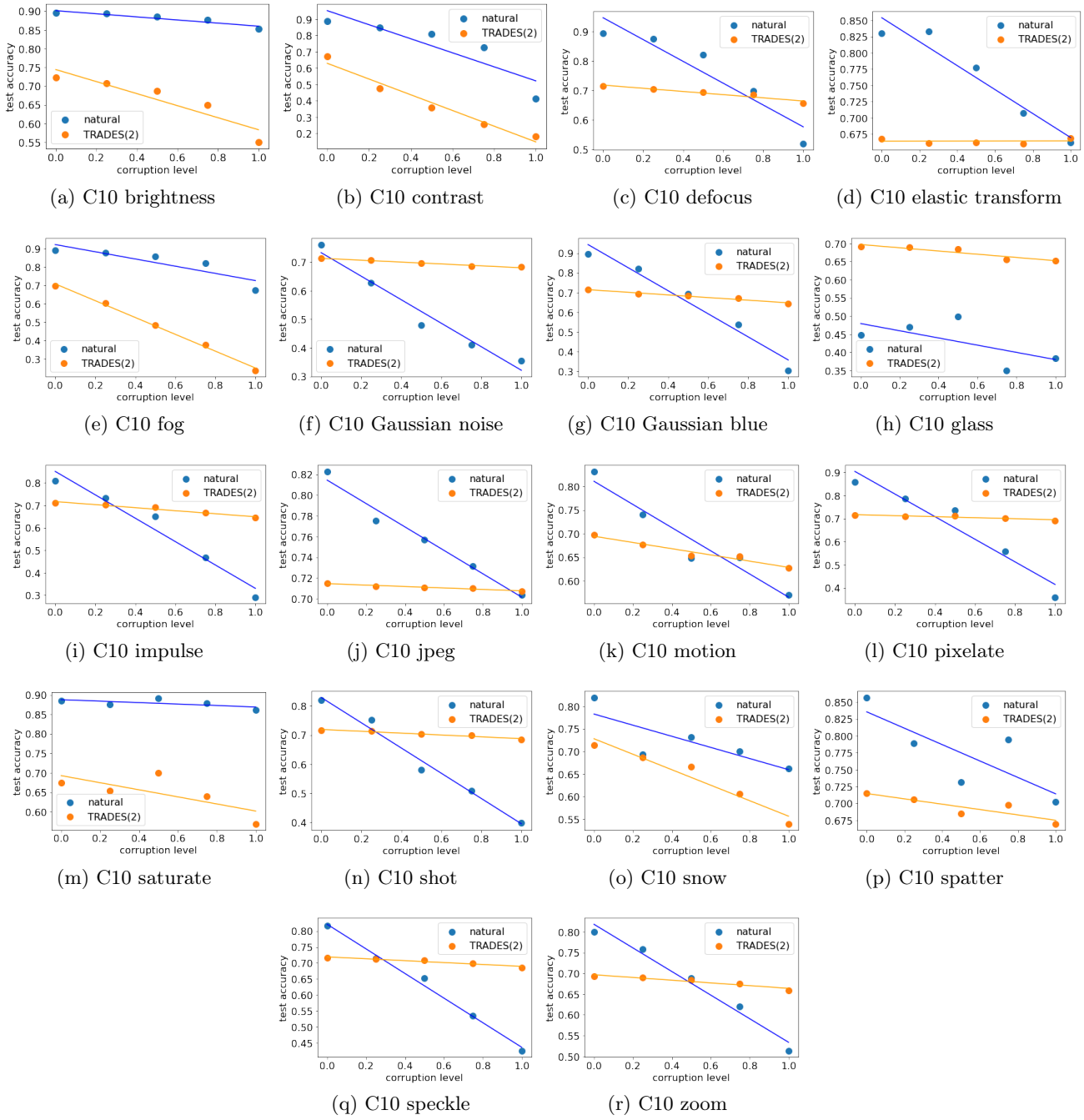
(a) M-0 (b) M-4 (c) M-9

(d) C10-0 (e) C10-4 (f) C10-9

(g) C100-0 (h) C100-4 (i) C100-9

(j) I-0 (k) I-1 (l) I-2

Figure 14: The NCG accuracy and the distance to the closest training example for MNIST, CIFAR10, CIFAR100, and ImageNet-100 in the feature space.



(a) C10 brightness (b) C10 contrast (c) C10 defocus (d) C10 elastic transform

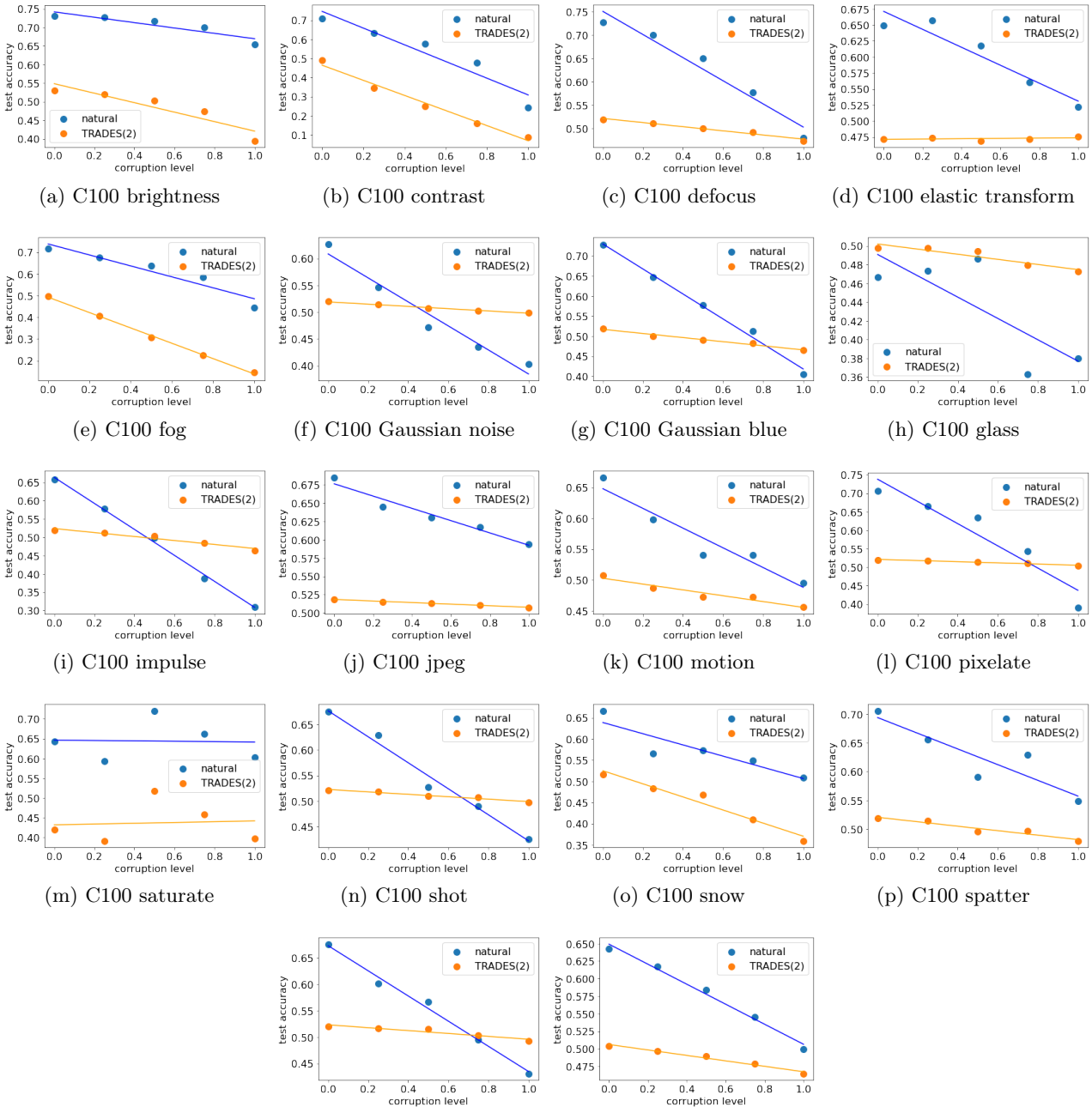
(e) C10 fog (f) C10 Gaussian noise (g) C10 Gaussian blue (h) C10 glass

(i) C10 impulse (j) C10 jpeg (k) C10 motion (l) C10 pixelate

(m) C10 saturate (n) C10 shot (o) C10 snow (p) C10 spatter

(q) C10 speckle (r) C10 zoom

Figure 15: The slopes of the test accuracy of naturally trained models and TRADES(2) on CIFAR10 in the pixel space.



(a) C100 brightness (b) C100 contrast (c) C100 defocus (d) C100 elastic transform

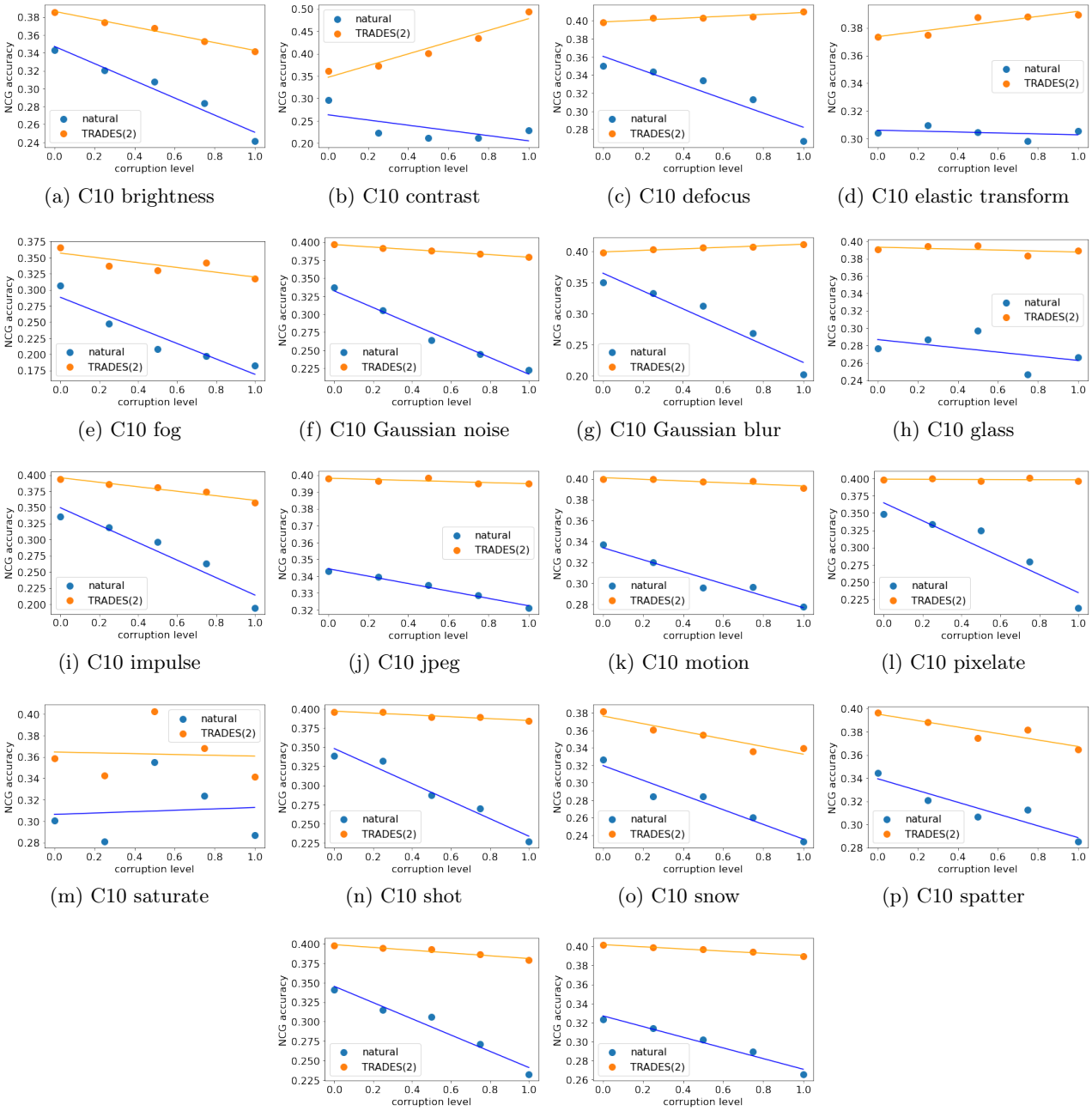
(e) C100 fog (f) C100 Gaussian noise (g) C100 Gaussian blue (h) C100 glass

(i) C100 impulse (j) C100 jpeg (k) C100 motion (l) C100 pixelate

(m) C100 saturate (n) C100 shot (o) C100 snow (p) C100 spatter

(q) C100 speckle (r) C100 zoom

Figure 16: The slopes of the test accuracy of naturally trained models and TRADES(2) on CIFAR100 in the pixel space.



(a) C10 brightness (b) C10 contrast (c) C10 defocus (d) C10 elastic transform

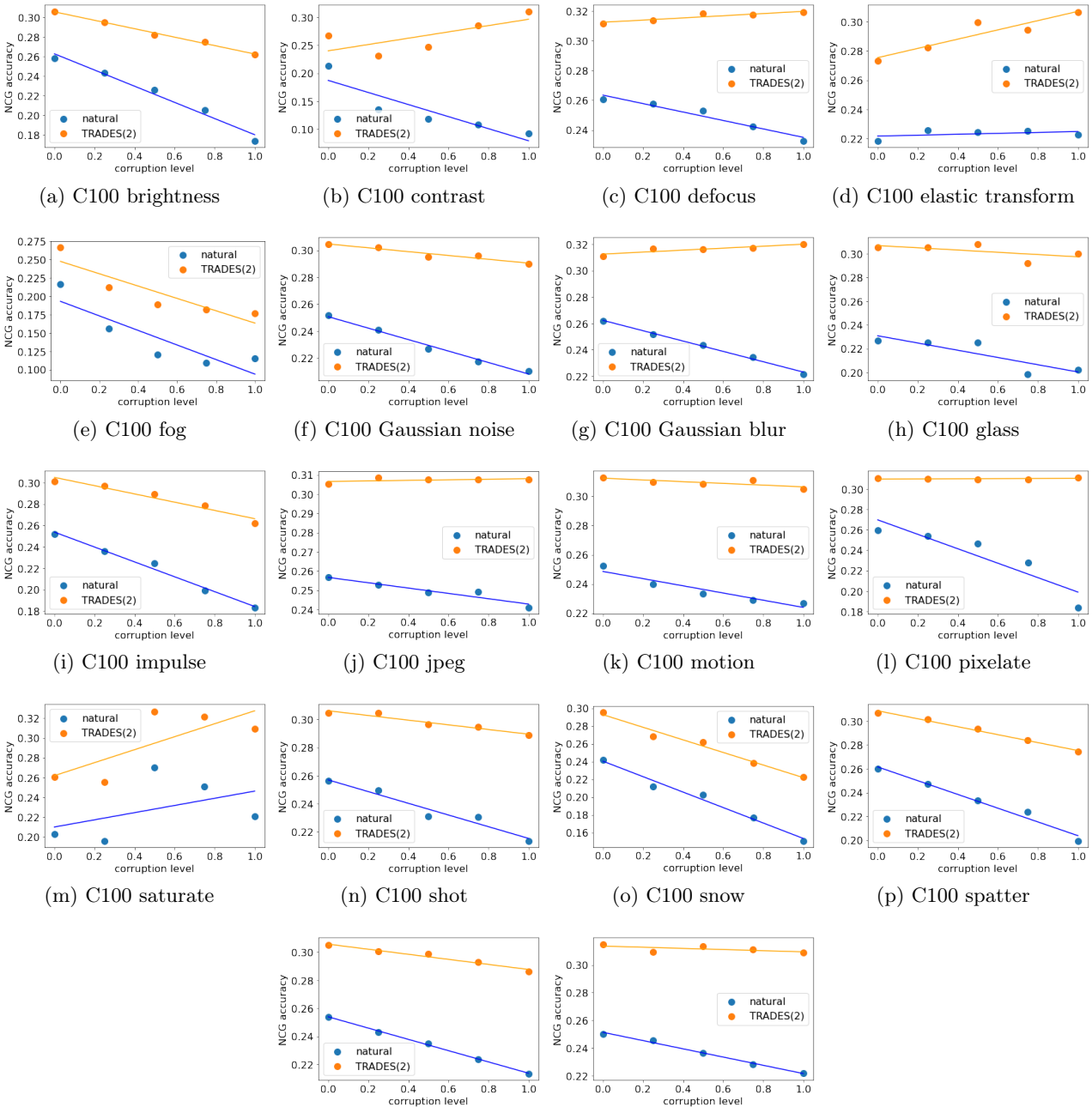
(e) C10 fog (f) C10 Gaussian noise (g) C10 Gaussian blur (h) C10 glass

(i) C10 impulse (j) C10 jpeg (k) C10 motion (l) C10 pixelate

(m) C10 saturate (n) C10 shot (o) C10 snow (p) C10 spatter

(q) C10 speckle (r) C10 zoom

Figure 17: The slopes of the NCG accuracy of naturally trained models and TRADES(2) on CIFAR10 in the pixel space.



(a) C100 brightness (b) C100 contrast (c) C100 defocus (d) C100 elastic transform

(e) C100 fog (f) C100 Gaussian noise (g) C100 Gaussian blur (h) C100 glass

(i) C100 impulse (j) C100 jpeg (k) C100 motion (l) C100 pixelate

(m) C100 saturate (n) C100 shot (o) C100 snow (p) C100 spatter

(q) C100 speckle (r) C100 zoom

Figure 18: The slopes of the NCG accuracy of naturally trained models and TRADES(2) on CIFAR100 in the pixel space.

	model		natural				TRADES(2)			
	dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
pixel	C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
		2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
		3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
		4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
		5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
	C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
		2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
		3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
		4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
		5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29
	I	1	0.42	0.41	0.68	0.04	0.36	0.35	0.51	0.06
		2	0.34	0.33	0.64	0.03	0.36	0.35	0.53	0.05
		3	0.22	0.21	0.49	0.03	0.34	0.33	0.49	0.05
		4	0.12	0.11	0.24	0.02	0.30	0.30	0.45	0.05
		5	0.04	0.04	0.07	0.02	0.22	0.22	0.34	0.04
feature	C10	1	0.74	0.39	0.78	0.89	0.72	0.32	0.77	0.89
		2	0.59	0.35	0.64	0.85	0.56	0.23	0.62	0.85
		3	0.45	0.33	0.48	0.82	0.40	0.19	0.45	0.83
		4	0.39	0.33	0.40	0.81	0.35	0.20	0.38	0.83
		5	0.34	0.28	0.35	0.82	0.31	0.18	0.33	0.83
	C100	1	0.60	0.25	0.72	0.74	0.62	0.29	0.71	0.78
		2	0.51	0.24	0.63	0.68	0.53	0.29	0.62	0.74
		3	0.43	0.23	0.54	0.64	0.44	0.25	0.53	0.69
		4	0.40	0.22	0.51	0.63	0.40	0.23	0.49	0.67
		5	0.37	0.21	0.46	0.61	0.37	0.21	0.46	0.65
	I	1	0.22	0.18	0.44	0.15	0.21	0.18	0.41	0.16
		2	0.19	0.16	0.36	0.14	0.18	0.15	0.34	0.15
		3	0.14	0.12	0.26	0.14	0.13	0.11	0.21	0.17
		4	0.09	0.08	0.16	0.13	0.08	0.07	0.14	0.16
		5	0.05	0.04	0.08	0.14	0.04	0.03	0.08	0.14

Table 20: Here we show models trained on CIFAR10 and CIFAR100 and evaluate them on the gaussian noise corrupted data. The NCG accuracy, test accuracy, the test accuracy on the NCG correct corrupted examples, the test accuracy on the NCG incorrect corrupted examples, and the distance to the closest training example.

			unseen class	top most predicted class	second most predicted class
M-0	pixel	natural	0	6	2
		TRADES(2)	0	2	6
	feature	natural	0	6	2
		TRADES(2)	0	6	2
M-4	pixel	natural	4	9	7
		TRADES(2)	4	9	7
	feature	natural	4	9	7
		TRADES(2)	4	9	7
M-9	pixel	natural	9	4	7
		TRADES(2)	9	4	7
	feature	natural	9	4	8
		TRADES(2)	9	4	8
C100-0	pixel	natural	aquatic mammals	fish	small mammals
		TRADES(2)	aquatic mammals	fish	medium-sized mammals
	feature	natural	aquatic mammals	fish	reptiles
		TRADES(2)	aquatic mammals	fish	reptiles
C100-4	pixel	natural	fruit and vegetables	flowers	food containers
		TRADES(2)	fruit and vegetables	flowers	food containers
	feature	natural	fruit and vegetables	flowers	food containers
		TRADES(2)	fruit and vegetables	flowers	food containers
C100-9	pixel	natural	large man-made outdoor things	large natural outdoor scenes	vehicles 2
		TRADES(2)	large man-made outdoor things	large natural outdoor scenes	trees
	feature	natural	large man-made outdoor things	large natural outdoor scenes	vehicles 2
		TRADES(2)	large man-made outdoor things	large natural outdoor scenes	vehicles 2
C10-0	pixel	natural	airplane	ship	bird
		TRADES(2)	airplane	ship	bird
	feature	natural	airplane	ship	bird
		TRADES(2)	airplane	ship	bird
C10-4	pixel	natural	deer	bird	horse
		TRADES(2)	deer	frog	bird
	feature	natural	deer	bird	horse
		TRADES(2)	deer	bird	cat
C10-9	pixel	natural	truck	automobile	airplane
		TRADES(2)	truck	automobile	ship
	feature	natural	truck	automobile	ship
		TRADES(2)	truck	automobile	ship
I-0	pixel	natural	American robin	lorikeet	stinkhorn mushroom
		TRADES(2)	American robin	lorikeet	hare
	feature	natural	American robin	hare	little blue heron
		TRADES(2)	American robin	hare	little blue heron
I-1	pixel	natural	Gila monster	eastern hog-nosed snake	dung beetle
		TRADES(2)	Gila monster	eastern hog-nosed snake	rock crab
	feature	natural	Gila monster	eastern hog-nosed snake	dung beetle
		TRADES(2)	Gila monster	eastern hog-nosed snake	dung beetle
I-2	pixel	natural	eastern hog-nosed snake	garter snake	Gila monster
		TRADES(2)	eastern hog-nosed snake	garter snake	Gila monster
	feature	natural	eastern hog-nosed snake	Gila monster	garter snake
		TRADES(2)	eastern hog-nosed snake	Gila monster	dung beetle

Table 21: This table shows the top and second most predicted classes on the examples of unseen classes for each dataset.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 22: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type gaussian in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 23: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type impulse in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 24: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type shot in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 25: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type defocus in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 26: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type motion in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 27: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type zoom in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 28: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type glass in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 29: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type snow in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 30: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type fog in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 31: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type contrast in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 32: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type pixelate in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 33: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type brightness in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 35: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type gaussian blur in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 36: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type jpeg compression in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 37: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type saturate in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 38: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type spatter in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.70	0.88	0.34	0.71	0.67	0.78	0.40
	2	0.63	0.54	0.82	0.30	0.71	0.66	0.78	0.39
	3	0.48	0.39	0.75	0.26	0.70	0.65	0.77	0.39
	4	0.41	0.32	0.70	0.24	0.69	0.63	0.77	0.38
	5	0.36	0.27	0.66	0.22	0.68	0.63	0.77	0.38
C100	1	0.63	0.56	0.84	0.25	0.52	0.43	0.72	0.30
	2	0.55	0.47	0.79	0.24	0.51	0.43	0.71	0.30
	3	0.47	0.39	0.74	0.23	0.51	0.42	0.71	0.30
	4	0.44	0.36	0.71	0.22	0.50	0.42	0.71	0.30
	5	0.40	0.33	0.67	0.21	0.50	0.41	0.71	0.29

Table 39: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type speckle noise in the pixel space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.74	0.39	0.78	0.89	0.72	0.32	0.77	0.89
	2	0.59	0.35	0.64	0.85	0.56	0.23	0.62	0.85
	3	0.45	0.33	0.48	0.82	0.40	0.19	0.45	0.83
	4	0.39	0.33	0.40	0.81	0.35	0.20	0.38	0.83
	5	0.34	0.28	0.35	0.82	0.31	0.18	0.33	0.83
C100	1	0.60	0.25	0.72	0.74	0.62	0.29	0.71	0.78
	2	0.51	0.24	0.63	0.68	0.53	0.29	0.62	0.74
	3	0.43	0.23	0.54	0.64	0.44	0.25	0.53	0.69
	4	0.40	0.22	0.51	0.63	0.40	0.23	0.49	0.67
	5	0.37	0.21	0.46	0.61	0.37	0.21	0.46	0.65

Table 40: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type gaussian in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.80	0.36	0.84	0.92	0.80	0.32	0.84	0.92
	2	0.72	0.35	0.77	0.89	0.71	0.31	0.76	0.89
	3	0.65	0.34	0.70	0.86	0.62	0.27	0.68	0.85
	4	0.47	0.36	0.50	0.80	0.42	0.23	0.48	0.79
	5	0.32	0.34	0.31	0.80	0.27	0.20	0.29	0.79
C100	1	0.63	0.26	0.75	0.76	0.65	0.29	0.73	0.80
	2	0.54	0.25	0.66	0.69	0.56	0.29	0.65	0.74
	3	0.46	0.22	0.59	0.65	0.47	0.25	0.57	0.71
	4	0.35	0.20	0.45	0.59	0.35	0.21	0.44	0.63
	5	0.28	0.19	0.34	0.56	0.27	0.17	0.34	0.57

Table 41: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type impulse in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.80	0.38	0.84	0.91	0.80	0.34	0.84	0.91
	2	0.72	0.35	0.77	0.88	0.71	0.30	0.76	0.88
	3	0.55	0.33	0.59	0.83	0.51	0.22	0.56	0.84
	4	0.47	0.30	0.51	0.83	0.43	0.19	0.48	0.83
	5	0.38	0.30	0.39	0.81	0.33	0.17	0.36	0.83
C100	1	0.65	0.27	0.76	0.77	0.67	0.31	0.75	0.81
	2	0.59	0.26	0.71	0.75	0.61	0.30	0.70	0.78
	3	0.49	0.24	0.61	0.68	0.50	0.26	0.59	0.73
	4	0.45	0.23	0.56	0.66	0.46	0.25	0.54	0.70
	5	0.39	0.21	0.49	0.63	0.39	0.22	0.48	0.66

Table 42: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type shot in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.89	0.41	0.91	0.95	0.88	0.37	0.91	0.95
	2	0.86	0.39	0.90	0.94	0.85	0.31	0.89	0.94
	3	0.80	0.41	0.83	0.92	0.77	0.27	0.82	0.90
	4	0.67	0.44	0.71	0.85	0.61	0.25	0.68	0.85
	5	0.49	0.42	0.51	0.77	0.42	0.24	0.47	0.79
C100	1	0.71	0.29	0.80	0.81	0.72	0.33	0.79	0.85
	2	0.68	0.28	0.78	0.80	0.69	0.31	0.76	0.83
	3	0.62	0.27	0.73	0.77	0.63	0.28	0.71	0.81
	4	0.55	0.29	0.65	0.72	0.54	0.26	0.62	0.77
	5	0.46	0.27	0.55	0.67	0.44	0.24	0.51	0.73

Table 43: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type defocus in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.81	0.39	0.85	0.92	0.78	0.24	0.84	0.90
	2	0.71	0.39	0.75	0.87	0.65	0.20	0.72	0.86
	3	0.61	0.43	0.65	0.83	0.54	0.21	0.61	0.82
	4	0.61	0.42	0.65	0.83	0.54	0.19	0.61	0.82
	5	0.54	0.45	0.56	0.80	0.45	0.21	0.51	0.80
C100	1	0.64	0.30	0.75	0.77	0.65	0.30	0.73	0.81
	2	0.58	0.29	0.69	0.73	0.58	0.28	0.66	0.78
	3	0.53	0.28	0.64	0.70	0.51	0.25	0.59	0.76
	4	0.53	0.29	0.63	0.69	0.51	0.26	0.59	0.76
	5	0.48	0.28	0.58	0.67	0.45	0.24	0.53	0.74

Table 44: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type motion in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.76	0.38	0.81	0.89	0.73	0.24	0.80	0.88
	2	0.72	0.38	0.77	0.86	0.68	0.24	0.75	0.86
	3	0.64	0.38	0.70	0.82	0.59	0.22	0.66	0.83
	4	0.57	0.39	0.62	0.79	0.51	0.21	0.58	0.81
	5	0.47	0.33	0.51	0.75	0.41	0.20	0.46	0.79
C100	1	0.62	0.29	0.72	0.77	0.62	0.29	0.70	0.80
	2	0.60	0.28	0.71	0.74	0.59	0.28	0.68	0.79
	3	0.56	0.28	0.67	0.72	0.55	0.26	0.64	0.77
	4	0.53	0.28	0.63	0.70	0.51	0.25	0.60	0.75
	5	0.49	0.28	0.59	0.67	0.46	0.24	0.55	0.73

Table 45: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type zoom in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.44	0.27	0.48	0.80	0.42	0.22	0.47	0.81
	2	0.46	0.29	0.50	0.81	0.44	0.23	0.49	0.81
	3	0.48	0.27	0.53	0.81	0.46	0.22	0.52	0.81
	4	0.36	0.23	0.40	0.79	0.35	0.20	0.39	0.80
	5	0.38	0.22	0.42	0.79	0.37	0.22	0.41	0.80
C100	1	0.42	0.21	0.53	0.67	0.44	0.24	0.51	0.73
	2	0.43	0.20	0.55	0.67	0.45	0.24	0.53	0.73
	3	0.45	0.20	0.56	0.68	0.48	0.28	0.55	0.73
	4	0.33	0.18	0.41	0.63	0.33	0.20	0.39	0.71
	5	0.35	0.17	0.45	0.64	0.37	0.22	0.43	0.71

Table 46: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type glass in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.81	0.38	0.85	0.92	0.80	0.34	0.84	0.92
	2	0.68	0.41	0.73	0.86	0.65	0.28	0.70	0.86
	3	0.72	0.39	0.77	0.87	0.70	0.30	0.75	0.88
	4	0.70	0.40	0.74	0.86	0.67	0.27	0.73	0.87
	5	0.65	0.40	0.70	0.85	0.62	0.27	0.68	0.86
C100	1	0.64	0.27	0.75	0.77	0.65	0.31	0.73	0.81
	2	0.54	0.26	0.65	0.71	0.55	0.28	0.63	0.76
	3	0.54	0.23	0.66	0.72	0.56	0.27	0.65	0.77
	4	0.52	0.23	0.64	0.71	0.54	0.26	0.62	0.76
	5	0.48	0.24	0.59	0.68	0.49	0.27	0.57	0.73

Table 47: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type snow in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.88	0.41	0.91	0.95	0.88	0.33	0.90	0.95
	2	0.87	0.40	0.90	0.93	0.85	0.29	0.89	0.94
	3	0.84	0.41	0.88	0.93	0.82	0.26	0.87	0.92
	4	0.81	0.41	0.84	0.91	0.78	0.26	0.83	0.90
	5	0.65	0.38	0.69	0.85	0.61	0.25	0.67	0.84
C100	1	0.70	0.28	0.80	0.80	0.71	0.31	0.78	0.84
	2	0.66	0.29	0.76	0.78	0.67	0.29	0.75	0.82
	3	0.62	0.28	0.73	0.76	0.62	0.30	0.70	0.81
	4	0.56	0.27	0.67	0.73	0.57	0.27	0.65	0.78
	5	0.43	0.23	0.52	0.68	0.44	0.24	0.51	0.74

Table 48: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type fog in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.88	0.39	0.91	0.95	0.87	0.31	0.90	0.95
	2	0.83	0.39	0.87	0.93	0.81	0.24	0.86	0.92
	3	0.79	0.43	0.83	0.90	0.76	0.27	0.81	0.90
	4	0.70	0.48	0.74	0.85	0.64	0.27	0.70	0.86
	5	0.40	0.37	0.42	0.70	0.33	0.24	0.36	0.79
C100	1	0.69	0.30	0.79	0.80	0.70	0.33	0.77	0.84
	2	0.62	0.31	0.72	0.76	0.62	0.31	0.69	0.80
	3	0.55	0.31	0.65	0.72	0.54	0.28	0.62	0.77
	4	0.46	0.30	0.54	0.68	0.43	0.25	0.50	0.73
	5	0.24	0.25	0.23	0.69	0.18	0.14	0.19	0.77

Table 49: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type contrast in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.85	0.37	0.88	0.93	0.84	0.36	0.88	0.93
	2	0.78	0.39	0.82	0.89	0.77	0.37	0.81	0.90
	3	0.73	0.40	0.77	0.88	0.71	0.35	0.76	0.89
	4	0.55	0.37	0.59	0.83	0.53	0.29	0.57	0.85
	5	0.36	0.27	0.38	0.82	0.33	0.18	0.36	0.83
C100	1	0.68	0.27	0.79	0.79	0.70	0.32	0.78	0.83
	2	0.64	0.24	0.75	0.77	0.66	0.31	0.74	0.81
	3	0.60	0.26	0.72	0.75	0.63	0.31	0.71	0.80
	4	0.51	0.24	0.62	0.70	0.53	0.30	0.60	0.77
	5	0.36	0.21	0.44	0.66	0.38	0.27	0.42	0.74

Table 50: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type pixelate in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.89	0.39	0.91	0.95	0.88	0.36	0.91	0.95
	2	0.88	0.37	0.91	0.95	0.88	0.30	0.91	0.95
	3	0.88	0.39	0.90	0.95	0.87	0.32	0.90	0.94
	4	0.87	0.39	0.90	0.94	0.86	0.30	0.89	0.94
	5	0.84	0.36	0.88	0.93	0.83	0.28	0.87	0.93
C100	1	0.71	0.29	0.81	0.81	0.72	0.32	0.79	0.85
	2	0.71	0.29	0.81	0.81	0.71	0.32	0.79	0.84
	3	0.70	0.28	0.80	0.80	0.71	0.31	0.78	0.84
	4	0.68	0.30	0.78	0.79	0.69	0.33	0.76	0.83
	5	0.62	0.28	0.74	0.75	0.64	0.30	0.72	0.80

Table 51: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type brightness in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.81	0.40	0.85	0.92	0.78	0.27	0.84	0.91
	2	0.81	0.39	0.85	0.91	0.79	0.27	0.84	0.91
	3	0.76	0.40	0.80	0.89	0.72	0.26	0.78	0.89
	4	0.67	0.34	0.73	0.86	0.63	0.20	0.70	0.86
	5	0.64	0.35	0.69	0.85	0.60	0.22	0.67	0.84
C100	1	0.63	0.27	0.74	0.76	0.64	0.30	0.72	0.80
	2	0.63	0.29	0.74	0.76	0.64	0.31	0.72	0.81
	3	0.59	0.27	0.70	0.75	0.60	0.28	0.68	0.80
	4	0.54	0.25	0.66	0.72	0.55	0.26	0.64	0.77
	5	0.50	0.22	0.61	0.70	0.52	0.27	0.60	0.76

Table 52: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type elastic transform in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.89	0.41	0.91	0.95	0.88	0.37	0.91	0.95
	2	0.80	0.41	0.83	0.91	0.77	0.27	0.82	0.91
	3	0.67	0.45	0.71	0.84	0.60	0.24	0.67	0.85
	4	0.51	0.45	0.53	0.77	0.44	0.25	0.48	0.79
	5	0.31	0.33	0.30	0.76	0.26	0.16	0.30	0.74
C100	1	0.71	0.29	0.81	0.81	0.72	0.33	0.79	0.85
	2	0.62	0.27	0.73	0.77	0.62	0.27	0.71	0.81
	3	0.55	0.29	0.66	0.72	0.54	0.26	0.63	0.77
	4	0.49	0.27	0.58	0.69	0.47	0.24	0.55	0.74
	5	0.40	0.26	0.48	0.62	0.35	0.19	0.41	0.73

Table 53: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type gaussian blur in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.82	0.38	0.86	0.92	0.80	0.31	0.85	0.91
	2	0.77	0.40	0.81	0.89	0.75	0.32	0.81	0.89
	3	0.75	0.40	0.79	0.89	0.73	0.33	0.79	0.88
	4	0.72	0.37	0.77	0.88	0.70	0.28	0.76	0.88
	5	0.70	0.38	0.75	0.86	0.68	0.30	0.73	0.86
C100	1	0.66	0.26	0.77	0.78	0.67	0.29	0.76	0.82
	2	0.62	0.24	0.74	0.76	0.64	0.30	0.72	0.81
	3	0.61	0.25	0.72	0.76	0.63	0.30	0.71	0.81
	4	0.59	0.24	0.71	0.75	0.61	0.29	0.69	0.80
	5	0.57	0.24	0.69	0.73	0.59	0.28	0.67	0.79

Table 54: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type jpeg compression in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.87	0.39	0.90	0.95	0.87	0.37	0.90	0.95
	2	0.87	0.36	0.90	0.94	0.86	0.32	0.89	0.94
	3	0.88	0.38	0.91	0.95	0.88	0.36	0.91	0.95
	4	0.87	0.39	0.90	0.94	0.87	0.37	0.90	0.94
	5	0.85	0.40	0.89	0.93	0.84	0.32	0.88	0.93
C100	1	0.62	0.25	0.73	0.77	0.64	0.30	0.72	0.81
	2	0.56	0.24	0.68	0.74	0.58	0.28	0.67	0.78
	3	0.69	0.29	0.80	0.79	0.70	0.32	0.78	0.83
	4	0.63	0.28	0.75	0.76	0.64	0.30	0.72	0.80
	5	0.56	0.26	0.67	0.73	0.58	0.31	0.66	0.77

Table 55: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type saturate in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.85	0.38	0.88	0.94	0.84	0.35	0.88	0.94
	2	0.78	0.34	0.82	0.91	0.78	0.33	0.82	0.91
	3	0.72	0.37	0.77	0.89	0.72	0.34	0.77	0.88
	4	0.79	0.37	0.83	0.91	0.79	0.36	0.83	0.92
	5	0.71	0.34	0.76	0.88	0.70	0.31	0.76	0.88
C100	1	0.68	0.28	0.79	0.79	0.69	0.31	0.77	0.83
	2	0.62	0.25	0.74	0.76	0.64	0.30	0.73	0.80
	3	0.55	0.25	0.68	0.72	0.58	0.30	0.66	0.76
	4	0.60	0.25	0.72	0.74	0.63	0.30	0.71	0.79
	5	0.52	0.23	0.65	0.69	0.54	0.28	0.63	0.75

Table 56: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type spatter in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
C10	1	0.80	0.35	0.84	0.92	0.79	0.33	0.84	0.92
	2	0.68	0.35	0.73	0.88	0.67	0.30	0.72	0.88
	3	0.62	0.32	0.67	0.86	0.59	0.22	0.66	0.86
	4	0.50	0.29	0.55	0.83	0.46	0.18	0.52	0.83
	5	0.40	0.29	0.43	0.82	0.35	0.15	0.40	0.82
C100	1	0.65	0.27	0.76	0.77	0.66	0.31	0.74	0.81
	2	0.57	0.26	0.69	0.73	0.59	0.29	0.67	0.77
	3	0.53	0.25	0.64	0.70	0.54	0.27	0.63	0.74
	4	0.45	0.23	0.57	0.66	0.46	0.25	0.55	0.71
	5	0.39	0.21	0.49	0.64	0.39	0.21	0.49	0.66

Table 57: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type speckle noise in the feature space for naturally trained and robust models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.50	0.49	0.70	0.04	0.35	0.35	0.48	0.06
	2	0.45	0.44	0.64	0.04	0.31	0.31	0.44	0.05
	3	0.38	0.38	0.49	0.04	0.24	0.24	0.29	0.05
	4	0.30	0.30	0.34	0.04	0.17	0.17	0.17	0.07
	5	0.23	0.22	0.25	0.04	0.12	0.12	0.12	0.09

Table 58: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type brightness in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.24	0.24	0.26	0.05	0.12	0.12	0.14	0.10
	2	0.15	0.15	0.18	0.06	0.07	0.07	0.10	0.13
	3	0.08	0.08	0.11	0.08	0.05	0.04	0.07	0.15
	4	0.04	0.03	0.08	0.12	0.03	0.03	0.06	0.16
	5	0.03	0.02	0.06	0.12	0.03	0.02	0.06	0.15

Table 59: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type contrast in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.32	0.31	0.57	0.05	0.32	0.31	0.48	0.06
	2	0.20	0.19	0.42	0.04	0.30	0.29	0.46	0.06
	3	0.07	0.06	0.28	0.04	0.25	0.24	0.38	0.07
	4	0.05	0.04	0.19	0.04	0.22	0.21	0.34	0.08
	5	0.04	0.04	0.18	0.05	0.19	0.18	0.29	0.08

Table 60: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type defocus in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.45	0.44	0.69	0.04	0.31	0.30	0.49	0.05
	2	0.32	0.31	0.49	0.03	0.22	0.22	0.30	0.05
	3	0.46	0.45	0.70	0.04	0.36	0.35	0.51	0.06
	4	0.42	0.40	0.67	0.04	0.35	0.34	0.52	0.06
	5	0.31	0.30	0.60	0.04	0.33	0.32	0.51	0.06

Table 61: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type elastic in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.22	0.23	0.17	0.04	0.06	0.06	0.06	0.08
	2	0.16	0.16	0.12	0.05	0.04	0.04	0.05	0.10
	3	0.11	0.11	0.08	0.05	0.03	0.03	0.03	0.10
	4	0.09	0.10	0.07	0.05	0.03	0.03	0.03	0.10
	5	0.05	0.05	0.07	0.06	0.02	0.02	0.02	0.09

Table 62: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type fog in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.33	0.33	0.45	0.03	0.23	0.22	0.27	0.05
	2	0.19	0.19	0.25	0.03	0.13	0.13	0.13	0.06
	3	0.13	0.12	0.20	0.03	0.09	0.09	0.09	0.08
	4	0.12	0.12	0.18	0.03	0.08	0.09	0.08	0.07
	5	0.09	0.09	0.14	0.03	0.07	0.07	0.06	0.08

Table 63: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type frost in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.42	0.41	0.68	0.04	0.36	0.35	0.51	0.06
	2	0.34	0.33	0.64	0.03	0.36	0.35	0.53	0.05
	3	0.22	0.21	0.49	0.03	0.34	0.33	0.49	0.05
	4	0.12	0.11	0.24	0.02	0.30	0.30	0.45	0.05
	5	0.04	0.04	0.07	0.02	0.22	0.22	0.34	0.04

Table 64: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type gaussian in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.42	0.41	0.65	0.05	0.34	0.33	0.48	0.06
	2	0.31	0.30	0.57	0.04	0.32	0.31	0.48	0.06
	3	0.19	0.18	0.45	0.04	0.30	0.29	0.41	0.07
	4	0.13	0.13	0.33	0.04	0.27	0.26	0.40	0.07
	5	0.07	0.06	0.23	0.04	0.22	0.21	0.36	0.07

Table 65: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type glass in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.38	0.37	0.70	0.03	0.36	0.35	0.51	0.05
	2	0.27	0.27	0.49	0.03	0.35	0.34	0.49	0.05
	3	0.21	0.20	0.41	0.03	0.34	0.34	0.50	0.05
	4	0.10	0.10	0.15	0.02	0.29	0.29	0.43	0.05
	5	0.04	0.04	0.09	0.02	0.22	0.21	0.34	0.04

Table 66: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type impulse in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.49	0.48	0.72	0.04	0.36	0.35	0.50	0.06
	2	0.49	0.48	0.73	0.04	0.36	0.35	0.49	0.06
	3	0.48	0.47	0.69	0.04	0.36	0.35	0.50	0.06
	4	0.47	0.46	0.70	0.04	0.36	0.35	0.52	0.06
	5	0.43	0.42	0.71	0.04	0.36	0.35	0.51	0.06

Table 67: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type jpeg in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.42	0.41	0.65	0.05	0.34	0.33	0.48	0.06
	2	0.29	0.28	0.54	0.04	0.32	0.31	0.45	0.07
	3	0.18	0.17	0.37	0.04	0.27	0.26	0.41	0.07
	4	0.12	0.11	0.27	0.05	0.24	0.23	0.38	0.07
	5	0.09	0.08	0.22	0.06	0.21	0.20	0.33	0.07

Table 68: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type motion in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.49	0.48	0.71	0.05	0.36	0.35	0.50	0.06
	2	0.49	0.48	0.71	0.05	0.35	0.35	0.50	0.06
	3	0.47	0.46	0.70	0.05	0.35	0.34	0.49	0.06
	4	0.43	0.42	0.69	0.04	0.34	0.33	0.48	0.06
	5	0.40	0.39	0.65	0.04	0.33	0.32	0.48	0.06

Table 69: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type pixelate in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.42	0.41	0.67	0.04	0.36	0.35	0.52	0.06
	2	0.33	0.32	0.59	0.03	0.36	0.35	0.51	0.05
	3	0.23	0.22	0.43	0.03	0.34	0.33	0.50	0.05
	4	0.12	0.11	0.30	0.02	0.29	0.28	0.48	0.05
	5	0.07	0.07	0.20	0.02	0.24	0.23	0.40	0.05

Table 70: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type shot in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.30	0.29	0.50	0.02	0.27	0.27	0.37	0.04
	2	0.15	0.15	0.28	0.02	0.17	0.17	0.18	0.04
	3	0.15	0.15	0.26	0.02	0.15	0.15	0.18	0.04
	4	0.09	0.09	0.10	0.02	0.08	0.08	0.09	0.05
	5	0.08	0.08	0.10	0.03	0.07	0.07	0.08	0.07

Table 71: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type snow in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.33	0.32	0.57	0.05	0.32	0.31	0.46	0.06
	2	0.26	0.25	0.50	0.05	0.30	0.29	0.45	0.06
	3	0.22	0.21	0.41	0.05	0.27	0.26	0.41	0.07
	4	0.19	0.18	0.37	0.05	0.25	0.24	0.42	0.07
	5	0.16	0.15	0.32	0.05	0.23	0.21	0.38	0.07

Table 72: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type zoom in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.26	0.21	0.51	0.16	0.24	0.19	0.45	0.18
	2	0.25	0.21	0.50	0.16	0.23	0.18	0.45	0.17
	3	0.23	0.19	0.45	0.16	0.21	0.17	0.42	0.17
	4	0.20	0.16	0.38	0.15	0.19	0.15	0.35	0.17
	5	0.17	0.14	0.31	0.16	0.16	0.13	0.29	0.17

Table 73: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type brightness in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.07	0.06	0.13	0.15	0.05	0.04	0.11	0.15
	2	0.04	0.04	0.08	0.19	0.03	0.02	0.08	0.19
	3	0.03	0.02	0.05	0.29	0.02	0.01	0.04	0.32
	4	0.02	0.02	0.02	0.47	0.01	0.01	0.01	0.56
	5	0.02	0.02	0.01	0.56	0.01	0.01	0.01	0.59

Table 74: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type contrast in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.10	0.09	0.17	0.13	0.09	0.07	0.16	0.14
	2	0.06	0.05	0.11	0.14	0.05	0.04	0.10	0.14
	3	0.04	0.03	0.07	0.19	0.03	0.02	0.06	0.15
	4	0.03	0.03	0.05	0.24	0.02	0.02	0.04	0.18
	5	0.03	0.03	0.04	0.30	0.02	0.01	0.03	0.23

Table 75: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type defocus in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.20	0.17	0.39	0.13	0.18	0.15	0.35	0.15
	2	0.13	0.11	0.26	0.12	0.11	0.10	0.21	0.15
	3	0.20	0.17	0.40	0.14	0.18	0.15	0.36	0.15
	4	0.19	0.16	0.34	0.13	0.16	0.13	0.29	0.16
	5	0.13	0.12	0.22	0.13	0.11	0.10	0.19	0.15

Table 76: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type elastic in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.05	0.05	0.09	0.14	0.04	0.04	0.09	0.15
	2	0.04	0.03	0.07	0.17	0.03	0.02	0.07	0.17
	3	0.03	0.02	0.05	0.19	0.02	0.02	0.05	0.21
	4	0.03	0.02	0.06	0.17	0.02	0.02	0.05	0.19
	5	0.03	0.02	0.05	0.16	0.02	0.01	0.04	0.19

Table 77: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type fog in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.19	0.16	0.37	0.13	0.17	0.15	0.32	0.15
	2	0.11	0.10	0.21	0.12	0.10	0.09	0.17	0.13
	3	0.07	0.06	0.14	0.13	0.06	0.05	0.13	0.13
	4	0.06	0.06	0.11	0.13	0.05	0.05	0.11	0.13
	5	0.05	0.04	0.09	0.12	0.04	0.04	0.09	0.13

Table 78: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type frost in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
	1	0.22	0.18	0.44	0.15	0.21	0.18	0.41	0.16
	2	0.19	0.16	0.36	0.14	0.18	0.15	0.34	0.15
I	3	0.14	0.12	0.26	0.14	0.13	0.11	0.21	0.17
	4	0.09	0.08	0.16	0.13	0.08	0.07	0.14	0.16
	5	0.05	0.04	0.08	0.14	0.04	0.03	0.08	0.14

Table 79: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type gaussian in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
	1	0.16	0.14	0.31	0.13	0.14	0.12	0.27	0.15
	2	0.10	0.09	0.16	0.13	0.08	0.07	0.12	0.15
I	3	0.06	0.05	0.12	0.13	0.05	0.04	0.09	0.15
	4	0.05	0.04	0.09	0.14	0.03	0.03	0.07	0.15
	5	0.04	0.03	0.08	0.15	0.02	0.02	0.06	0.15

Table 80: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type glass in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
	1	0.19	0.16	0.38	0.15	0.19	0.15	0.36	0.16
	2	0.15	0.13	0.29	0.14	0.14	0.12	0.24	0.16
I	3	0.12	0.10	0.25	0.13	0.11	0.09	0.23	0.16
	4	0.08	0.07	0.13	0.13	0.06	0.05	0.11	0.15
	5	0.04	0.04	0.07	0.14	0.03	0.03	0.07	0.14

Table 81: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type impulse in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.25	0.21	0.46	0.16	0.23	0.19	0.43	0.16
	2	0.25	0.20	0.46	0.16	0.22	0.18	0.42	0.17
	3	0.25	0.20	0.47	0.16	0.23	0.19	0.43	0.17
	4	0.24	0.20	0.45	0.15	0.22	0.18	0.39	0.16
	5	0.23	0.19	0.45	0.15	0.21	0.17	0.41	0.16

Table 82: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type jpeg in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.16	0.14	0.31	0.13	0.15	0.13	0.26	0.15
	2	0.11	0.10	0.17	0.13	0.09	0.08	0.15	0.15
	3	0.07	0.06	0.10	0.16	0.05	0.04	0.09	0.17
	4	0.04	0.04	0.07	0.19	0.03	0.03	0.06	0.21
	5	0.04	0.03	0.06	0.21	0.03	0.02	0.05	0.23

Table 83: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type motion in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.23	0.19	0.45	0.15	0.21	0.17	0.41	0.15
	2	0.23	0.19	0.43	0.15	0.20	0.17	0.38	0.17
	3	0.20	0.17	0.39	0.14	0.17	0.14	0.34	0.15
	4	0.17	0.15	0.32	0.13	0.16	0.13	0.29	0.15
	5	0.15	0.13	0.30	0.13	0.14	0.12	0.27	0.14

Table 84: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type pixelate in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.22	0.18	0.43	0.15	0.21	0.17	0.39	0.17
	2	0.18	0.15	0.34	0.14	0.17	0.14	0.33	0.15
	3	0.14	0.12	0.27	0.14	0.13	0.11	0.23	0.16
	4	0.09	0.08	0.16	0.13	0.07	0.06	0.14	0.15
	5	0.06	0.05	0.10	0.14	0.05	0.04	0.09	0.16

Table 85: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type shot in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.14	0.11	0.29	0.14	0.13	0.11	0.28	0.15
	2	0.08	0.07	0.16	0.15	0.08	0.07	0.14	0.16
	3	0.07	0.06	0.13	0.15	0.07	0.05	0.12	0.18
	4	0.05	0.05	0.07	0.17	0.05	0.05	0.07	0.19
	5	0.06	0.05	0.08	0.16	0.05	0.05	0.09	0.17

Table 86: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type snow in the pixel space for naturally trained and robsut models.

	model	natural				TRADES(2)			
dataset	level	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.	tst acc.	NCG incorrect tst acc.	NCG correct tst acc.	NCG acc.
I	1	0.13	0.11	0.22	0.13	0.12	0.10	0.18	0.16
	2	0.10	0.09	0.18	0.13	0.09	0.08	0.14	0.15
	3	0.09	0.08	0.14	0.14	0.07	0.06	0.12	0.15
	4	0.08	0.07	0.14	0.14	0.06	0.05	0.12	0.16
	5	0.06	0.06	0.10	0.15	0.05	0.04	0.09	0.16

Table 87: The NCG accuracy, test accuacy and the test accuracy conditioned on the NCG correctness on corruption type zoom in the pixel space for naturally trained and robsut models.