Quantifying Adversarial Sensitivity of a Model as a Function of the Image Distribution

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we propose an adaptation to the area under the curve (AUC) metric to measure the adversarial robustness of a model over a particular ϵ-interval $[\epsilon_0, \epsilon_1]$ (interval of adversarial perturbation strengths) that facilitates comparisons across models when they have different initial ϵ_0 performance. This can be used to determine how adversarially sensitive a model is to different image distributions; and/or to measure how robust a model is comparatively to other models for the same distribution. We used this adversarial robustness metric on MNIST, CIFAR-10, and a Fusion dataset (CIFAR-10 + MNIST) where trained models performed either a digit or object recognition task using a LeNet, ResNet50, or a fully connected network (FullyConnectedNet) architecture and found the following: 1) CIFAR-10 models are more adversarially sensitive than MNIST models; 2) Pretraining with another image distribution *sometimes* carries over the adversarial sensitivity induced from the image distribution – contingent on the pretrained image manifold; 3) Increasing the complexity of the image manifold increases the adversarial sensitivity of a model trained on that image manifold, but also shows that the task plays a role on the sensitivity. Collectively, our results imply non-trivial differences of the learned representation space of one perceptual system over another given its exposure to different image statistics (mainly objects vs digits). Moreover, these results hold even when model systems are equalized to have the same level of performance, or when exposed to matched image statistics of fusion images but with different tasks.

1 Introduction

Adversarial images are perturbed visual stimuli that can fool a high performing image classifier with carefully chosen noise that is often imperceptible to humans (Szegedy et al., 2013; Goodfellow et al., 2014). These images are synthesized using an optimization procedure that *maximizes* the wrong output class of a model observer, while *minimizing* any noticeable differences in the image for a reference observer – usually a human (Lubin, 1997). Understanding why they exist has been studied extensively in machine learning as a way to explore gaps in generalization (Gilmer et al., 2018; Yuan et al., 2019; Ilyas et al., 2019), computer vision with applications to real-world robustness (Dubey et al., 2019; Yin et al., 2019; Richardson & Weiss, 2020), and recently in vision science to understand similar and divergent visual representations with humans (Zhou & Firestone, 2019; Feather et al., 2019; Golan et al., 2019; Reddy et al., 2020; Dapello et al., 2020).

Similarly, we are interested in understanding how training on a specific natural image distribution plays a role in the adversarial sensitivity of a model, which could lead to the construction of more adversarially robust models and a greater understanding of why machines have a divergent visual representation than humans. For example, Ilyas et al. (2019) found that adversarial sensitivity (which

Submitted to 2nd Workshop on Shared Visual Representations in Human and Machine Intelligence (SVRHM).

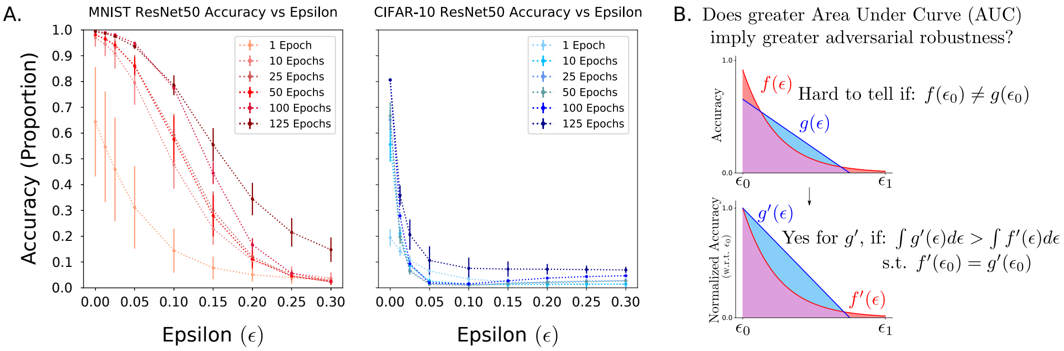

Figure 1: (A) After using the same hyperparameters and training scheme (SGD) for both models, MNIST achieves around 99% accuracy, while CIFAR-10 peaks around 80% with ResNet50 (both without data-augmentation). In cases like these it may be obvious to say that better performing models will be more adversarially robust – but this is not always the case, in some cases it is *the opposite* when fixing the image distribution (Zhang et al., 2019); (B) One solution: example graphs showing the area under the curve of two models before (top) and after (bottom) accuracy normalization. Here we show how at ϵ_0, models go from un-matched accuracy to a matched upper-bounded score of 1, allowing a 'fair' computation of area under the curve.

they refer to as adversarial vulnerability) is not necessarily tied to the training scheme, but rather is a property of the dataset. Similarly, Ding et al. (2019) finds that semantic-preserving shifts on the image distribution could result in drastically different adversarial robustness for adversarially trained models. Ultimately, we are interested in the inherent properties of *natural* image distributions and how they contribute to adversarial sensitivity, rather than how to engineer a robust model via adversarial training, or the impact of manipulating images on a model's adversarial sensitivity. At a higher level, our goal is to understand what it means for a model observer to inherently be more adversarially sensitive to objects vs scenes or objects vs digits, where the latter is addressed in this paper. These experiments may also provide insight to how robust visual representations are learned in the human visual system.

2 Defining a Normalized Adversarial Robustness Metric

Before we begin to make comparisons of how robust or sensitive a model is to adversarial perturbations, we must define a metric of choice for these comparisons. The adversarial robustness R, should be a measure of the rate at which accuracy changes as ϵ (adversarial perturbation strength) increases over a particular ϵ-interval of interest. The faster the accuracy of a model decreases as ϵ increases, the lower the adversarial robustness is for that model. We propose an adaptation to measure adversarial robustness based on the commonly used: area under the curve (AUC). A good measure of how much change is occurring in an ϵ-interval is the AUC of a function that outputs the accuracy given an ϵ for that model. This AUC provides a total measure of model performance for an ϵ-interval. If the accuracy decreases quickly as ϵ increases, then the AUC will be smaller.

However, despite how intuitive as the previous notion may sound, we immediately run into a problem: Some datasets are more *discriminable* than others independent of model observers and even if equalized with the same *chance* baseline (*e.g.* chance for both datasets is 10% if there are 10 equally sampled classes; Figure 1 (A)); How do we take this into account when computing the area under the curve? It could be possible that under un-equal initial performances, one model seems more 'adversarially robust' over the other by virtue purely of the initial offset in the better performance, but is this positive differential the inherent property that drives robustness?

Figure 1 (B) shows one simple solution to solve the differences in accuracy between two model systems is by normalizing them with respect to their accuracy under non-adversarial ($\epsilon_0 = 0$) Fast Gradient Sign Method (FGSM) attacks (Goodfellow et al. (2014)). This yields the following formula:

$$R = \frac{1}{f(\epsilon_0)(\epsilon_1 - \epsilon_0)} \int_{\epsilon_0}^{\epsilon_1} f(\epsilon) d\epsilon \tag{1}$$

which can be interpreted as the bounded area under the curve for a given interval of ϵ-attacks (i.e. $[\epsilon_0, \epsilon_1]$). Note that we must have $f(\epsilon_0) > 0$ and $\epsilon_1 > \epsilon_0$. The term $\frac{1}{f(\epsilon_0)(\epsilon_1 - \epsilon_0)}$ of Eq. 1, normalizes the accuracies and puts the area under the curve of the normalized accuracy between $[0, 1]$. The division by $f(\epsilon_0)$ normalizes the function because the function now represents the change in accuracy with respect to no adversarial perturbations (or whatever ϵ_0 is set as). The accuracy at $f(\epsilon_0)$ can be considered an *'oracle'* for the adversarial attacks of the model (i.e. the likely optimal or best performance for that ϵ-interval).

This metric is only as accurate as the fitted function $f(\epsilon)$, which outputs the accuracy of an adversarial attack for a model given an ϵ. We have two methods to find $f(\epsilon)$: 1) to empirically compute multiple values of ϵ and estimate the normalized area under the curve with the trapezoid method; 2) to find the closed form expression of $f(\circ)$ as one would do for psychometric functions (Wichmann & Hill, 2001) and integrate. In this paper we do the former (compute multiple values of ϵ), although our method is extendable to the latter.

3 Experiments

Given the nature of our question, and the introduction of a simple normalization metric, the following experiments try to provide some answers on the nature of an image distribution and it's relationship to adversiarial robustness via the normalization scheme. All experiments used 3 key networks: LeNet, ResNet50, and a fully connected network (FullyConnectedNet) where we explored adversarial sensitivity over 20 paired network runs and their learning dynamics. It is important to note that *all hyperparameters are held constant*, the only difference between the models using a certain architecture is the datasets they are trained and tested on. All differences that are mentioned are statistically significant using a Welch's t-test with significance level $\alpha = 0.05$. Detailed Methods can be accessed in Appendix A.2.

3.1 Comparing intrinsic adversarial sensitivity: MNIST vs CIFAR-10 Robustness

Are CIFAR-10 trained models inherently more adversarially sensitive than MNIST trained models? We investigate this question by comparing the adversarial robustness of CIFAR-10 models and the MNIST models for the three architectures. Figure 2(A) (top row) shows normalized accuracy graphs for the CIFAR-10 trained models and Figure 2(A) (bottom row) shows graphs of normalized accuracy for MNIST trained models. We find that for both LeNet and FullyConnectedNet, the MNIST models were more adversarially robust than CIFAR-10 models *(i.e. less adversarially sensitive)*, for each epoch we examined. The same pattern of results held for ResNet50 models except for the 1st epoch where there was no difference between the MNIST and CIFAR-10 models.

Result 1: For the three network architectures tested (that all vary in approximation power and architectural constraints), CIFAR-10 trained models are inherently more adversarially sensitive than MNIST models.

3.2 Impact of Pretraining on Out-Of-Distribution (o.o.d) image datasets

Does pretraining on CIFAR-10, then training on MNIST result in a more adversarially robust learned representation of MNIST? Humans are regarded as being adversarially robust (i.e. $R \approx 1$ for some ϵ-interval) and we know that humans learn objects before they learn digits. So can imitating the order that humans learn visual recognition tasks increase adversarial robustness or can CIFAR-10 objects induce a more adversarially sensitive learned representation?

We find that for the FullyConnectedNet the MNIST models were more adversarially robust than the MNIST pretrained on CIFAR-10 model during early stages of learning, but the pretrained models were more robust when examined at 150 and 300 epochs. The MNIST LeNet models were more adversarially robust for all stages of learning than the pretrained model. The pretrained ResNet50 models had no differences in robustness compared to the MNIST ResNet50 models, except for the 1st epoch where the pretrained models were more robust. This result is unexpected as this does not occur for the other architectures. These results would seem to suggest that architecture plays a role in the adversarial sensitivity of the learned representation contingent on the given datasets and potentially compositional nature.

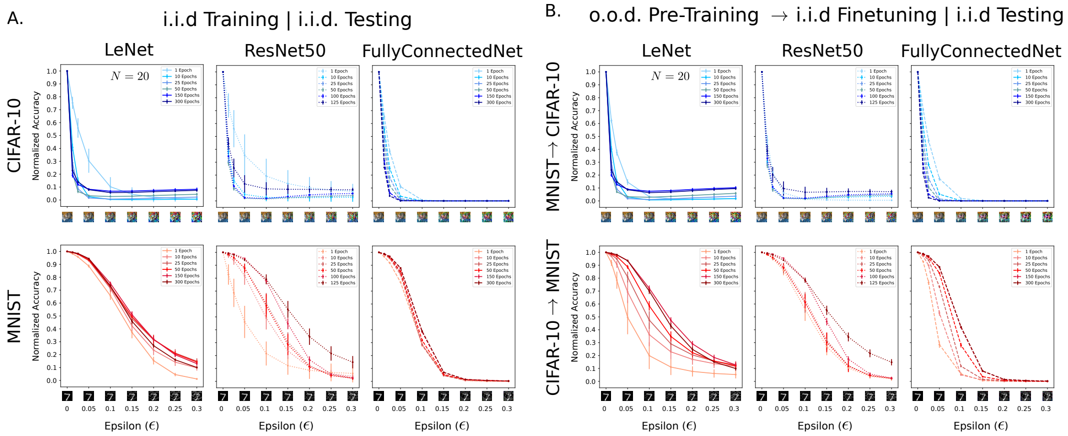

Figure 2: (A) MNIST-trained networks (across all architectures) show greater adversarial robustness after accuracy normalization than CIFAR-10 trained networks. Notice too that ResNet50 appears to be the more adversarially robust network *across* network architectures (LeNet and FullyConnectedNet) independent of learning dynamics; (B) *Perceptual Hysteresis*: FullyConnectedNet and LeNet networks seems to carry over the learned representation and adversarial vulnerability from the pretrained system. However, only LeNet experiences a clear perceptual hysteresis where pretraining on CIFAR-10 for MNIST *is worse* (more adversarially sensitive) than only training on MNIST, yet pretraining on MNIST for CIFAR-10 *is better* (more adversarially robust) than only training on CIFAR-10 (See Appendix A.7).

We found that pretraining on CIFAR-10 and then training on MNIST generally does not lead to more adversarially robust models. Conversely, does pretraining on MNIST and then training on CIFAR-10 have this same effect? We find that this is not always the case. Pretraining on MNIST then training on CIFAR-10 led to marginal improvements in adversarial robustness for LeNet, except for the 1st epoch (Figure 2 (B) (top row)). For ResNet50, pretraining resulted in more adversarially sensitive models at the start and end of training (1 and 125 epochs), otherwise there was no difference compared to not pretraining. The FullyConnectedNet pretrained models were more adversarially robust in earlier stages of learning, but were less robust in later stages. Tables of the robustness metrics for the CIFAR-10 models pretrained on MNIST (as well as for other experiments) can be found in Appendix A.7. This finding also requires further investigation.

Result 2: Therefore, pretraining on CIFAR-10, then training on MNIST does not generally result in a more adversarially robust model than training on MNIST alone using the FullyConnectedNet, LeNet, ResNet50 architectures. This is a counter-intuitive result as one would have expected that – like humans – seeing objects first may bring positive impacts to learned representations (Janini & Konkle, 2019) and robustness. On the other hand, pretraining on MNIST, then training on CIFAR-10 only aided LeNet; for FullyConnectedNet it helped in earlier stages of learning, while decreased robustness later. Generally, however, ResNet50 models were not affected in terms of carried-over robustness at any intermediate stages of learning. Investigating the origins of this perceptual hysteresis (Sadr & Sinha, 2004) and how it may relate to shape/texture bias (Geirhos et al., 2018; Hermann & Kornblith, 2019), spatial frequency sensitivity (Dapello et al., 2020; Deza & Konkle, 2020), or common perturbations (Hendrycks & Dietterich, 2019) is a subject of on-going work.

3.3 Approximate image-statistics matching and re-evaluating robustness

Our previous results suggested that after taking into account different measures of accuracy normalization, the MNIST dataset is intrinsically more adversarially robust (*i.e.* less adversarially sensitive) than CIFAR-10. This implies that it is harder to fool a system performing digit recognition, than a system performing object detection, likely due to the fact that number digits are highly selective to shape, and show less perceptual variance than objects.

Naturally, the next question that arises is if the image complexity itself is somehow making each perceptual system more adversarially sensitive. To test this hypothesis we created a new hybrid

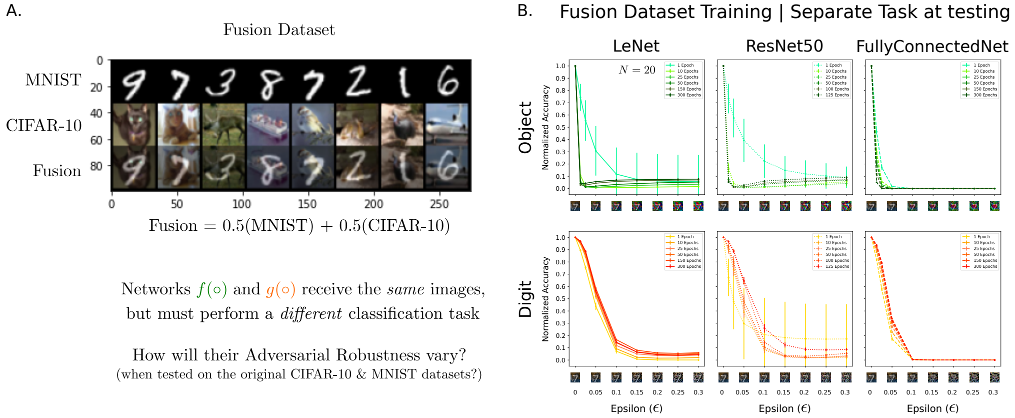

Figure 3: A. The experimental setup and motivation of the Fusion Dataset; B. Graphs of the normalized accuracy of the Fusion dataset on the object recognition task (top row) and digit recognition task (bottom row) using LeNet, ResNet50, and FullyConnectedNet. Generally, models trained on the digit task were more adversarially robust than those trained on the object task, showing the role that task plays in the adversarial sensitivity of a model. Additionally, these models were generally more adversarially sensitive than their MNIST and CIFAR-10 model counterparts.

dataset: a fusion of the MNIST digits α-blended with the CIFAR-10 images. Models were trained to perform either digit recognition or object recognition on these fusion images – thus we have fixed the image distribution but varied the approximation task (Deza et al., 2020). These images were created by scaling the pixels in each CIFAR-10 and MNIST image by 0.5 and adding them together (similar to *Texture shiftMNIST* from Jacobsen et al. (2018), and see Figure 3(A)). For more on this dataset, see Appendix A.4. The goal with this new hybrid dataset is to re-run the same set of previous experiments and test adversarial robustness for both the digit recognition task and the object recognition task and *test* if image complexity alone is the *sole culprit* of adversarial sensitivity.

Result 3: Figure 3 (B) contains the normalized curves of the results for the digit and object recognition tasks on the fusion dataset for each of the architectures. The FullyConnectedNet (all epochs), ResNet50 and LeNet fusion image models were more adversarially robust on the digit recognition task than the object recognition task for all epochs examined excluding the first epoch. This suggests that even if the image distribution is equalized at training, the representation learned varies given the task, and impacts adversarial sensitivity differently. Additional results of comparing the three architectures trained on the Fusion Dataset vs their regular image-distribution trained models (Appendix A.4) show that increasing the image complexity (by adding a conflicting image with the hope of increasing invariance) in fact *decreases* adversarial robustness when compared to regularly trained networks – a result also observed in Jacobsen et al. (2018).

4 Discussion

Our first step in this work verified that the image distribution can impact adversarial sensitivity of a model, our next step is to investigate why, and what factors play a role (Jacobsen et al., 2018; Ding et al., 2019; Ilyas et al., 2019). It is likely that MNIST trained networks are *intrinsically* more adversarially robust than CIFAR-10 trained networks due to the lower subspace in which they live in given their image structure (Hénaff et al., 2014) compared to CIFAR-10. Indeed, we have only scratched the surface of this question by comparing two well known candidate datasets over their learning dynamics: MNIST and CIFAR-10, and continuing this line of work onto exploring the role of the image distribution on adversarial sensitivity for texture or scenes is a promising next step. Finally, future experiments should continue to investigate the effect of the learning objective on the learned representation induced from the image distribution. We have already seen how the task affects the adversarial sensitivity of a model even when image statistics are matched under a supervised training paradigm. With the advent of self-supervised (Konkle & Alvarez, 2020) and unsupervised (Zhuang et al., 2020) objectives that may be predictive of human visual coding, it may be relevant to investigate the changes in adversarial sensitivity for the current (objects, digits) and new (texture, scenes) image distributions with our normalized metric for these new learning objectives.

References

Joel Dapello, Tiago Marques, Martin Schrimpf, Franziska Geiger, David D Cox, and James J DiCarlo. Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. *BioRxiv*, 2020.

Arturo Deza and Talia Konkle. Emergent properties of foveated perceptual systems. *arXiv preprint arXiv:2006.07991*, 2020.

Arturo Deza, Qianli Liao, Andrzej Banburski, and Tomaso Poggio. Hierarchically local tasks and deep convolutional networks. *arXiv preprint arXiv:2006.13915*, 2020.

Gavin Weiguang Ding, Kry Yik-Chau Lui, Xiaomeng Jin, Luyu Wang, and Ruitong Huang. On the sensitivity of adversarial robustness to input data distributions. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=S1xNEhR9KX`.

Abhimanyu Dubey, Laurens van der Maaten, Zeki Yalniz, Yixuan Li, and Dhruv Mahajan. Defense against adversarial images using web-scale nearest-neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8767–8776, 2019.

Jenelle Feather, Alex Durango, Ray Gonzalez, and Josh McDermott. Metamers of neural networks reveal divergence from human perceptual systems. In *Advances in Neural Information Processing Systems*, pp. 10078–10089, 2019.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

Tal Golan, Prashant C Raju, and Nikolaus Kriegeskorte. Controversial stimuli: pitting neural networks against each other as models of human recognition. *arXiv preprint arXiv:1911.09288*, 2019.

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

David Marvin Green, John A Swets, et al. *Signal detection theory and psychophysics*, volume 1. Wiley New York, 1966.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Olivier J Hénaff, Johannes Ballé, Neil C Rabinowitz, and Eero P Simoncelli. The local low-dimensionality of natural images. *arXiv preprint arXiv:1412.6626*, 2014.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HJz6tiCqYm`.

Katherine L Hermann and Simon Kornblith. Exploring the origins and prevalence of texture bias in convolutional neural networks. *arXiv preprint arXiv:1911.09071*, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pp. 125–136, 2019.

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

Daniel Janini and Talia Konkle. Shape features learned for object classification can predict behavioral discrimination of written symbols. *Journal of Vision*, 19(10):32d–32d, 2019.

Talia Konkle and George A Alvarez. Instance-level contrastive learning yields human brain-like representation without category-supervision. *bioRxiv*, 2020.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541. URL `https://doi.org/10.1162/neco.1989.1.4.541`.

Jeffrey Lubin. A human vision system model for objective picture quality measurements. In *IEE conference publication*, volume 1, pp. 498–503. IET, 1997.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

Hrushikesh N Mhaskar and Tomaso Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(06):829–848, 2016.

Behnam Neyshabur. Towards learning convolutions from scratch. *arXiv preprint arXiv:2007.13657*, 2020.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

Manish V. Reddy, Andrzej Banburski, Nishka Pant, and Tomaso Poggio. Biologically inspired mechanisms for adversarial robustness, 2020.

Eitan Richardson and Yair Weiss. A bayes-optimal view on adversarial examples. *arXiv preprint arXiv:2002.08859*, 2020.

Javid Sadr and Pawan Sinha. Object recognition and random image structure evolution. *Cognitive Science*, 28(2):259–287, 2004.

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

Felix A Wichmann and N Jeremy Hill. The psychometric function: I. fitting, sampling, and goodness of fit. *Perception & psychophysics*, 63(8):1293–1313, 2001.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems*, pp. 13276–13286, 2019.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

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.

Zhenglong Zhou and Chaz Firestone. Humans can decipher adversarial images. *Nature communications*, 10(1):1–9, 2019.

Chengxu Zhuang, Siming Yan, Aran Nayebi, Martin Schrimpf, Michael Frank, James DiCarlo, and Daniel Yamins. Unsupervised neural network models of the ventral visual stream. *bioRxiv*, 2020.

A Appendix

A.1 Normalization Score (Extended)

Picking ϵ_0 and ϵ_1 is an experimental choice. Choosing $\epsilon_0 = 0$ allows you to measure adversarial robustness starting from no perturbations, yet we can also have that $\epsilon_0 > 0$. An example of a good choice for ϵ_1 is the result of psychophysical experiments determining for what ϵ can humans no longer recognize the perturbed images, or a reference point up until any one of the models reaches it's chance performance. For too high a choice of ϵ_1, the image can saturate and the performance will likely approach chance, *e.g.* this rebounding effect can be seen in some of the CIFAR-10 curves.

There are certain assumptions for this normalization scheme to hold. For example, in both of our experiments MNIST and CIFAR-10 are equalized to have 10 classes and we assume an independent and identically distributed testing distribution such that chance performance for any model observers is the same at 10%. One could see how the normalization scheme would give a misleading result if one dataset has 2 i.i.d classes that yield 50% chance and another dataset yields 10 i.i.d classes that yield 10% chance. Propotion corrects are not comparable and a more principled way of equalizing performance – likely using d' would be needed (Green et al., 1966).

Overall, this robustness metric can be used to assert whether a model is adversarially robust over a particular ϵ-interval or to measure how adversarially robust a model is comparatively to other models over that interval. If for a particular model, $R = 1$, this implies for all ϵ in the ϵ-interval, the model classifies the perturbed images correctly. If for a model, $R = 0$, that means that for all ϵ in the interval, the model classifies the perturbed images incorrectly. Since a model has low adversarial sensitivity if and only if it has high adversarial robustness, the metric for adversarial sensitivity $S = 1 - R$.

A.2 Methods

A.2.1 Architectures

Three network architectures were considered: FullyConnectedNet, LeNet (LeCun et al. (1989)), and ResNet50 (He et al. (2015)). FullyConnectedNet is a fully connected artificial neural network that has 1 hidden layer with 7500 hidden units. 7500 hidden units were chosen so the number of parameters for the FullyConnectedNet has the same magnitude of parameters as ResNet50. 1 hidden layer was chosen so the network is not as hierarchical as a convolutional neural network (See Mhaskar & Poggio (2016); Poggio et al. (2017) and recently Neyshabur (2020); Deza et al. (2020)).

A.2.2 Datasets

The datasets used were MNIST, CIFAR-10, and a Fusion Dataset. For more on the Fusion dataset, see Appendix A.4. To use the exact same architectures with the datasets, MNIST was upscaled to 32×32 and converted to 3 channels to match the dimensions of CIFAR-10 (i.e. $32 \times 32 \times 3$).

A.2.3 Hyperparameters, Optimization Scheme, and Initialization

It is important to note that *all hyperparameters other than dataset for a model are held constant*, the only difference between the models for a particular architecture is the datasets they are trained and tested on. The loss function used for all models was cross-entropy loss and the optimizer used was stochastic gradient descent (SGD) with weight decay of 5×10^{-4}, momentum of 0.9, and with an initial learning rate of 0.01 for the FullyConnectedNet and LeNet models and an initial learning rate of 0.1 for the ResNet50 models. The learning rate was divided by 10 at 50% of the training. The FullyConnectedNet and LeNet models were trained to 300 epochs and the ResNet50 models were trained to 125 epochs. A batch size of 125 was used. See Appendix A.4 for why this batch size was used. We chose these hyperparameters and optimization scheme since we found they had best results in preliminary experiments.

For all experiments, each model was trained 20 times with matched initial random weights across different datasets. For example for LeNet, 20 different LeNet models all with different initial random weights: $\{w_1, w_2, ..., w_{20}\}$ were used to train for CIFAR-10 in our first experiment, and these same initial random weights were used to train for MNIST. This removed the variance induced by a particular initialization (*e.g.* a lucky/unlucky noise seed) that could bias the comparisons by arriving to a better solution via SGD. This procedure was tractable because our MNIST dataset was resized to a 3-channeled version with a new size of $32 \times 32 \times 3$ instead of $28 \times 28 \times 1$ (original MNIST).

A.2.4 Adversarial Attacks

The method used for generating adversarial images is the Fast Gradient Sign Method (FSGM) presented in Goodfellow et al. (2014). FGSM was chosen over Projected Gradient Descent (PGD) (Madry et al. (2019)) based on preliminary results as FGSM was sufficient to successfully adversarially attack the model (indeed we did not want to perform adversarial training or other data-augmentations schemes that may bias our results). FGSM also has a lower computational cost than PGD allowing us to run more experiments. Overall, we were interested in creating images that cause the model to misclassify in general, rather than misclassifying an image to a particular class.

The ϵ-interval used here is $[0, 0.3]$ (i.e. $\epsilon_0 = 0, \epsilon_1 = 0.3$). 0.3 was chosen as the upper bound because adversarial images at that magnitude are very difficult for many classifiers to classify correctly. The models were adversarially attacked at different stages of learning. The trained models were adversarially attacked with ϵ values of 0, 0.0125, 0.025, 0.05, 0.1, 0.15, 0.2, 0.25, and 0.3 to create (ϵ). For the models using LeNet and FullyConnectedNet architectures, they were adversarially attacked at 1, 10, 25, 50, 150, and 300 epochs. Models using the ResNet50 architecture were adversarially attacked at 1, 10, 25, 50, 100, and 125 epochs.

A.3 Pretraining (Extended)

This pretraining procedure was done by using the existing CIFAR-10 or MNIST FullyConnectedNet, LeNet, and ResNet50 models as bases and then training/fine-tuning them using the same training scheme but with MNIST or CIFAR-10 respectively.

For the ResNet50, LeNet, and FullyConnectedNet architectures, the models pretrained on CIFAR-10 then trained on MNIST were statistically significantly more adversarially robust than models pretrained on MNIST then trained on CIFAR-10 for all epochs examined.

A.4 Fusion Dataset

Each fusion image can be concisely summarized with the following α-blending procedure as follows:

$$F = 0.5M + 0.5C, \tag{2}$$

where F is a new fusion image, M is an MNIST image modified to 32x32x3 (by upscaling and increasing number of color channels), and C is a CIFAR-10 image. Example fusion images can be found in Figure 3(A) and Figure 6(C).

The fusion dataset was created online during training or testing during each mini-batch by formula 2. The fusion image training set was constructed using the MNIST and CIFAR-10 training set and the fusion image test set was constructed using the MNIST and CIFAR-10 test set. During training, the MNIST and CIFAR-10 datasets are shuffled at the start of every epoch. Therefore, it is likely that no fusion images are shown to the model more than once. This was done to ensure that the model cannot learn any correlation between any CIFAR-10 object and any MNIST digit, as well as, improve generalization of the model.

The batch size was 125 since this is the closest number to a more typical batch size of 128 that divides both the number of CIFAR-10 images and MNIST images. This was needed to ensure that the batches align properly when creating the fusion images.

Observation: When examining the first epoch for the fusion trained models, we find that the standard deviation is generally high. This is likely due to our choice of avoiding to show the same fusion image twice. This does not occur in later stages of training.

Comparing fusion image models trained on the digit task and MNIST models: for the FullyConnectedNet and LeNet architecture, the MNIST models were more robust. The same holds for the ResNet50 MNIST models except at epoch 1, where there were no difference. CIFAR-10 models using the FullyConnectedNet architecture were more adversarially robust than the fusion image models trained on the object recognition task for all epochs tested. The same was true for the LeNet and ResNet50 architectures except there were no differences between CIFAR-10 models and fusion images with object task in adversarial robustness for 1 and 50 epochs.

A.5 Un-normalized Adversarial Robustness curves (Raw-Data)

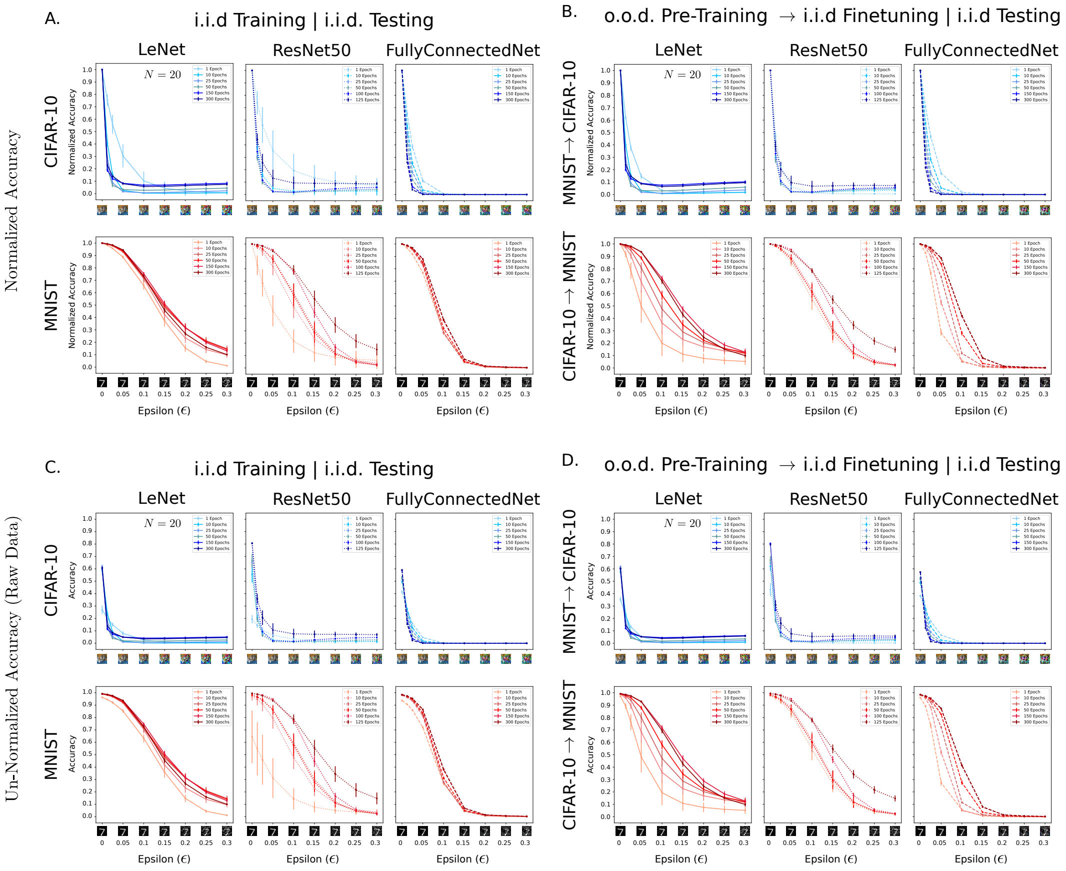

Figure 4: (A,B): Redrawn graphs from Figure 2; (C,D): The un-normalized adversarial robustness trade-off curves for each network (LeNet, ResNet50, FullyConnectedNet) and dataset (MNIST and CIFAR-10) for the FGSM-based Attack (Goodfellow et al., 2014).

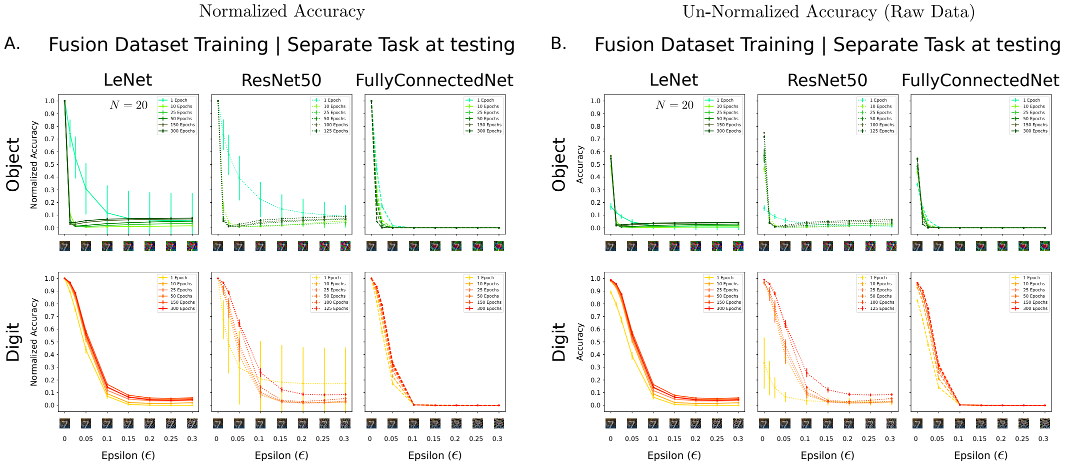

Figure 5: (A): Redrawn graphs from Figure 3; (B): The un-normalized adversarial robustness trade-off curves for each network (LeNet, ResNet50, FullyConnectedNet) and Fusion dataset for the FGSM-based Attack (Goodfellow et al., 2014).

A.6 Zoomed in Sample Adversarial Stimuli

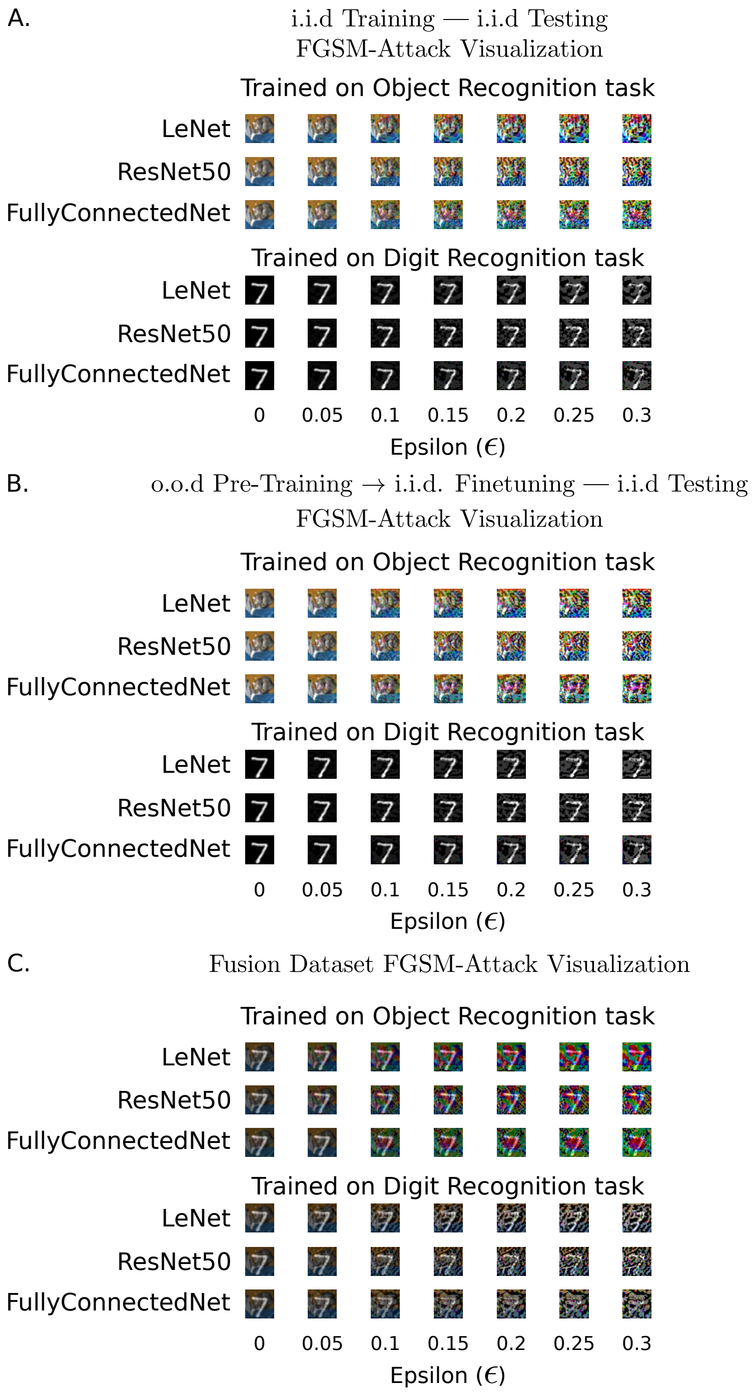
A.
i.i.d Training — i.i.d Testing
FGSM-Attack Visualization
Trained on Object Recognition task
LeNet
ResNet50
FullyConnectedNet
Trained on Digit Recognition task
LeNet
ResNet50
FullyConnectedNet
0
0.05
0.1
0.15
0.2
0.25
0.3
Epsilon (ϵ)
B.
o.o.d Pre-Training → i.i.d. Finetuning — i.i.d Testing
FGSM-Attack Visualization
Trained on Object Recognition task
LeNet
ResNet50
FullyConnectedNet
Trained on Digit Recognition task
LeNet
ResNet50
FullyConnectedNet
0
0.05
0.1
0.15
0.2
0.25
0.3
Epsilon (ϵ)
C.
Fusion Dataset FGSM-Attack Visualization
Trained on Object Recognition task
LeNet
ResNet50
FullyConnectedNet
Trained on Digit Recognition task
LeNet
ResNet50
FullyConnectedNet
0
0.05
0.1
0.15
0.2
0.25
0.3
Epsilon (ϵ)

Figure 6: Zoomed in versions of the adversarial patches created after an FGSM-Attack. Shown images are the perturbed stimuli for the networks at the 300-*th*,125-*th*,300-*th* epoch for LeNet, ResNet50 and FullyConnectedNet respectively. A) Stimuli from our first experiment; B) Stimuli from our second experiment; C) Stimuli from our third experiment, notice how color bleeds (stronger for objects) into both stimuli when trained on the same dataset but are imposed a different task. Interestingly, the differences in the adversarial noise pattern are more salient across architectures for the Fusion Dataset.

A.7 Statistical Testing of Results (Extended)

Table Legend:
⋆: Denotes statistically significantly higher adversarial robustness for MNIST vs CIFAR-10
•: Denotes statistically significantly higher adversarial robustness for a pretrained MNIST model vs a pretrained CIFAR-10 model
⋄: Denotes statistically significantly higher adversarial robustness for a pretrained model vs non-pretrained model both trained on same dataset
†: Denotes statistically significantly higher adversarial robustness for Fusion digit task vs Fusion object task
‡: Denotes statistically significantly higher adversarial robustness for MNIST vs Fusion digit task or CIFAR-10 vs Fusion object task

Table 1: MNIST LeNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.438451⋆ ⋄ ‡	0.020373
10	0.499361⋆ ⋄ ‡	0.023713
25	0.539882⋆ ⋄ ‡	0.025632
50	0.543602⋆ ⋄ ‡	0.019689
150	0.549709⋆ ⋄ ‡	0.013361
300	0.520547⋆ ⋄ ‡	0.010607

Table 2: CIFAR-10 LeNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.155815⋄	0.048486
10	0.053812‡	0.003744
25	0.050288‡	0.00447
50	0.063126	0.009296
150	0.102256‡	0.009832
300	0.097237‡	0.008319

Table 3: Pretrained on CIFAR-10 trained on MNIST LeNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.249363•	0.05819
10	0.364303•	0.049268
25	0.420838•	0.039604
50	0.462459•	0.022959
150	0.529665•	0.013517
300	0.503499•	0.013186

Table 4: Pretrained on MNIST trained on CIFAR-10 LeNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.102636	0.008461
10	0.059423⋄	0.005675
25	0.057364⋄	0.008041
50	0.069428⋄	0.006776
150	0.113739⋄	0.007872
300	0.109077⋄	0.008987

Table 5: Fusion digit task LeNet adversarial robustness

Epoch:	Mean Robustness:	STD:
1	0.173351	0.008095
10	0.204077†	0.006719
25	0.225057†	0.008112
50	0.238905†	0.0096
150	0.254536†	0.005374
300	0.239281†	0.004036

Table 6: Fusion object task LeNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.181474	0.1951
10	0.035735	0.007395
25	0.046072	0.005877
50	0.058456	0.006017
150	0.077806	0.005403
300	0.087175	0.00396

Table 7: MNIST ResNet50 adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.243206	0.064093
10	0.37482⋆‡	0.040255
25	0.413437⋆‡	0.037413
50	0.405367⋆‡	0.03901
100	0.486548⋆‡	0.015996
125	0.572381⋆‡	0.03999

Table 8: CIFAR-10 ResNet50 adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.222479⋄	0.098118
10	0.072652‡	0.027665
25	0.06675‡	0.010177
50	0.062354	0.007254
100	0.075903 ‡	0.00379
125	0.138936⋄‡	0.038124

Table 9: Pretrained on CIFAR-10 trained on MNIST ResNet50 adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.392197•⋄	0.036562
10	0.422000•	0.023966
25	0.414428•	0.035434
50	0.413840•	0.034093
100	0.491886•	0.012084
125	0.569465•	0.020692

Table 10: Pretrained on MNIST trained on CIFAR-10 ResNet50 adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.067825	0.008714
10	0.063501	0.0099
25	0.068075	0.012437
50	0.064227	0.010571
100	0.074815	0.004281
125	0.114783	0.03279

Table 11: Fusion digit task ResNet50 adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.252273	0.267929
10	0.180888†	0.014562
25	0.190701†	0.022835
50	0.202564†	0.009819
100	0.242128†	0.006506
125	0.296214†	0.018022

Table 12: Fusion object task ResNet50 adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.243001	0.10571
10	0.053983	0.014983
25	0.045387	0.008189
50	0.065714	0.017707
100	0.070246	0.00476
125	0.084441	0.007446

Table 13: MNIST FullyConnectedNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.272342⋆ ⋄ ‡	0.004514
10	0.280533⋆ ⋄ ‡	0.002443
25	0.284781⋆ ⋄ ‡	0.002756
50	0.29422⋆ ⋄ ‡	0.002926
150	0.311588⋆‡	0.000868
300	0.312896⋆‡	0.00083

Table 14: CIFAR-10 FullyConnectedNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.086983‡	0.003529
10	0.064134‡	0.002695
25	0.054232‡	0.00201
50	0.045977⋄‡	0.001153
150	0.037929⋄‡	0.000255
300	0.034145⋄‡	0.000274

Table 15: Pretrained on CIFAR-10 trained on MNIST FullyConnectedNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.153764●	0.005824
10	0.194182●	0.00334
25	0.235387●	0.002786
50	0.281417●	0.004191
150	0.320011●⋄	0.001317
300	0.321303●⋄	0.001126

Table 16: Pretrained on MNIST trained on CIFAR-10 FullyConnectedNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.104627⋄	0.004495
10	0.069091⋄	0.002008
25	0.053113	0.002535
50	0.044058	0.001362
150	0.036256	0.00035
300	0.031516	0.000222

Table 17: Fusion digit task FullyConnectedNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.113213†	0.003298
10	0.129283†	0.001951
25	0.139871†	0.001997
50	0.145459†	0.001856
150	0.15254†	0.000776
300	0.15016†	0.000758

Table 18: Fusion object task FullyConnectedNet adversarial robustness

Epoch:	Mean Robustness:	SD:
1	0.052986	0.002406
10	0.036917	0.00154
25	0.03305	0.001239
50	0.030376	0.001014
150	0.026116	0.00027
300	0.022955	0.000141