Scoring Black-Box Models for Adversarial Robustness

Jian Vora [*1] **Pranay Reddy Samala** [*1]

Abstract

Deep neural networks are susceptible to adversarial inputs and various methods have been proposed to defend these models against adversarial attacks under different perturbation models. The robustness of models to adversarial attacks has been analyzed by first constructing adversarial inputs for the model, and then testing the model performance on the constructed adversarial inputs. Most of these attacks require the model to be white-box, need access to data labels and finding adversarial inputs can be computationally expensive. We propose a simple scoring method for black-box models which indicates their robustness to adversarial input. We show that adversarially more robust models have a smaller l_1-norm of LIME weights and sharper explanations.

1. Introduction

Deep neural networks have shown impressive performance on a variety of tasks in domains such as vision (Dosovitskiy et al., 2020; Ramesh et al., 2021), natural language (Brown et al., 2020), speech (Hsu et al., 2021; Baevski et al., 2020). As and when we start deploying these models in the real world, it becomes important that these models are truly robust and reliable. One way is to have *model cards* [1](Mitchell et al., 2019) associated with models which not only include their downstream performance metric but also important aspects such as robustness (Goodfellow et al., 2015), fairness (Mehrabi et al., 2021), training datasets, etc. Such model cards would provide practitioners a much better insight into model selection for a particular application leading to the safer deployment of these models in the real world. Among other issues with current deep learning models, one crucial concern while deploying these models is their brittleness to specially designed adversarial inputs that fool them. In this work, we aim at scoring black-box models based on their robustness to adversarial samples. We score models based on the explanations they generate which leads to an interesting connection between robustness and explainability of models. In the upcoming subsections, we provide a brief overview of adversarial robustness and input attribution-based explanation methods for machine learning models.

1.1. Adversarial Robustness

Current deep learning models have been shown to be vulnerable to human-imperceptible adversarial perturbations which significantly degrade the model's performance. This is not desirable, especially in safety-critical applications such as autonomous driving (Xu et al., 2022) and medical applications (Finlayson et al., 2018). Most classes of adversarial attacks try to add small perturbations to the input so as to move the input sample across a decision boundary, thus changing the model's prediction. Various types of attacks have been tried to target classifiers followed by various defense mechanisms as well. A common theme for image-classifier based attack methods is to find a perturbation $\boldsymbol{\delta}$ to be added to an input $\mathbf{x} \in \mathbb{R}^D$ such that $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\delta}$ fools the classifier. Different constraints of the perturbation $\boldsymbol{\delta}$ lead to different types of attacks, the most common of which is trying to bound some l_p norm of $\boldsymbol{\delta}$. Concretely, if L is the loss and $\mathcal{M}$ is the the neural network, all these attacks try to find the perturbation $\boldsymbol{\delta}$ by solving the following optimization problem,

$$\max_{\boldsymbol{\delta}} L(\mathcal{M}(\mathbf{x} + \boldsymbol{\delta})) \text{ s.t. } \|\boldsymbol{\delta}\|_p \leq \epsilon \tag{1}$$

White-box attacks are the most common ones in which the adversary has complete access to the model parameters and details of any defense mechanism. Some common white-box attacks include Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015), Carlini & Wagner (Carlini & Wagner, 2017), DeepFool (Moosavi-Dezfooli et al., 2016), L-BFGS (Szegedy et al., 2013), JSMA (Papernot et al., 2016a). The mainstream approach to defend against these attacks is by training on adversarial examples (Goodfellow et al., 2015; Kurakin et al., 2017; Madry et al., 2018; Shaham et al., 2018; Na et al., 2018; Tramèr et al., 2018). Other methods include defense distillation (Papernot et al.,

[1]Department of Computer Science, Stanford University. Correspondence to: Jian Vora <jianv@stanford.edu>.

2^{nd} *AdvML Frontiers workshop at* 40^{th} *International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1]https://modelcards.withgoogle.com/

2016b), and input transformations to make inputs closer to the training distribution (e.g. via randomization or generative modelling) (Zhang & Liang, 2019; Samangouei et al., 2018; Yoon et al., 2021; Nie et al., 2022; Song et al., 2018).

1.2. Explainability

Many methods have been proposed that try to provide post-hoc explanations of predictions from a black-box machine learning model (Ribeiro et al., 2016; Lundberg & Lee, 2017). For a given input sample, all these methods try to attribute parts of the input that led to a particular prediction from the model. In this work, we utilize LIME (Ribeiro et al., 2016), a popular method for explanations. Linear functions are inherently explainable with the coefficients giving the relative importance of a feature attribute in predicting the outcome. LIME uses this fact to approximate a model as a linear function of the input around the query data points.

Concretely, given a black-box model $\mathcal{M}$ trained on input samples $\{\mathbf{x}_i\}_{i=1}^N$ with $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ generating predictions $\hat{y}_i = \mathcal{M}(\mathbf{x}_i)$, LIME does the following to generate explanations for query sample $\mathbf{x_i}$:

1. Sample k points near $\mathbf{x}_i$ and let them be called $\{\mathbf{z}_{i,j}\}_{j=1}^k$
2. Fit a linear model $\mathbf{w}_i^T\mathbf{z}_i(\mathbf{w}_i \in \mathbb{R}^d)$ on a new dataset $\{\mathbf{z}_{i,j}, \mathcal{M}(\mathbf{z}_{i,j})\}_{j=1}^k$

We refer the reader to (Ribeiro et al., 2016) for different methods of sampling near $\mathbf{x}_i$ for Step 1. These methods mainly depend on the domain where $\mathbf{x}_i$ belongs (for example images, text, etc.). Step 2 is simply performing linear regression on the sampled points around $\mathbf{x}_i$ and predictions from the model $\mathcal{M}$ on those points. The learnt weights $\mathbf{w}_i \in \mathbb{R}^d$ can be used to explain the input sample $\mathbf{x}_i$. These weights can be thought of as approximating the model M with a linear model around $\mathbf{x}_i$ and indicate how the model predictions will change given small changes in $\mathbf{x}_i$. In the subsequent sections, we shall denote these linear weights $\mathbf{w}_i$ as $\text{LIME}(\mathcal{M}(\mathbf{x}_i))$. An important point to note here is that computing $\text{LIME}(\mathcal{M}(\mathbf{x}_i))$ does not require access to the model weights and can be done for any black-box model $\mathcal{M}$.

2. Method

In this section, we shall define our scoring method for any black-box model based on the adversarial robustness of the model. Typically, robustness is measured by first generating adversarial inputs to the model and then measuring the performance of the models on the generated adversarial inputs. For a model $\mathcal{M}$, we shall denote $\texttt{robust-acc}(\mathcal{M})$ as the accuracy of model $\mathcal{M}$ on adversarial inputs generated by one of the perturbation models described in Section 1.1. Most common adversarial attacks are white-box, i.e. they need the exact model weights to generated adversarial inputs. Given a perturbation model and two models $\mathcal{M}_1$ and $\mathcal{M}_2$, we call $\mathcal{M}_1$ to be more robust then $\mathcal{M}_2$ *iff* $\texttt{robust-acc}(\mathcal{M}_1) > \texttt{robust-acc}(\mathcal{M}_2)$.

We note that the above way of comparing methods for robustness has certain issues:

1. Most methods which generate adversarial inputs require model weights. However, we note that many current models are available as usable APIs which makes them black-box. It is hard to find to adversarial inputs for black-box models which makes it hard to compute `robust-acc` and subsequently compare models for robustness.
2. To compute `robust-acc`, we need access to labels of the input samples to compare with model predictions which might not always be available.
3. Even for white-box models, computing `robust-acc` is computationally expensive. Finding a single adversarial sample requires performing several steps of gradient ascent on the input space to solve Eq. 1.

We propose a scoring method which does not have the above limitations. The method works for any black-box model off-the-shelf, does not require finding adversarial samples and works for any unlabeled dataset. Under these constraints, we validate that our scoring method correlates well with `robust-acc`. More formally, the only information we have access to is an unlabelled dataset $\mathcal{X}$, a trained black-box model $\mathcal{M}$.

Let $\mathcal{X}$ be the space of inputs and $\mathcal{Y}$ be the output space. $\mathcal{M}_1$ and $\mathcal{M}_2$ are two black-box models mapping $\mathcal{X} \to \mathcal{Y}$. Let $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n$ denote n sample points drawn from $\mathcal{X}$ and we do not have associated labels y for any of them. Denoting lime weights for model input $\mathbf{x}$ by $\text{LIME}(\mathcal{M}(\mathbf{x}))$, input dataset size by n, and data dimensionality d, we propose,

$$\texttt{brittle-score}_{\mathcal{X}}(\mathcal{M}) = \frac{1}{nd}\sum_{\mathbf{x}_i \in \mathcal{X}} \|\text{LIME}(\mathcal{M}(\mathbf{x}_i))\|_1 \tag{2}$$

where $\|.\|_1$ denotes the vector l_1 norm. `brittle-score` indicates the model's *'brittleness'* to adversarial inputs and hence is inversely related to `robust-acc`. Note that the above score only depends on the black-box model $\mathcal{M}$ and input dataset $\mathcal{X}$. Our *hypothesis* is that,

$$\mathcal{M}_1 \text{ is more robust than } \mathcal{M}_2 \Leftrightarrow \texttt{robust-acc}(\mathcal{M}_1) > \texttt{robust-acc}(\mathcal{M}_2)$$
$$\Leftrightarrow$$
$$\texttt{brittle-score}(\mathcal{M}_1) < \texttt{brittle-score}(\mathcal{M}_2)$$

The intuition is simple – more robust models should have the property that small changes in input do not affect the output a lot. Let $\nabla_{\mathbf{x}}\mathcal{M}(\mathbf{x})$ denote the gradient of the model output with respect to the input $\mathbf{x}$. We expect that the adversarially robust models will have a lower norm of $\nabla_{\mathbf{x}}\mathcal{M}(\mathbf{x})$. As the models are black-box, we cannot compute gradients with respect to the input and hence use $\textsc{Lime}(\mathcal{M}(\mathbf{x})) \approx \nabla_{\mathbf{x}}M(\mathbf{x})$ as a proxy. The intuition behind this is similar to prior work that explains adversarial examples for linear models by Goodfellow et al. (Goodfellow et al., 2015).

Note that the value of `brittle-score` as defined in Eq. 2 is simply the magnitude of the lime weights averaged over n points, hence this value is not directly interpretable (unlike `robust-acc`). However, the real purpose of this score is to *compare* the adversarial robustness across two black-box models. Also note that the above scoring method is specific to a dataset $\mathcal{X}$ and should not be compared across models trained on different datasets. In the subsequent section, we present experimental results which show that `brittle-score` is a very good indicator of the robustness of a model and correlates inversely with `robust-acc`.

3. Experiments

In this section, we demonstrate how adversarial training affects both `robust-acc` and `brittle-score`. The main takeaway from the results presented in this Section is that `brittle-score` is inversely correlated to `robust-acc` for a variety of model architectures and attacks and hence makes it an attractive scoring method for black-box models.

We first define some controlled experiments on MNIST (LECUN, 1999) dataset. Our first set of experiments involves adversarial training over different model architectures. We experiment with three different model architectures – 2-layer MLP, a 3-layer CNN and a 5-layer CNN. For each of these architectures, we train two models:

1. One without adversarial training trained on samples from the data distribution using a cross-entropy loss with learning rate 10^{-3} and a batch size of 32.
2. One with adversarial training where the samples were generated by attacking each model with PGD (Madry et al., 2018) attack under l_∞ constraint ($p = \infty$ in Eq. 1) of $\epsilon = 8/255$ for $\boldsymbol{\delta}$.

In Figure 1, we plot the `brittle-score` (Eq. 2) computed on $n = 1000$ samples for the each of the models. These samples were drawn from the data distribution (validation split in this case) and were not *attack samples*. We observe a significant difference in `brittle-score` between robust and non-robust models with the robust models having a lower value of `brittle-score`.

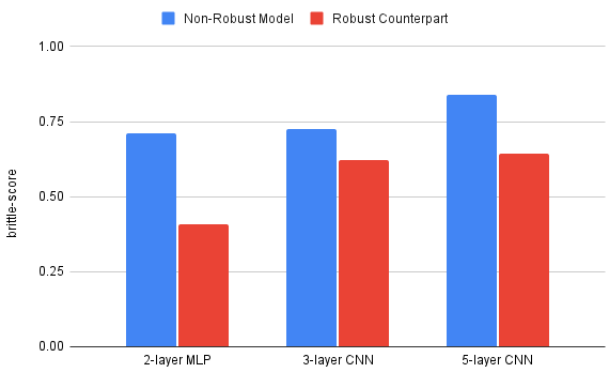

Figure 1. Comparison of `brittle-score` of various types of models trained normally or in an adversarial manner. Red bar being consistently lower than blue validates our hypothesis.

We also look at the explanations generated by these models qualitatively. We notice that explanations generated by the more robust model are typically sharper and focus on the most relevant parts of the input images (Fig. **??**). For non-robust models, the LIME weights are not as sparse. This in-turn motivates the definition of `brittle-score` which depends on the hypothesis that LIME weights of a robust model has a lower norm than the non-robust counterpart.

In addition to our controlled studies on MNIST, we also perform experiments on off-the-shelf open-source adversarially trained models available as a part of `robustbench`[2](Croce et al., 2021). We test with three other datasets – CIFAR10, CIFAR100 and ImageNet. `robustbench` provides a bunch of pre-trained checkpoints on these datasets which were defended using different methods and hence they also have different robust accuracies. This gives us a good way to compare `robust-acc` and `brittle-score` in order to validate our hypothesis. We use $n = 3000$ in Eq. 2 for CIFAR10 and CIFAR100 and $n = 100$ for ImageNet with $k = 1000$ samples for fitting the LIME function around a datapoint.

In Table 1, we show that with increasing `robust-acc`, the `brittle-score` decreases. The correlation between these two metrics is particularly intriguing since these two quantities prima facie appear to be completely unrelated. `robust-acc` is directly reported from

[2] `https://github.com/RobustBench/robustbench`

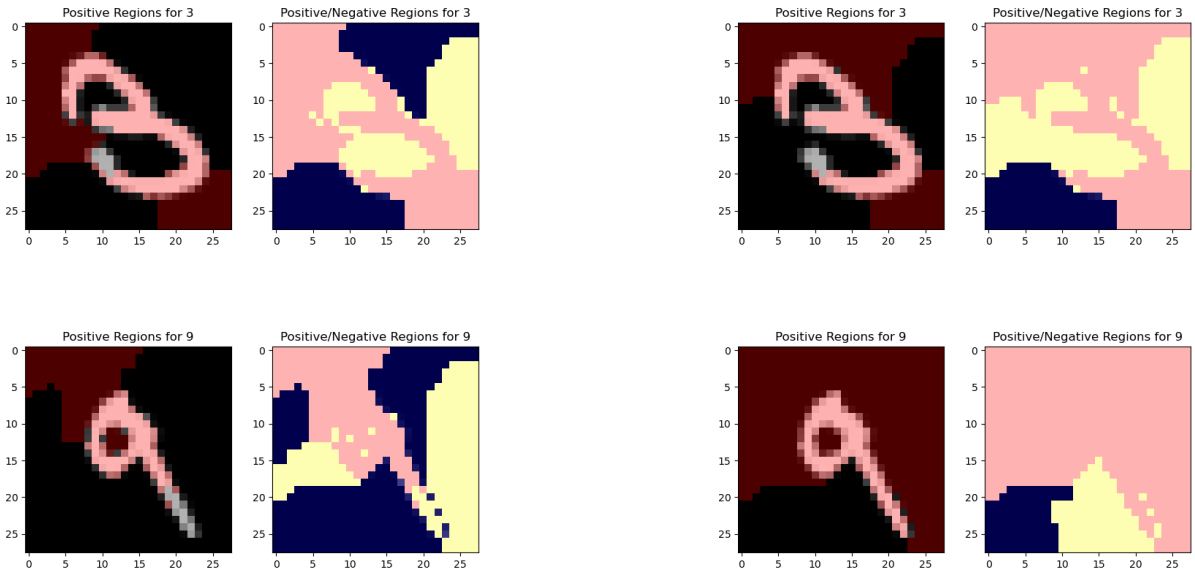

Figure 2. (Left) Explanations of robust 3-layer CNN and (Right) Explanations of non-robust 3-layer CNN. The robust model has better lime weights (sparser positive region) which is seen in larger yellow colored regions in the mask (blue=negative, pink=positive, yellow=zero)

`robustbench`(Croce et al., 2021). We also report the relative improvement (R.I.) for a model $\mathcal{M}$ over the standard model denoted as $\mathcal{S}.\mathcal{M}$ for each dataset which is defined as:

$$\text{Relative Improvement R.I.} = \frac{\texttt{brittle-score}(\mathcal{S}.\mathcal{M}) - \texttt{brittle-score}(\mathcal{M})}{\texttt{brittle-score}(\mathcal{S}.\mathcal{M})} \tag{3}$$

As mentioned earlier, `brittle-score` should only be compared within datasets. The absolute value of `brittle-score` is not interpretable as it is but it serves as a good metric to compare models. If we try to compare two different models which were trained and tested on different datasets, the absolute numbers and differences do not have a mathematical meaning. Moreover, it is sometimes hard to interpret the scale of `brittle-score` by itself. Unlike `robust-acc`, the scale of `brittle-score` is unbounded, and could be any number > 0, although we empirically observe that the number is < 1. For robust models, the difference in `brittle-score` itself is constantly shrinking, indicating a sub-linear relationship between `robust-acc` and `brittle-score`. We plot `robust-acc` versus `brittle-score` in Fig. 3 as both of them should be negatively correlated.

4. Conclusion & Future Directions

We illustrate that `brittle-score` is a great indicator of the robustness of any black-box model. We believe this is quite encouraging from not only a practical standpoint, but from a theoretical standpoint as well in understanding adversarial robustness and adversarial training. Moreover, this method makes evaluating and comparing black box models (for instance models provided via APIs), feasible and easy. With the increasing proliferation of deep learning methods, methods that can preserve intellectual property rights while indicating model safety are paramount for safety-centric deep learning. Some potential extensions of our work include: (a) a theoretical analysis to understand the relation between adversarial robustness and LIME weights, (b) developing black-box scoring methods that capture other safety related metrics.

Dataset	Model Name	`robust-acc`	`brittle-score`	R.I.
64emCIFAR-10	Standard (Non-Robust) (He et al., 2016)	0.0	0.6028	0
	Ding et al. (Ding et al., 2020)	41.44	0.5984	0.74
	Engstrom et al. (Engstrom et al., 2019)	49.25	0.5962	1.11
	Wu et al. (Wu et al., 2020)	56.17	0.5960	1.14
	Sehwag et al. (Sehwag et al., 2022)	60.27	0.5960	1.14
	Rebuffi et al. (Rebuffi et al., 2021)	**66.56**	**0.5956**	**1.22**
64emCIFAR-100	Standard (Non-Robust) (He et al., 2016)	3.95	0.6073	0
	Rice et al. (Rice et al., 2020)	18.95	0.6012	1.00
	Rebuffi et al. (Rebuffi et al., 2021)	28.5	0.5992	1.34
	Gowal et al. (Gowal et al., 2020)	30.03	0.5984	1.49
	Gowal et al. (Extra Model) (Gowal et al., 2020)	**36.88**	**0.5977**	**1.60**
64emImageNet	Standard (Non-Robust) (He et al., 2016)	0.0	0.74341	0
	Wong et al.(Wong et al., 2020)	26.24	0.74314	0.036
	Engstrom et al. (Engstrom et al., 2019)	29.22	0.74312	0.039
	Salman et al. (Salman et al., 2020)	**38.14**	**0.74309**	**0.043**

Table 1. A comparison of the trend between robust accuracy and brittle score across model architectures, adversarial defenses and datasets. R.I. standards for relative improvement as defined in Eq. 3.

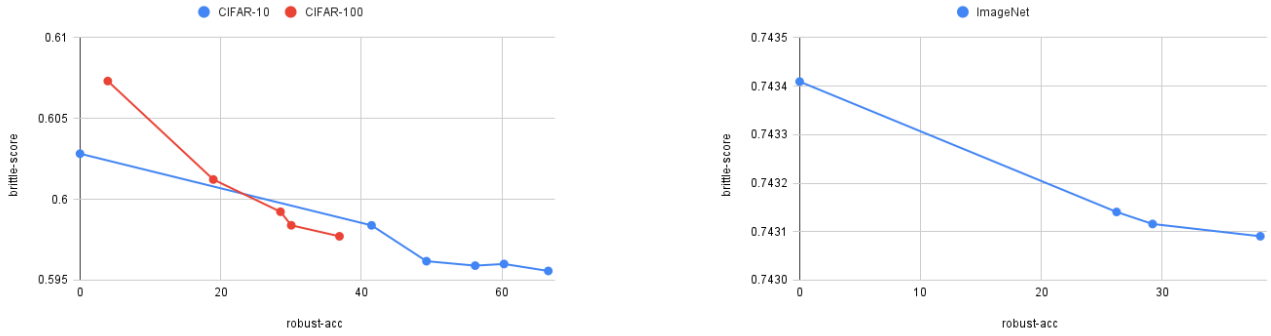

Figure 3. We plot `brittle-score` versus `robust-acc` for various datasets based on Table 1 showing that both of them are negatively correlated to one another

References

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

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

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://openreview.net/forum?id=SSKZPJCt7B.

Ding, G. W., Sharma, Y., Lui, K. Y. C., and Huang, R.

Mma training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkeryxBtPB.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., and Tsipras, D. Robustness (python library), 2019. URL https://github.com/MadryLab/robustness.

Finlayson, S. G., Chung, H. W., Kohane, I. S., and Beam, A. L. Adversarial attacks against medical deep learning systems. *arXiv preprint arXiv:1804.05296*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples, 2020. URL https://arxiv.org/abs/2010.03593.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJm4T4Kgx.

LECUN, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1999.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-pre pdf.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021. ISSN 0360-0300. doi: 10.1145/3457607. URL https://doi.org/10.1145/3457607.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., and Gebru, T. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pp. 220–229, 2019.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Na, T., Ko, J. H., and Mukhopadhyay, S. Cascade adversarial machine learning regularized with a unified embedding. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyRVBzap-.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. Diffusion models for adversarial purification. In *ICML*, 2022.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, 2016a. doi: 10.1109/EuroSP.2016.36.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, 2016b. doi: 10.1109/SP.2016.41.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ramesh21a.html.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. Data augmentation can improve robustness. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=kgVJBBThdSZ.

Ribeiro, M. T., Singh, S., and Guestrin, C. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Rice, L., Wong, E., and Kolter, J. Z. Overfitting in adversarially robust deep learning. In *ICML*, 2020.

Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., and Madry, A. Do adversarially robust imagenet models transfer better? In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3533–3545. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/24357dd085d2c4b1a88a7e0692e60294-Paper.pdf.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkJ3ibb0-.

Sehwag, V., Mahloujifar, S., Handina, T., Dai, S., Xiang, C., Chiang, M., and Mittal, P. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WVX0NNVBBkV.

Shaham, U., Yamada, Y., and Negahban, S. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomput.*, 307(C):195–204, sep 2018. ISSN 0925-2312. doi: 10.1016/j.neucom.2018.04.027. URL https://doi.org/10.1016/j.neucom.2018.04.027.

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJUYGxbCW.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

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

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJx040EFvH.

Wu, D., Xia, S.-T., and Wang, Y. Adversarial weight perturbation helps robust generalization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Xu, C., Ding, W., Lyu, W., Liu, Z., Wang, S., He, Y., Hu, H., Zhao, D., and Li, B. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=dwi57JI_-K.

Yoon, J., Hwang, S. J., and Lee, J. Adversarial purification with score-based generative models. In *ICML*, 2021.

Zhang, Y. and Liang, P. Defending against whitebox adversarial attacks via randomized discretization. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 684–693. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/zhang19b.html.