Efficient Estimation of Local Robustness of Machine Learning Models

Tessa Han[1] **Suraj Srinivas**[1] **Himabindu Lakkaraju**[1]

Abstract

Machine learning models often need to be robust to noisy input data. The effect of real-world noise (which is often random) on model predictions is captured by a model's local robustness, i.e., the consistency of model predictions in a local region around an input. However, the naïve approach to computing local robustness based on Monte-Carlo sampling is statistically inefficient, leading to prohibitive computational costs for large-scale applications. In this work, we develop the first analytical estimators to efficiently compute local robustness of multi-class discriminative models using local linear function approximation and the multivariate Normal CDF. Through the derivation of these estimators, we show how local robustness is connected to concepts such as randomized smoothing and softmax probability. We also confirm empirically that these estimators accurately and efficiently compute the local robustness of standard deep learning models. In addition, we demonstrate these estimators' usefulness for various tasks involving local robustness, such as measuring robustness bias and identifying examples that are vulnerable to noise perturbation in a dataset. By developing these analytical estimators, this work not only advances conceptual understanding of local robustness, but also makes its computation practical, enabling the use of local robustness in critical downstream applications.

1. Introduction

A desirable attribute of machine learning models is robustness to perturbations of input data. One common notion of robustness is adversarial robustness, a model's ability to maintain its prediction under adversarial perturbations. Although adversarial robustness can identify whether an adversarial perturbation exists, real-world noise (e.g., measurement noise) is rarely adversarial and often random. The effect of such noise on model predictions is captured by *local robustness*, the fraction of points in a local region around an input for which the model provides consistent predictions. This is a generalization of adversarial robustness – if this fraction is less than 1, then an adversarial perturbation exists. By capturing model behavior under average case noise, local robustness provides a more comprehensive characterization of real-world model behavior.

In this paper, we take the first steps towards measuring local robustness. We show that the naïve approach to estimating local robustness is statistically inefficient, leading to prohibitive computational costs for large-scale applications. To address this problem, we develop the first analytical estimators to efficiently compute local robustness. Specifically:

1. We derive a set of novel analytical estimators to efficiently compute the local robustness of multi-class discriminative models using local linear function approximation and the multivariate Normal CDF. Through the derivation, we show how local robustness is connected to randomized smoothing and softmax probability.
2. We empirically confirm that these analytical estimators accurately and efficient compute the local robustness of standard deep learning models.
3. We demonstrate the usefulness of our estimators for various tasks involving local robustness, such as measuring robustness bias and identifying examples that are vulnerable to noise perturbation. Such dataset-level analyses of local robustness are made practical only by having these efficient analytical estimators.

To our knowledge, this work is the first to investigate local robustness in a multi-class setting and develop efficient analytical estimators. The analytical aspect of these estimators not only advances conceptual understanding of local robustness, but also enables local robustness to be used in applications that require differentiability. The efficiency of these estimators makes the computation of local robustness practical, enabling tasks that assist in such important objectives as debugging models and establishing user trust.

[1] Harvard University, Cambridge, MA. Correspondence to: Tessa Han <than@g.harvard.edu>, Suraj Srinivas <ssrinivas@seas.harvard.edu>, Himabindu Lakkaraju <hlakkaraju@hbs.edu>.

Presented at the 2^{nd} Workshop on Formal Verification of Machine Learning, co-located with the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA., 2023. Copyright 2023 by the author(s).

2. Related Work

Linearization of neural networks. Prior works have used local linear function approximation to obtain feature attributions (Ribeiro et al., 2016; Han et al., 2022) or counterfactual explanations for binary classifiers (Pawelczyk et al., 2023). In contrast to these prior works which apply local linear function approximation to post hoc explainability, this work applies it to local robustness and uses it to develop analytical estimators for binary and multi-class classification.

Adversarial robustness. Prior works have proposed methods to generate adversarial attacks (Carlini and Wagner, 2017; Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016) and to provide dataset-level guarantees of model robustness (Cohen et al., 2019; Carlini et al., 2022). In contrast to these prior works on adversarial robustness, this work investigates local robustness, a generalization of adversarial robustness. Prior work has also studied robustness bias in terms of vulnerability to adversarial attacks (Nanda et al., 2021). In contrast, this work investigates robustness bias in terms of local robustness.

Uncertainty estimation. Prior works have developed approaches to measuring prediction uncertainty, including calibration (Guo et al., 2017), Bayesian uncertainty (Kendall and Gal, 2017), and conformal prediction (Shafer and Vovk, 2008). In contrast to these prior works in which uncertainty is with respect to a calibration set or model parameters, this work investigates local robustness, which can be thought of as uncertainty with respect to input noise.

3. Our Framework: The Local Robustness Estimator Family

In this section, we describe the mathematical problem of local robustness estimation. Then, we present the naïve estimator and derive more efficient analytical estimators. Lastly, we explore the connections between local robustness and softmax probability.

3.1. Notation and Preliminaries

Assume there is a neural network $f : \mathbb{R}^d \rightarrow \mathbb{R}^C$ with C classes, and the model predicts class $t \in [1, ...C]$ for a given input $\mathbf{x} \in \mathbb{R}^d$, i.e., $t = \arg\max_{i=1}^{C} f_i(\mathbf{x})$, where f_i denotes the logits for the i^{th} class. Given this model, the local robustness estimation problem is to compute the probability of consistent classification (to class t) under noise perturbation of the inputs.

Definition 1. We define the average local robustness of a model f at a point $\mathbf{x}$ as the probability of being classified to class t under Normal noise $\mathcal{N}(0, \sigma^2)$ added to the inputs:

$$p_\sigma^{\text{robust}}(\mathbf{x}, t) = P_{\epsilon \sim \mathcal{N}(0,\sigma^2)} \left[\arg\max_i f_i(x + \epsilon) = t \right]$$

The higher p_σ^{robust} is, the more robust the model is in the local region around $\mathbf{x}$. In this paper, given that local robustness is always with respect to the predicted class t, we henceforth suppress the dependence on t in the notation. Note that p_σ^{robust} *generalizes adversarial robustness*. Adversarial robustness detects the presence of a perturbation that leads to inconsistent classification (i.e., $\mathbf{1}(p_\sigma^{\text{robust}} < 1)$), while local robustness computes the probability of consistent classification (i.e., p_σ^{robust}). In the rest of this section, we derive estimators for p_σ^{robust}.

3.1.1. Estimator 0: The Monte-Carlo Estimator

A naïve estimator of p_σ^{robust} is a Monte-Carlo estimator, i.e.,

$$\begin{aligned} p_\sigma^{\text{robust}}(\mathbf{x}) &= \mathop{\mathbb{E}}_{\epsilon \sim \mathcal{N}(0,\sigma^2)} \left[\mathbf{1}_{\arg\max_i f_i(x+\epsilon)=t} \right] \\ &\approx \frac{1}{M} \sum_{j=1}^{M} \left[\mathbf{1}_{\arg\max_i f_i(x+\epsilon_j)=t} \right] = p_\sigma^{\text{mc}}(\mathbf{x}) \end{aligned}$$

In practice, p_σ^{mc} requires a large number of random samples to converge. For example, for CIFAR10 CNNs, it takes around $M = 10,000$ samples per point for p_σ^{mc} to converge, which is computationally infeasible. Thus, we seek to address this problem by developing more efficient estimators.

3.2. Analytical Estimators of Local Robustness

3.2.1. Estimator 1: The Taylor Estimator

To derive efficient estimators of local robustness, we locally linearize non-linear models and compute the local robustness of the resulting linear models. However, even computing the local robustness of linear models is challenging due to the complex geometry of decision boundaries given C classes. We derive the estimator for the linear model in Appendix A.1. Using this, we derive the Taylor estimator.

Proposition 1. The Taylor estimator for a model f and point $\mathbf{x}$ is given by linearizing f around $\mathbf{x}$ using $\mathbf{w} = \nabla_{\mathbf{x}} f(\mathbf{x})$ and $b = f(\mathbf{x})$, with decision boundaries $g_i(\mathbf{x}) = f_t(\mathbf{x}) - f_i(\mathbf{x})$, $\forall i \neq t$, leading to

$$p_\sigma^{\text{taylor}}(\mathbf{x}) = \text{CDF}_{\mathcal{N}(0,UU^\top)}\left(\left[\frac{g_1(\mathbf{x})}{\sigma \|\nabla_{\mathbf{x}} g_1(\mathbf{x})\|_2}, ..., \frac{g_C(\mathbf{x})}{\sigma \|\nabla_{\mathbf{x}} g_C(\mathbf{x})\|_2} \right] \right)$$

$$\text{where } \; U = \left[\frac{\nabla_{\mathbf{x}} g_1(\mathbf{x})'}{\|\nabla_{\mathbf{x}} g_1(\mathbf{x})'\|_2}, ..., \frac{\nabla_{\mathbf{x}} g_C(\mathbf{x})'}{\|\nabla_{\mathbf{x}} g_C(\mathbf{x})'\|_2} \right] \in \mathbb{R}^{(C-1)\times d}$$

The proof is in Appendix A.1. The smaller the σ, the more faithful the local linearization of the model, thus the more accurate the Taylor estimator.

3.2.2. Estimator 2: The MMSE Estimator

While the Taylor estimator is more efficient than the naïve one, it has a drawback: its linear approximation is less valid farther away from $\mathbf{x}$. To fix this, we use a linearization that is faithful to the model on the entire noise distribution, not just near $\mathbf{x}$, using SmoothGrad (Smilkov et al., 2017) (which has been described as the MMSE optimal linearization of the model (Han et al., 2022; Agarwal et al., 2021)). We propose the MMSE estimator as follows.

Proposition 2. The MMSE estimator for a model f and point $\mathbf{x}$ is given by linearizing f around $\mathbf{x}$ using $\mathbf{w} = \sum_{j=1}^{N} \nabla_{\mathbf{x}} f(\mathbf{x}+\epsilon)$ and $b = \sum_{j=1}^{N} f(\mathbf{x})$, with decision boundaries $g_i(\mathbf{x}) = f_t(\mathbf{x}) - f_i(\mathbf{x}), \forall i \neq t$, leading to

$$p_\sigma^{\text{mmse}}(\mathbf{x}) = \text{CDF}_{\mathcal{N}(0,UU^\top)}\Bigg(\Bigg[\frac{\frac{1}{N}\sum_{j=1}^{N} g_1(\mathbf{x}+\epsilon)}{\sigma\|\frac{1}{N}\sum_{j=1}^{N} \nabla_{\mathbf{x}} g_1(\mathbf{x}+\epsilon)\|_2}, \dots, \frac{\frac{1}{N}\sum_{j=1}^{N} g_C(\mathbf{x}+\epsilon)}{\sigma\|\frac{1}{N}\sum_{j=1}^{N} \nabla_{\mathbf{x}} g_C(\mathbf{x}+\epsilon)\|_2}\Bigg]\Bigg)$$

with $U \in \mathbb{R}^{(C-1)\times d}$ defined as in the Taylor estimator and N as the number of perturbations.

The proof is in Appendix A.1. p_σ^{mmse} creates a randomized smooth model (Cohen et al., 2019) from the base model and then computes the decision boundaries of this smooth model. We show, for the first time, that performing such randomization helps compute robustness information for the original base model.

Like p_σ^{mc}, p_σ^{mmse} also requires sampling over the input space. However, due to p_σ^{mmse}'s use of model gradients, it requires far fewer samples to converge (we observed $N = 5-10$ to suffice in practice), thus making it computationally efficient.

3.2.3. Estimators 3 & 4 : Approximate Taylor and MMSE Estimators

One drawback of the Taylor and MMSE estimators is their use of the *mvn-cdf* which does not have a closed form solution. As a result, these estimators can be slow when used for a large number of classes C and are non-differentiable (which is inconvenient for applications which require differentiating p_σ^{robust}). Thus, we wish to approximate the *mvn-cdf* with a closed-form expression. To this end, the *univariate* Normal CDF is well-approximated by the sigmoid function, and has been used to propose the GeLU activation function (Hendrycks and Gimpel, 2016). Inspired by this, we propose to approximate the *mvn-cdf* with the multivariate-sigmoid function:

Definition 2. The multivariate sigmoid is defined as $\text{mv-sigmoid}(\mathbf{x}) = \frac{1}{1+\sum_i \exp(-\mathbf{x}_i)}$

We find experimentally that *mv-sigmoid* well-approximates the *mvn-cdf* for practical values of the covariance matrix $UU^\top$. Substituting *mv-sigmoid* for the *mvn-cdf* in p_σ^{taylor} and p_σ^{mmse}, we get estimators $p_\sigma^{\text{taylor\_mvs}}$ and $p_\sigma^{\text{mmse\_mvs}}$.

3.3. Exploring the Connections Between Local Robustness and Softmax Probability

3.3.1. Estimator 5: Softmax Estimator

Lastly, for linear models with a specific σ, the common softmax function taken with respect to the output logits can be viewed as an estimator of p_σ^{robust}, albeit in a very restricted setting. Full discussion is in Appendix A.1.

4. Experimental Evaluation

We first evaluate the accuracy and efficiency of the analytical estimators. Then, we analyze the relationship between local robustness and softmax probability. Lastly, we demonstrate the usefulness of the estimators in real-world applications.

Datasets and Models. We use four datasets: MNIST (Deng, 2012), FashionMNIST (Xiao et al., 2017), CIFAR10 (Krizhevsky et al., 2009), and CIFAR100 (Krizhevsky et al., 2009). For MNIST and FashionMNIST, we train linear models and CNNs. For CIFAR10 and CIFAR100, we train ResNet18 (He et al., 2016) models with varying levels of gradient norm regularization (λ) for varying levels of robustness. For experiments, we use 1,000 randomly-selected points from each dataset's test set. Details about datasets and models are in Appendix A.3 and A.4.

4.1. Evaluation of the accuracy of analytical estimators

The analytical estimators accurately compute local robustness. We calculate p_σ^{robust} for each model using all six estimators for different σ's. Then, we measure the absolute and relative difference between p_σ^{mc} and the other estimators (Figure 1). The results indicate that $p_\sigma^{\text{mmse\_mvs}}$ and p_σ^{mmse} are the best estimators of p_σ^{robust}, followed closely by $p_\sigma^{\text{taylor\_mvs}}$ and p_σ^{taylor}, and trailed by p_T^{softmax}. The smaller the σ, the more accurate the estimators. In addition, for robust models, the analytical estimators are more accurate over a larger σ (Appendix A.2). The *mv-sigmoid* function also approximates the *mvn-cdf* well in practice (Appendix A.2). Consistent with the theory in Section 3, these results indicate that the analytical estimators accurately compute p_σ^{robust}.

	CPU: Intel x86_64		GPU: Tesla V100	
Estimator	Serial	Batched	Serial	Batched
$p_\sigma^{\text{mc}}(n=10000)$	1:41:11	1:14:38	0:19:56	0:00:35
p_σ^{taylor}	0:00:08	0:00:07	0:00:02	< 0:00:01
$p_\sigma^{\text{mmse}}(n=5)$	0:00:41	0:00:31	0:00:06	0:00:02

Table 1: Runtimes of p_σ^{robust} estimators (H:M:S). Each estimator computes $p_{\sigma=0.1}^{\text{robust}}$ for the CIFAR10 ResNet18 model for 50 points using the minimum number of samples n necessary for convergence. The analytical estimators (p_σ^{taylor} and p_σ^{mmse}) are more efficient than the naïve estimator (p_σ^{mc}).

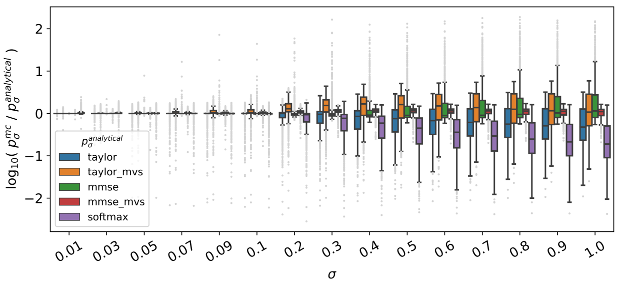


Figure 1: Experimental validation of analytical estimators (FashionMNIST CNN). p_σ^{mmse} and $p_\sigma^{\text{mmse\_mvs}}$ are the best estimators of p_σ^{robust}. The smaller the σ, the more accurate all of the estimators are.

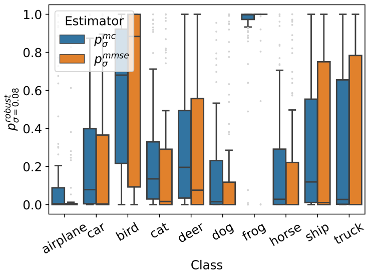


Figure 2: Local robustness bias among classes (CIFAR10 ResNet18). p_σ^{robust} reveals that the model is less locally robust for some classes than for others. The analytical estimator p_σ^{mmse} properly captures this model bias.

4.2. Evaluation of the efficiency of analytical estimators

The naïve estimator is statistically inefficient. We calculate p_σ^{mc} for each model using different sample sizes (n) over different σ's, and measure the absolute and relative difference between p_σ^{mc} at a given n and p_σ^{mc} at $n = 50,000$ (Appendix A.2). The results indicate that p_σ^{mc} requires around 10,000 samples per point to converge, which is impractical.

The analytical estimators are more efficient than the naïve estimator. We measure the runtimes of the estimators when calculating $p_{\sigma=0.1}^{\text{robust}}$ for the CIFAR10 ResNet18 model for 50 points (Table 1). Results indicate that p_σ^{taylor} and p_σ^{mmse} perform at least 35x and 17x faster than p_σ^{mc}, respectively.

4.3. Comparison of local robustness and softmax probability

Local robustness and softmax probability are two distinct measures. Consistent with the theory in Section 3, we find that p_σ^{robust} and p_T^{softmax} are not strongly correlated, indicating that in general settings, p_T^{softmax} is not a good estimator for p_σ^{robust}. Details are in Appendix A.2.

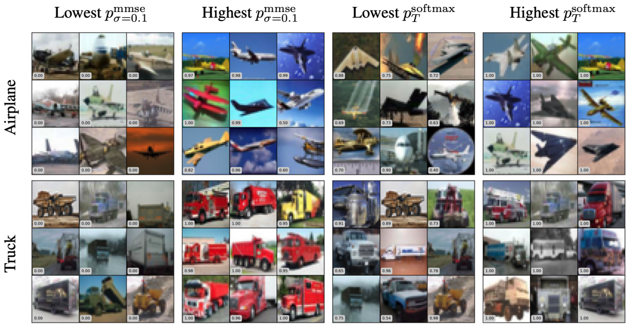

Lowest $p^{\text{mmse}}_{\sigma=0.1}$
Highest $p^{\text{mmse}}_{\sigma=0.1}$
Lowest p_T^{softmax}
Highest p_T^{softmax}
Airplane
Truck

Figure 3: Images with the lowest and highest p_σ^{robust} among CIFAR10 classes. Images with high p_σ^{robust} are brighter with stronger object-background contrast (making them more robust to random noise) than those with low p_σ^{robust}. This difference is less evident for p_T^{softmax}.

4.4. Applications of local robustness

p_σ^{robust} detects local robustness bias. We calculate p_σ^{robust} using p_σ^{mmse} and examine its distribution across classes (Figure 2). Results show that the models are more locally robust for some classes than for others. Thus, p_σ^{robust} can be applied to detect local robustness bias, which is critical when models are deployed in high-stakes, real-world settings.

p_σ^{robust} identifies images that are robust to and images that are vulnerable to random noise. We visualize images with the highest and lowest p_σ^{robust} and p_T^{softmax} in each class (Figure 3). Images with low p_σ^{robust} tend to have neutral colors and low object-background contrast while images with high p_σ^{robust} tend to be brightly-colored with high object-background contrast. These differences make the prediction more and less likely to change, respectively, when the image is perturbed. These differences are not as evident for p_T^{softmax}.

For all experiments described above, additional results are in Appendix A.5.

5. Conclusion

In this work, we take the first steps towards estimating local model robustness. We show that the naïve approach is inefficient and develop efficient analytical estimators. We empirically confirm the estimators' accuracy and efficiency. Then, we demonstrate the usefulness of these estimators in performing various real-world tasks.

To our knowledge, this work is the first to investigate local robustness in a multi-class setting and develop efficient analytical estimators. The analytical aspect of these estimators not only advances conceptual understanding of local robustness, connecting it to randomized smoothing and softmax probability, but also enables local robustness to be used in applications that require differentiability. In addition, the efficiency of these estimators makes the computation of local robustness practical.

One limitation of this work is its focus on classification. Defining local robustness and developing efficient analytical estimators for regression represent future research directions. Other directions include exploring additional applications of local robustness, such as uncertainty calibration and training locally robust models.

References

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you? Explaining the predictions of any classifier. *International Conference on Knowledge Discovery and Data Mining*, 2016.

Tessa Han, Suraj Srinivas, and Himabindu Lakkaraju. Which explanation should I choose? A function approximation perspective to characterizing post hoc explanations. *Neural Information Processing Systems*, 2022.

Martin Pawelczyk, Teresa Datta, Johannes van-den Heuvel, Gjergji Kasneci, and Himabindu Lakkaraju. Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse. *International Conference on Learning Representations*, 2023.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy*, 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A simple and accurate method to fool deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. *International Conference on Machine Learning*, 2019.

Nicholas Carlini, Florian Tramer, Krishnamurthy Dvijotham, Leslie Rice, Mingjie Sun, and Zico Kolter. (Certified!!) adversarial robustness for free! *International Conference on Learning Representations*, 2022.

Vedant Nanda, Samuel Dooley, Sahil Singla, Soheil Feizi, and John P Dickerson. Fairness through robustness: Investigating robustness disparity in deep learning. *ACM Conference on Fairness, Accountability, and Transparency*, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *International Conference on Machine Learning*, 2017.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Neural Information Processing Systems*, 2017.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 2008.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Sushant Agarwal, Shahin Jabbari, Chirag Agarwal, Sohini Upadhyay, Steven Wu, and Himabindu Lakkaraju. Towards the unification and robustness of perturbation and gradient based explanations. *International Conference on Machine Learning*, 2021.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 2012.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Zdravko I Botev. The normal law under linear restrictions: Simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 2017.

SciPy multivariate normal CDF. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.multivariate_normal.html.

A. Appendix

A.1. Proofs

A.1.1. Local robustness for linear models

Our goal is to derive analytical estimators which reduce the complexity of estimating local robustness. To this end, we first locally linearize non-linear models and then compute the local robustness of the resulting linear models. However, even the problem of computing the local robustness of linear models is more challenging than it appears due to the complex geometry of linear decision boundaries given C classes. In particular, the relative orientation and similarities of these class-wise decision boundaries needs to be taken into account to compute local robustness.

Given a linear model for a three-class classification problem with weights w_1, w_2, w_3 and biases b_1, b_2, b_3, such that $y = \arg\max_i \{w_i^\top \mathbf{x} + b_i \mid i \in [1, 2, 3]\}$, the decision boundary between classes 1 and 2 is given by $y_{12} = (w_1 - w_2)^\top \mathbf{x} + (b_1 - b_2)$. This is easy to verify as for any $\mathbf{x}$ such that $y_{12} = 0$, we have $w_1^\top \mathbf{x} + b_1 = w_2^\top \mathbf{x} + b_2$, making it the decision boundary. Thus, the relevant quantities are the pairwise difference terms among the weights and biases which characterize the decision boundaries. We take this into account and provide the expression for the linear case below.

Lemma A.1. *The local robustness of a multi-class linear model $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, with $\mathbf{w} \in \mathbb{R}^{d \times C}$ and $b \in \mathbb{R}^C$, with respect to a target class t is given by the following. Define the decision boundary weights $w'_i = w_t - w_i \in \mathbb{R}^d, \forall i \neq t$, where w_t, w_i are rows of $\mathbf{w}$ and biases $b'_i = (w'_t - w'_i)^\top \mathbf{x} + (b_t - b_i) \in \mathbb{R}$, then*

$$p_\sigma^{robust}(\mathbf{x}) = CDF_{\mathcal{N}(0, UU^\top)} \left(\frac{b'_1}{\sigma \|w'_1\|_2}, \cdots \frac{b'_i}{\sigma \|w'_i\|_2}, \cdots \frac{b'_C}{\sigma \|w'_C\|_2} \right)$$

$$\textit{where} \quad U = \left[\frac{w'_1}{\|w'_1\|_2}; \cdots \frac{w'_i}{\|w'_i\|_2}; \cdots \frac{w'_C}{\|w'_C\|_2} \right] \in \mathbb{R}^{(C-1) \times d}$$

and $CDF_{\mathcal{N}(0, UU^\top)}$ refers to the $(C-1)$-dimensional Normal CDF with covariance $UU^\top$.

The proof is below. The matrix U exactly captures the geometry of the linear decision boundaries and the covariance matrix $UU^\top$ encodes the relative similarity between pairs of decision boundaries. If the decision boundaries are all orthogonal to each other, then the covariance matrix is the identity matrix. However, we find that, in practice, the covariance matrix is strongly non-diagonal, indicating that the decision boundaries are not orthogonal to each other.

For diagonal covariance matrices, the multivariate Normal CDF (*mvn-cdf*) can be written as the product of univariate Normal CDFs, which would be easy to compute. However, the strong non-diagonal nature of covariance matrices in practice leads to the resulting *mvn-cdf* not having a simple closed form solution, with the only alternative being approximation of the integral via sampling (Botev, 2017; Sci). However, this sampling is performed in the $(C-1)$-dimensional space as opposed to the d-dimensional space that p_σ^{mc} performs. In practice, for classification problems, we often have $C << d$, making sampling in $(C-1)$-dimensions more efficient.

Proof. First, we rewrite $p_\sigma^{\mathrm{robust}}$ in the following manner, by defining $g_i(\mathbf{x}) = f_t(\mathbf{x}) - f_i(\mathbf{x}) > 0$, which is the "decision boundary function".

$$p_\sigma^{\mathrm{robust}} = P_{\epsilon \sim \mathcal{N}(0, \sigma^2)} \left[\max_i f_i(\mathbf{x} + \epsilon) < f_t(\mathbf{x} + \epsilon) \right] = P_{\epsilon \sim \mathcal{N}(0, \sigma^2)} \left[\bigcup_{i=1; i \neq t}^{C} g_i(\mathbf{x} + \epsilon) > 0 \right]$$

Now, assuming that f, g are linear such that $g_i(\mathbf{x}) = {w'_i}^\top \mathbf{x} + g(0)$, we have $g_i(\mathbf{x} + \epsilon) = g_i(\mathbf{x}) + {w'_i}^\top \epsilon$, and obtain

$$
\begin{aligned}
p_\sigma^{\text{robust}} &= P_{\epsilon\sim\mathcal{N}(0,\sigma^2)}\left[\bigcup_{i=1;i\neq t}^{C} {w'_i}^\top \epsilon > -g_i(\mathbf{x})\right] \\
&= P_{z\sim\mathcal{N}(0,I_d)}\left[\bigcup_{i=1;i\neq t}^{C} \frac{w'_i}{\|w'_i\|_2}^\top z > -\frac{g_i(\mathbf{x})}{\sigma\|w'_i\|_2}\right] \qquad \text{(Rescaling and standardization)}
\end{aligned}
$$

We now make the following observations:

- For any matrix $U \in \mathbb{R}^{C\times d}$ and a d-dimensional Gaussian random variable $z \sim \mathcal{N}(0, I_d) \in \mathbb{R}^d$, we have $U^\top z \sim \mathcal{N}(0, UU^\top)$, i.e., an C-dimensional Gaussian random variable.
- CDF of a multivariate Gaussian RV is defined as $P_z[\bigcup_i z_i < t_i]$ for some input values t_i

Using these observations, if we construct $U = [\frac{w'_1}{\|w'_1\|_2}; \frac{w'_2}{\|w'_2\|_2}; \ldots \frac{w'_C}{\|w'_C\|_2}] \in \mathbb{R}^{(C-1)\times d}$, and obtain

$$
\begin{aligned}
p_{\text{robust}} &= P_{u\sim\mathcal{N}(0,UU^\top)}\left[\bigcup_{i=1;i\neq t}^{C} u_i < \frac{g_i(\mathbf{x})}{\sigma\|w'_i\|_2}\right] \\
&= \text{CDF}_{\mathcal{N}(0,UU^\top)}(\left[\frac{g_1(\mathbf{x})}{\sigma\|w'_1\|_2}, \frac{g_2(\mathbf{x})}{\sigma\|w'_2\|_2}, \ldots \frac{g_C(\mathbf{x})}{\sigma\|w'_C\|_2}\right])
\end{aligned}
$$

where $g_i(\mathbf{x}) = {w'_i}^\top \mathbf{x} + g_i(0) = (w'_t - w'_i)^\top \mathbf{x} + (b_t - b_i)$

□

A.1.2. Taylor Estimator

Proposition 3. The Taylor estimator for a model f and point $\mathbf{x}$ is given by linearizing f around $\mathbf{x}$ using $\mathbf{w} = \nabla_{\mathbf{x}} f(\mathbf{x})$ and $b = f(\mathbf{x})$, with decision boundaries $g_i(\mathbf{x}) = f_t(\mathbf{x}) - f_i(\mathbf{x})$, leading to

$$
p_\sigma^{\text{taylor}}(\mathbf{x}) = \text{CDF}_{\mathcal{N}(0,UU^\top)}(\left[\frac{g_1(\mathbf{x})}{\sigma\|\nabla_{\mathbf{x}} g_1(\mathbf{x})\|_2}, \ldots \frac{g_i(\mathbf{x})}{\sigma\|\nabla_{\mathbf{x}} g_i(\mathbf{x})\|_2}, \ldots \frac{g_C(\mathbf{x})}{\sigma\|\nabla_{\mathbf{x}} g_C(\mathbf{x})\|_2}\right])
$$

with $U \in \mathbb{R}^{(C-1)\times d}$ defined as in the linear case.

Proof. Using the notations from the previous Lemma A.1, we can use $g(\mathbf{x} + \epsilon) \approx g(\mathbf{x}) + \nabla_{\mathbf{x}} g(\mathbf{x})^\top \epsilon$ using a first order Taylor series expansion. Thus we use $w'_i = \nabla_{\mathbf{x}} g_i(\mathbf{x})$ and $b' = g(\mathbf{x})$, and plug it into the result of Lemma A.1. □

A.1.3. MMSE Estimator

Proposition 4. The MMSE estimator for a model f and point $\mathbf{x}$ is given by linearizing f around $\mathbf{x}$ using $\mathbf{w} = \sum_{j=1}^{N} \nabla_{\mathbf{x}} f(\mathbf{x} + \epsilon)$ and $b = \sum_{j=1}^{N} f(\mathbf{x})$, with decision boundaries $g_i(\mathbf{x}) = f_t(\mathbf{x}) - f_i(\mathbf{x})$, leading to

$$
p_\sigma^{\text{mmse}}(\mathbf{x}) = \text{CDF}_{\mathcal{N}(0,UU^\top)}(\left[\frac{\sum_{j=1}^{N} g_1(\mathbf{x}+\epsilon)}{\sigma\|\sum_{j=1}^{N} \nabla_{\mathbf{x}} g_1(\mathbf{x}+\epsilon)\|_2}, \ldots \frac{\sum_{j=1}^{N} g_C(\mathbf{x}+\epsilon)}{\sigma\|\sum_{j=1}^{N} \nabla_{\mathbf{x}} g_C(\mathbf{x}+\epsilon)\|_2}\right])
$$

with $U \in \mathbb{R}^{(C-1)\times d}$ defined as in the linear case, where N is the number of perturbations.

Proof. We would like to improve upon the Taylor approximation to $g(\mathbf{x}+\epsilon)$ by using an MMSE local function approximation. Essentially, we'd like the find $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that

$$(w^*(\mathbf{x}), b^*(\mathbf{x})) = \arg\min_{w,b} \mathop{\mathbb{E}}_{\epsilon\sim\mathcal{N}(0,\sigma^2)} (g(x+\epsilon) - w^\top \epsilon - b)^2$$

A straightforward solution by finding critical points and equating it to zero gives us the following:

$$\begin{aligned} w^*(\mathbf{x}) &= \mathop{\mathbb{E}}_{\epsilon} \left[g(x+\epsilon)\epsilon^\top \right] / \sigma^2 = \mathop{\mathbb{E}}_{\epsilon} \left[\nabla_{\mathbf{x}} g(\mathbf{x}+\epsilon) \right] \qquad \text{(Stein's Lemma)} \\ b^*(\mathbf{x}) &= \mathop{\mathbb{E}}_{\epsilon} g(x+\epsilon) \end{aligned}$$

Plugging in these values of w^*, b^* into Lemma A.1, we have the result.

□

A.1.4. Softmax Estimator

Lastly, we observe that for linear models with a specific noise perturbation σ, the common softmax function taken with respect to the output logits can be viewed as an estimator of p_σ^{robust}, albeit in a very restricted setting. Specifically,

Lemma A.2. *For linear models $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, such that the decision boundary weight norms $\|w'_i\|_2 = \|w'_j\|_2 = \|w\|_2, \forall i, j$, we have*

$$p_T^{\textit{softmax}} = p_\sigma^{\textit{taylor\_mvs}} \quad \textit{where} \quad T = \sigma \|w\|_2$$

Proof. Let us consider softmax with respect to the t^{th} output class and define $g_i(\mathbf{x}) = f_t(\mathbf{x}) - f_i(\mathbf{x})$, with f being the linear model logits. Using this, we first show that softmax is identical to *mv-sigmoid*:

$$\begin{aligned} p_T^{\text{softmax}}(\mathbf{x}) &= \text{softmax}_t(f_1(\mathbf{x})/T, ...f_C(\mathbf{x})/T) \\ &= \frac{\exp(f_t(\mathbf{x})/T)}{\sum_i \exp(f_i(\mathbf{x})/T)} \\ &= \frac{1}{1 + \sum_{i;i\neq t} \exp((f_i(\mathbf{x}) - f_t(\mathbf{x}))/T)} \\ &= \text{mv-sigmoid}\left[g_1(\mathbf{x})/T, g_2(\mathbf{x})/T, ...g_C(\mathbf{x})/T\right] \end{aligned}$$

Next, by denoting $w'_i = w_t - w_i$, each row has equal norm $\|w'_i\|_2 = \|w'_j\|_2, \forall i, j, t \in [1, ...C]$ which implies:

$$\begin{aligned} p_\sigma^{\text{taylor\_mvs}}(\mathbf{x}) &= \text{mv-sigmoid}\left[\frac{g_1(\mathbf{x})}{\sigma\|w'_1\|_2}, ... \frac{g_C(\mathbf{x})}{\sigma\|w'_C\|_2}\right] \\ &= \text{mv-sigmoid}\left[g_1(\mathbf{x})/T, g_2(\mathbf{x})/T, ...g_C(\mathbf{x})/T\right] \qquad \because T = \sigma\|w'_i\|_2 \\ &= p_T^{\text{softmax}}(\mathbf{x}) \end{aligned}$$

□

This indicates that the temperature parameter T of softmax roughly corresponds to the σ of the added Normal noise with respect to which local robustness is measured. Overall, this shows that under the restricted setting where the local linear model consists of decision boundaries with equal weight norms, the softmax outputs can be viewed as an estimator of the $p_\sigma^{\text{taylor\_mvs}}$ estimator, which itself is an estimator of p_σ^{robust}. However, due to the multiple levels of approximation, we can expect the quality of p_T^{softmax}'s approximation of p_σ^{robust} to be poor in general settings (outside of the very restricted setting), so much so that in general settings, p_σ^{robust} and p_T^{softmax} would be unrelated.

A.2. Experiments referenced in main paper

For robust models, the analytical estimators compute local robustness more accurately over a larger noise neighborhood. The performance of p_σ^{mmse} for CIFAR10 ResNet18 models of varying levels of robustness is shown in Figure 4. The results indicate that for more robust models (larger λ), the estimator is more accurate over a larger σ. This is because gradient norm regularization leads to models that are more locally linear, making the estimator's linear approximation of the model around the input more accurate over a larger σ.

The mv-sigmoid function approximates the multivariate Normal CDF well in practice. To examine *mv-sigmoid*'s approximation of *mvn-cdf*, we compute both functions using the same inputs ($z = \left[\frac{g_1(\mathbf{x})}{\sigma\|\nabla_{\mathbf{x}} g_1(\mathbf{x})\|_2}, ..., \frac{g_C(\mathbf{x})}{\sigma\|\nabla_{\mathbf{x}} g_C(\mathbf{x})\|_2}\right]$, as described in Proposition 1) for the CIFAR10 ResNet18 model and its test set for different σ's. The plot of *mv-sigmoid(z)* against *mvn-cdf(z)* for $\sigma = 0.1$ is shown in Figure 5. The results indicate that the two functions are strongly positively correlated, suggesting that *mv-sigmoid* approximates the *mvn-cdf* well in practice.

Local robustness and softmax probability are two distinct measures. To examine the relationship between $p_\sigma^{\mathrm{robust}}$ and p_T^{softmax}, we calculate p_σ^{mmse} and p_T^{softmax} for CIFAR10 and CIFAR100 models of varying levels of robustness, and measure the correlation of their values and ranks using Pearson and Spearman correlations. Results are in Appendix A.2. For a non-robust model, $p_\sigma^{\mathrm{robust}}$ and p_T^{softmax} are not strongly correlated (Figure 7). As model robustness increases, the two quantities become more correlated (Figures 8 and 9). However, even for robust models, the relationship between the two quantities is mild (Figure 9). That $p_\sigma^{\mathrm{robust}}$ and p_T^{softmax} are not strongly correlated is consistent with the theory in Section 3: in general settings, p_T^{softmax} is not a good estimator for $p_\sigma^{\mathrm{robust}}$.

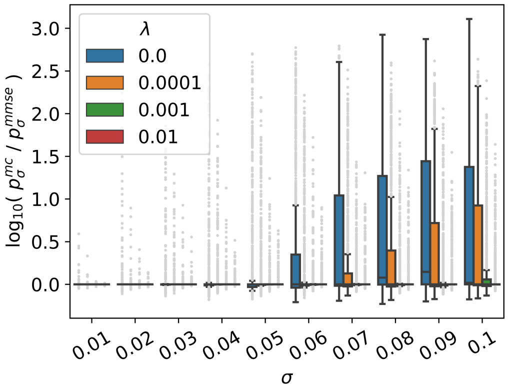


Figure 4: Experimental validation of analytical estimators. Figure shows results for the CIFAR10 ResNet18 model. For more robust models, the estimators compute $p_\sigma^{\mathrm{robust}}$ more accurately over a larger noise neighborhood.

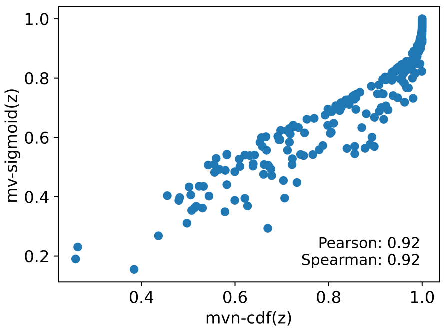


Figure 5: Correlation of *mvn-cdf(z)* and *mv-sigmoid(z)* for the CIFAR10 ResNet18 model. The formulation of z is described in Section 4.1. In practice, *mv-sigmoid* approximates *mvn-cdf* well.

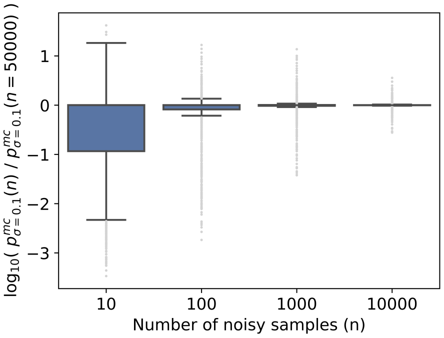


Figure 6: Convergence of the naïve estimator p_σ^{mc} for the CIFAR10 ResNet18 model as the number of noisy samples increases. In practice, p_σ^{mc} is statistically inefficient.

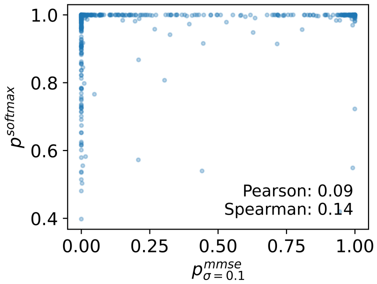


Figure 7: Relationship between p_σ^{robust} and p_T^{softmax} for a non-robust CIFAR10 ResNet18 models. For a non-robust model ($\lambda = 0$), p_σ^{robust} and p_T^{softmax} are not strongly correlated.

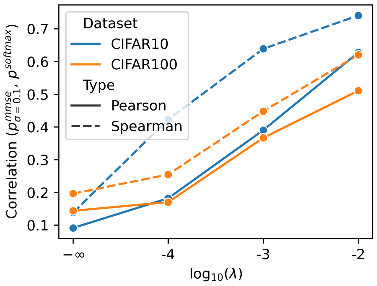


Figure 8: Relationship between p_σ^{robust} and p_T^{softmax} for the CIFAR10 ResNet18 and CIFAR100 ResNet18 models. As model robustness increases, p_σ^{robust} and p_T^{softmax} become more correlated.

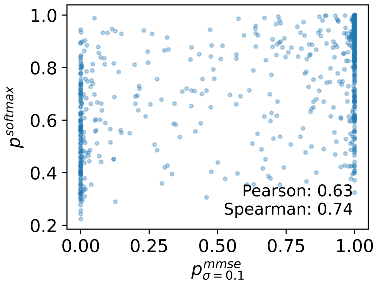


Figure 9: Relationship between p_σ^{robust} and p_T^{softmax} for a robust CIFAR10 ResNet18 model. Although p_σ^{robust} and p_T^{softmax} become more correlated as model robustness increases, even for robust models, the relationship between p_σ^{robust} and p_T^{softmax} is mild. These results indicate that, consistent with the theory in Section 3, p_T^{softmax} is not a good estimator for p_σ^{robust} in general settings.

A.3. Datasets

The MNIST dataset consists of images of gray-scale handwritten digits. The images span 10 classes: digits 0 through 9. Each image is of size 28 pixels x 28 pixels. The training set consists of 60,000 images and the test set consists of 10,000 images.

The FashionMNIST dataset consists of gray-scale images of articles of clothing. The images span 10 classes: t-shirt, trousers, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. Each image is of size 28 pixels x 28 pixels. The training set consists of 60,000 images and the test set consists of 10,000 images.

The CIFAR10 dataset consists of color images of common objects and animals. The images span 10 classes: airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck. Each image is of size 3 pixels x 32 pixels x 32 pixels. The training set consists of 50,000 images and the test set consists of 10,000 images.

The CIFAR100 dataset consists of color images of common objects and animals. The images span 100 classes: apple, bowl, chair, dolphin, lamp, mouse, plain, rose, squirrel, train, etc. Each image is of size 3 pixels x 32 pixels x 32 pixels. The training set consists of 50,000 images and the test set consists of 10,000 images.

For experiments, we use 1,000 randomly-selected test set images for each dataset.

A.4. Models

For the MNIST and FashionMNIST (FMNIST) datasets, we train a linear model and a convolutional neural network (CNN) to perform 10-class classification. The linear model consists of one hidden layer with 10 neurons. The CNN consists of four hidden layers: one convolutional layer with 5x5 filters and 10 output channels, one convolutional layer 5x5 filters and 20 output channels, and one linear layer with 50 neurons, and one linear layer 10 neurons.

For CIFAR10 and CIFAR100 datasets, we train a ResNet18 model to perform 10-class and 100-class classification, respectively. The model architecture is described in (He et al., 2016). We train the ResNet18 models using varying levels of gradient norm regularization to obtain models with varying levels of robustness. The larger the weight of gradient norm regularization (λ), the more robust the model.

All models were trained using stochastic gradient descent. Hyperparameters were selected to achieve decent model performance. The emphasis is on analyzing the estimators' estimates of local robustness of each model, not on high model performance. Thus, we do not focus on tuning model hyperparameters. All models were trained for 200 epochs. The test set accuracy (on each dataset's full 10,000-point test set) for each model is shown in Table 2.

Dataset	Model	λ	Test set accuracy
MNIST	Linear	0	92%
MNIST	CNN	0	99%
FashionMNIST	Linear	0	84%
FashionMNIST	CNN	0	91%
CIFAR10	ResNet18	0	94%
CIFAR10	ResNet18	0.0001	93%
CIFAR10	ResNet18	0.001	90%
CIFAR10	ResNet18	0.01	85%
CIFAR100	ResNet18	0	76%
CIFAR100	ResNet18	0.0001	74%
CIFAR100	ResNet18	0.001	69%
CIFAR100	ResNet18	0.01	60%

Table 2: Accuracy of models on test set.

A.5. Experiments

A.5.1. Convergence of p_σ^{mc}

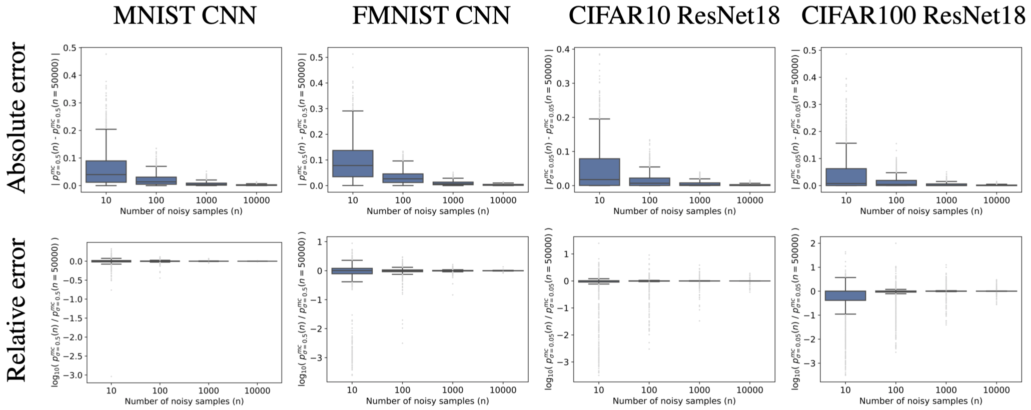


Figure 10: Convergence of p_σ^{mc}.

A.5.2. Convergence of p_σ^{mmse}

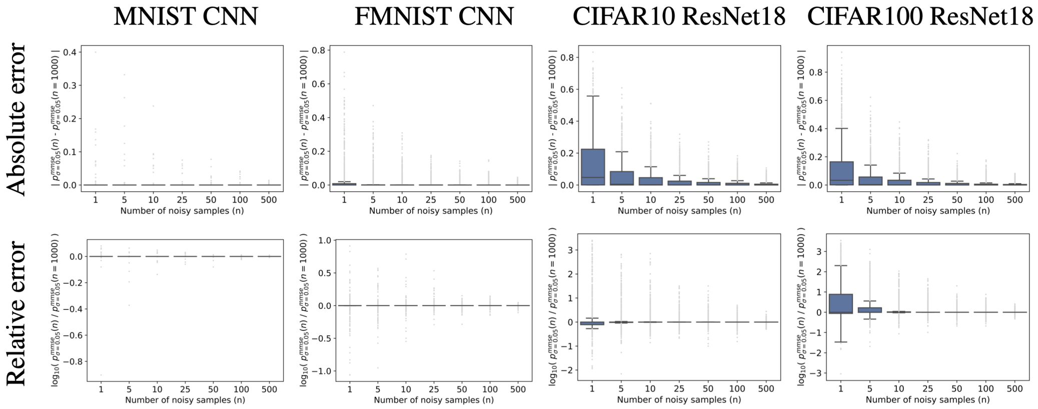
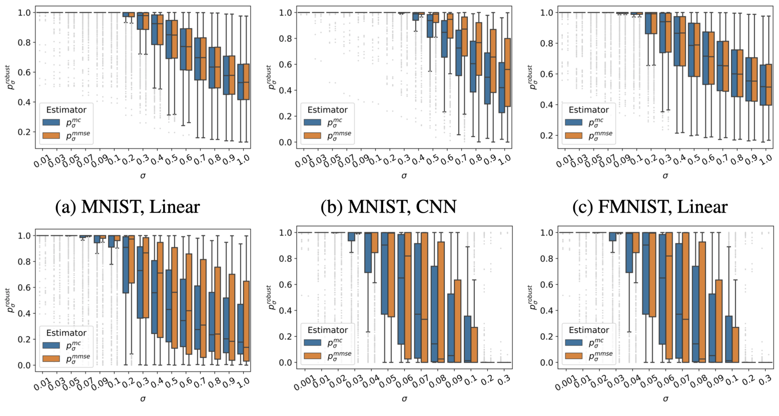


Figure 11: Convergence of p_σ^{mmse}.

A.5.3. Distribution of p_σ^{robust} over noise

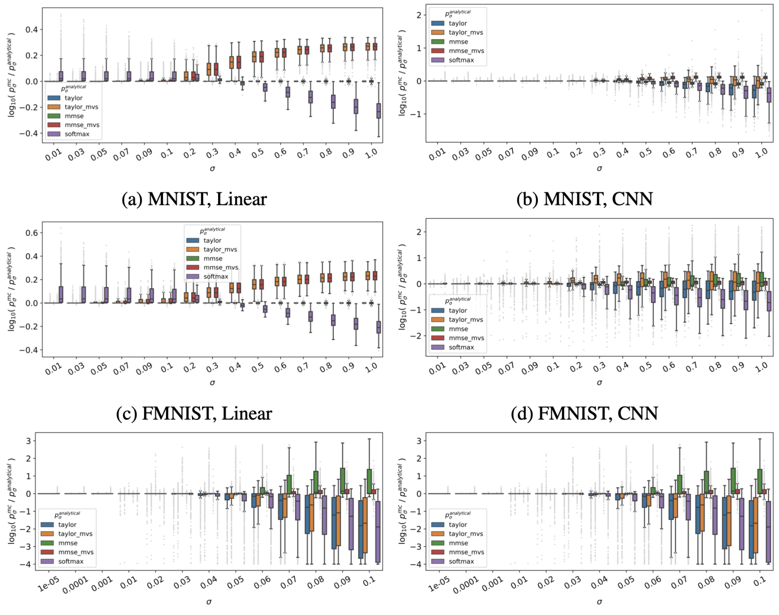


(a) MNIST, Linear (b) MNIST, CNN (c) FMNIST, Linear

(d) FMNIST, CNN (e) CIFAR10, ResNet18 (f) CIFAR100, ResNet18

Figure 12: Distribution of p_σ^{robust} over σ.

A.5.4. Accuracy of p_σ^{robust} estimators



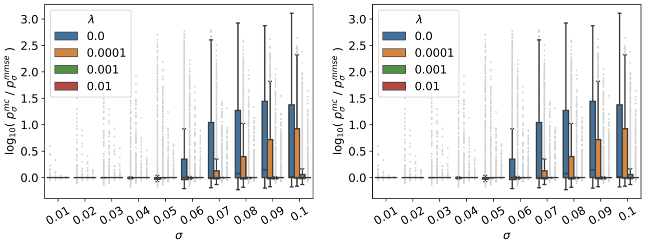
(a) MNIST, Linear (b) MNIST, CNN

(c) FMNIST, Linear (d) FMNIST, CNN

(e) CIFAR10, ResNet18 (f) CIFAR100, ResNet18

Figure 13: Accuracy of p_σ^{robust} estimators over σ.

A.5.5. Accuracy of p_σ^{robust} estimators for robust models



(a) CIFAR10, ResNet18 (b) CIFAR100, ResNet18

Figure 14: Accuracy of p_σ^{robust} estimators over σ for robust models.

A.5.6. mv-sigmoid function's approximation of mvn-cdf function

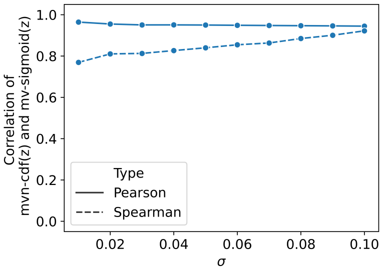
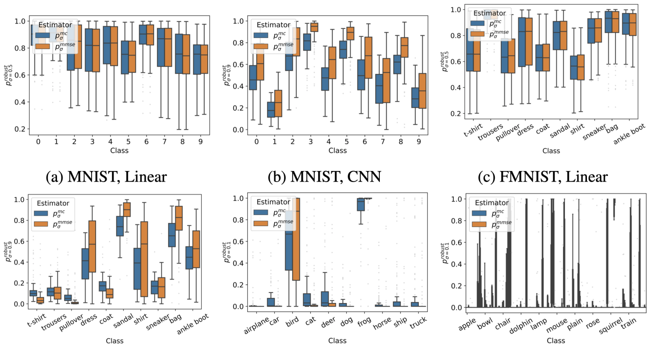


Figure 15: mv-sigmoid function's approximation of mvn-cdf function over σ.

A.5.7. Local robustness bias among classes



(a) MNIST, Linear (b) MNIST, CNN (c) FMNIST, Linear

(d) FMNIST, CNN (e) CIFAR10, ResNet18 (f) CIFAR100, ResNet18

Figure 16: Distribution of p_σ^{robust} for across classes.

A.5.8. Runtimes of p_σ^{robust} estimators

		CPU: Intel x86_64		GPU: Tesla V100-PCIE-32GB	
Estimator	# samples (n)	Serial	Batched	Serial	Batched
p_σ^{mc}	$n = 100$	0:00:59	0:00:42	0:00:12	0:00:01
	$n = 1000$	0:09:50	0:07:22	0:02:00	0:00:04
	$n = 10000$	*1:41:11*	*1:14:38*	*0:19:56*	*0:00:35*
p_σ^{taylor}	N/A	0:00:08	0:00:07	0:00:02	$<$ 0:00:01
$p_\sigma^{\text{taylor\_mvs}}$	N/A	0:00:08	0:00:07	0:00:01	$<$ 0:00:01
p_σ^{mmse}	$n = 1$	0:00:08	0:00:10	0:00:02	0:00:02
	$n = 5$	*0:00:41*	*0:00:31*	*0:00:06*	*0:00:02*
	$n = 10$	0:01:21	0:01:02	0:00:11	0:00:02
	$n = 25$	0:03:21	0:02:44	0:00:26	0:00:03
	$n = 50$	0:06:47	0:05:38	0:00:51	0:00:04
	$n = 100$	0:13:57	0:11:31	0:01:42	0:00:06
$p_\sigma^{\text{mmse\_mvs}}$	$n = 1$	0:00:08	0:00:08	0:00:01	0:00:01
	$n = 5$	*0:00:41*	*0:00:32*	*0:00:05*	*0:00:01*
	$n = 10$	0:01:21	0:01:00	0:00:10	0:00:02
	$n = 25$	0:03:24	0:02:37	0:00:25	0:00:02
	$n = 50$	0:06:47	0:05:35	0:00:51	0:00:03
	$n = 100$	0:13:28	0:11:32	0:01:42	0:00:06
p_T^{softmax}	N/A	0:00:01	$<$ 0:00:01	$<$ 0:00:01	$<$ 0:00:01

Table 3: Runtimes of each p_σ^{robust} estimator. Each estimator computes $p_{\sigma=0.1}^{\text{robust}}$ for the CIFAR10 ResNet18 model for 50 data points. For estimators that use sampling, the row with the minimum number of samples necessary for convergence is italicized. The analytical estimators (p_σ^{taylor}, $p_\sigma^{\text{taylor\_mvs}}$, p_σ^{mmse}, and $p_\sigma^{\text{mmse\_mvs}}$) are more efficient than the naïve estimator (p_σ^{mc}). Runtimes are in the format of hour:minute:second.

A.6. Broader Impact

This work is concerned with improving estimation of local robustness of machine learning models, and as such does not have any immediate foreseeable negative societal impact. However, inexact estimation can affect downstream decisions, and as such, estimator quality must always be taken into account to mitigate such cases.