A APPENDIX

A.1 PROOFS

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 $b \in \mathbb{R}^C$) at point $\mathbf{x}$ with respect to a target class t is given by the following. Define weights $\mathbf{u}_i = \mathbf{w}_t - \mathbf{w}_i \in \mathbb{R}^d, \forall i \neq t$, where $\mathbf{w}_t, \mathbf{w}_i$ are rows of $\mathbf{w}$ and biases $c_i = {\mathbf{u}_i}^\top \mathbf{x} + (b_t - b_i) \in \mathbb{R}$. Then,*

$$p_\sigma^{robust}(\mathbf{x}) = \Phi_{\mathbf{U}\mathbf{U}^\top} \left(\frac{c_i}{\sigma \|\mathbf{u}_i\|_2} \Big|_{\substack{i=1 \\ i \neq t}}^{C} \right)$$
$$\text{where } \mathbf{U} = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2} \Big|_{\substack{i=1 \\ i \neq t}}^{C} \in \mathbb{R}^{(C-1) \times d}$$

and $\Phi_{\mathbf{U}\mathbf{U}^\top}$ is the $(C-1)$-dimensional Normal CDF with zero mean and covariance $\mathbf{U}\mathbf{U}^\top$.

Proof. First, we rewrite p_σ^{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".

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

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

$$p_\sigma^{\text{robust}} = P_{\epsilon \sim \mathcal{N}(0,\sigma^2)} \left[\bigcup_{i=1; i \neq t}^{C} {\mathbf{u}_i}^\top \epsilon > -g_i(\mathbf{x}) \right] \tag{1}$$
$$= P_{z \sim \mathcal{N}(0,I_d)} \left[\bigcup_{i=1; i \neq t}^{C} \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2}^\top z > -\frac{g_i(\mathbf{x})}{\sigma \|\mathbf{u}_i\|_2} \right] \tag{2}$$

This step simply involves rescaling and standardizing the Gaussian to be unit normal. We now make the following observations:

- For any matrix $\mathbf{U} \in \mathbb{R}^{C-1 \times d}$ and a d-dimensional Gaussian random variable $z \sim \mathcal{N}(0, I_d) \in \mathbb{R}^d$, we have $\mathbf{U}^\top z \sim \mathcal{N}(0, \mathbf{U}\mathbf{U}^\top)$, i.e., an (C-1) -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 $\mathbf{U} = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2} \Big|_{\substack{i=1 \\ i \neq t}}^{C} \in \mathbb{R}^{(C-1) \times d}$, and obtain

$$\begin{aligned} p_\sigma^{\text{robust}} &= P_{r \sim \mathcal{N}(0, \mathbf{U}\mathbf{U}^\top)} \left[\bigcup_{i=1; i \neq t}^{C} r_i < \frac{g_i(\mathbf{x})}{\sigma \|\mathbf{u}_i\|_2} \right] \\ &= \text{CDF}_{\mathcal{N}(0, UU^\top)} \left(\frac{g_i(\mathbf{x})}{\sigma \|\mathbf{u}_i\|_2} \Big|_{\substack{i=1 \\ i \neq t}}^{C} \right) \end{aligned}$$

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

□

Lemma A.2. ***(Extension to non-Gaussian noise)*** *For high-dimensional data ($d \to \infty$), Lemma 1 generalizes to any coordinate-wise independent noise distribution that satisfies Lyapunov's condition.*

Proof. Applying Lyupanov's central limit theorem, given $\epsilon \sim \mathcal{R}$ is sampled from some distribution $\mathcal{R}$ to equation 2 in the previous proof, we have we have $\frac{\mathbf{u}}{\sigma\|\mathbf{u}\|_2}{}^\top \epsilon = \sum_{j=1}^{d} \frac{\mathbf{u}_j}{\sigma\|\mathbf{u}\|_2}\epsilon_j \xrightarrow{d} \mathcal{N}(0,1)$, which holds as long as the sequence $\{\frac{\mathbf{u}_j}{\|\mathbf{u}\|_2}\epsilon_j\}$ are independent random variables and satisfy the Lyapunov condition. In particular, this implies that $\mathbf{U}^\top z \sim \mathcal{N}(0, \mathbf{U}\mathbf{U}^\top)$, and the proof proceeds as similar to the Gaussian case after this step. □

Lemma A.3. ***(Extension to non-isotropic Gaussian)*** *Lemma 1 can be extended to the case of $\epsilon \sim \mathcal{N}(0, \mathcal{C})$ for an arbitrary positive definite covariance matrix $\mathcal{C}$:*

$$p_\sigma^{robust}(\mathbf{x}) = \Phi_{\mathbf{U}\mathcal{C}\mathbf{U}^\top}\left(\frac{c_i}{\|\mathbf{u}_i\|_2}\Big|_{\substack{i=1\\ i\neq t}}^{C}\right)$$

Proof. We observe that the Gaussian random variable $\frac{\mathbf{u}_i}{\|\mathbf{u}_i\|}{}^\top \epsilon|_{\substack{i=1\\ t\neq t}}^{C} = \mathbf{U}^\top\epsilon$ has mean zero as ϵ is mean zero. Computing its covariance matrix, we have $\mathbb{E}_\epsilon\, \mathbf{U}^\top\epsilon\epsilon^\top\mathbf{U} = \mathbf{U}^\top\, \mathbb{E}_\epsilon(\epsilon\epsilon^\top)\mathbf{U} = \mathbf{U}^\top\mathcal{C}\mathbf{U}$. We use this result after equation 2 in the proof of Lemma 1. □

Proposition A.1. *The **Taylor estimator** for the local robustness of a classifier f at point $\mathbf{x}$ with respect to target class t is given by linearizing f around $\mathbf{x}$ using a first-order Taylor expansion, with decision boundaries $g_i(\mathbf{x}) = f_t(\mathbf{x}) - f_i(\mathbf{x})$, $\forall i \neq t$, leading to*

$$p_\sigma^{taylor}(\mathbf{x}) = \Phi_{\mathbf{U}\mathbf{U}^\top}\left(\frac{g_i(\mathbf{x})}{\sigma\|\nabla_{\mathbf{x}} g_i(\mathbf{x})\|_2}\Big|_{\substack{i=1\\ i\neq t}}^{C}\right)$$

with $\mathbf{U}$ and Φ defined as in the linear case.

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

Proposition A.2. *The **estimation error** of the Taylor estimator for a classifier with a quadratic decision boundary $g_i(\mathbf{x}) = \mathbf{x}^\top A_i \mathbf{x} + \mathbf{u}_i^\top \mathbf{x} + c_i$ for positive-definite A_i, is upper bounded by*

$$|p_\sigma^{robust}(\mathbf{x}) - p_\sigma^{taylor}(\mathbf{x})| \leq k\sigma^{C-1} \prod_{\substack{i=1\\ i\neq t}}^{C} \frac{\lambda_{\max}^{A_i}}{\|\mathbf{u}_i\|_2}$$

for noise $\epsilon \sim \mathcal{N}(0, \sigma^2/d)$, in the limit of $d \to \infty$.

Proof. Without loss of generality, assume that $\mathbf{x} = 0$. For any other $\mathbf{x}_1 \neq 0$, we can simply perform a change of variables of the underlying function to center it at $\mathbf{x}_1$ to yield a different quadratic. We first write an expression for p_σ^{robust} for the given quadratic classifier $g_i(\mathbf{x})$ at $\mathbf{x} = 0$.

$$\begin{aligned} p_\sigma^{robust}(0) &= P_\epsilon \left(\bigcup_i g_i(\epsilon) > 0 \right) \\ &= P_\epsilon \left(\bigcup_i \mathbf{u}_i^\top \epsilon + c > -\epsilon^\top A_i \epsilon \right) \end{aligned}$$

Similarly, computing, p_σ^{taylor} we have $\nabla_{\mathbf{x}} g_i(0) = \mathbf{u}^\top$ and $g_i(0) = c_i$, resulting in

$$\begin{aligned} p_\sigma^{taylor}(0) &= P_\epsilon \left(\bigcup_i g_i^{taylor}(\epsilon) > 0 \right) \\ &= P_\epsilon \left(\bigcup_i \mathbf{u}_i^\top \epsilon + c > 0 \right) \end{aligned}$$

Subtracting the two, we have

$$\begin{aligned} &|p_\sigma^{robust}(0) - p_\sigma^{taylor}(0)| \\ &= \left| P \left(\bigcup_i 0 > \mathbf{u}_i^\top \epsilon + c > -\epsilon^\top A_i \epsilon \right) \right| \\ &= \left| P \left(\bigcup_i 0 > \frac{\mathbf{u}_i^\top \epsilon + c}{\sigma \|\mathbf{u}_i\|_2} > -\frac{\epsilon^\top A_i \epsilon}{\sigma \|\mathbf{u}_i\|_2} \right) \right| \end{aligned}$$

For high-dimensional Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2/d)$, with $d \to \infty$, we have that $\|\epsilon\|^2 = \sum_i \epsilon_i^2 \to \sigma^2$ from the law of large numbers. See [Vershynin, 2018] for an extended discussion. Thus we have $\epsilon^\top A \epsilon \leq \lambda_{\max}^A \|\epsilon\|^2 = \lambda_{\max}^A \sigma^2$.

Also let $z_i = \frac{\mathbf{u}_i^\top \epsilon + c}{\sigma \|\mathbf{u}_i\|_2}$ be a random variable. We observe that $z_i|_i$ is a tensor extension of z_i, has a covariance matrix of $\mathbf{U}\mathbf{U}^\top$ as before. Let us also define $\mathcal{C}_i = \frac{\lambda_{\max}^{A_i}}{\|\mathbf{u}_i\|_2}$.

$$
\begin{aligned}
&|p_\sigma^{robust}(0) - p_\sigma^{taylor}(0)| \\
&= \left| P\left(\bigcup_i 0 > z_i(\epsilon) > -\frac{\epsilon^\top A_i \epsilon}{\sigma \|\mathbf{u}_i\|_2} \right) \right| \\
&\leq \left| P\left(\bigcup_i 0 > z_i > -\frac{\lambda_{\max}^{A_i}}{\|\mathbf{u}_i\|_2} \sigma \right) \right| \quad (\epsilon^\top A \epsilon < \lambda_{\max}^A \sigma^2) \\
&= \left| \int \ldots \int_{-\mathcal{C}_i \sigma}^{0} \text{pdf}(z_i|_i)\, \mathrm{d}z_i|_i \right| \quad \text{(Defn of mvn cdf)} \\
&\leq \max_{z_i|_i} \text{pdf}(z_i|_i) \prod_i |C_i \sigma| \quad \text{(Upper bound pdf with its max)} \\
&\leq (2\pi)^{-(C-1)/2} \det(\mathbf{U}\mathbf{U}^\top)^{-1/2} \prod_{\substack{i=1 \\ i \neq t}}^{C} C_i \sigma \\
&= k \left(\sigma^{C-1} \prod_{\substack{i=1 \\ i \neq t}}^{C} \frac{\lambda_{\max}^{A_i}}{\|\mathbf{u}_i\|_2} \right)
\end{aligned}
$$

where $k = \max_z pdf(z) = (2\pi)^{-(C-1)/2} \det(\mathbf{U}\mathbf{U}^\top)^{-1/2}$, which is the max value of the Gaussian pdf. Note that as the rows of $\mathbf{U}$ are normalized, $\det(\mathbf{U}) \leq 1$ and $\det(\mathbf{U}\mathbf{U}^\top) = \det(\mathbf{U})^2 \leq 1$.

□

We note that these bounds are rather pessimistic, as in high-dimensions $\epsilon^\top A_i \epsilon \sim \lambda_{\text{mean}}^{A_i} \leq \lambda_{\max}^{A_i}$, and thus in reality the errors are expected to be much smaller.

Proposition A.3. *The **MMSE estimator** for the local robustness of a classifier f at point $\mathbf{x}$ with respect to target class t is given by an MMSE linearization f around $\mathbf{x}$, for decision boundaries $g_i(\mathbf{x}) = f_t(\mathbf{x}) - f_i(\mathbf{x})$, $\forall i \neq t$, leading to*

$$
\begin{aligned}
p_\sigma^{mmse}(\mathbf{x}) &= \Phi_{\mathbf{U}\mathbf{U}^\top} \left(\frac{\tilde{g}_i(\mathbf{x})}{\sigma \|\nabla_{\mathbf{x}} \tilde{g}_i(\mathbf{x})\|_2} \Big|_{\substack{i=1 \\ i \neq t}}^{C} \right) \\
\text{where } \tilde{g}_i(\mathbf{x}) &= \frac{1}{N} \sum_{j=1}^{N} g_i(\mathbf{x} + \epsilon)\ ,\ \epsilon \sim \mathcal{N}(0, \sigma^2)
\end{aligned}
$$

with $\mathbf{U}$ and Φ defined as in the linear case, and 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 $\mathbf{u} \in \mathbb{R}^d$ and $c \in \mathbb{R}$ such that

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

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

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

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

□

Proposition A.4. *The **estimation error** of the MMSE estimator for a classifier with a quadratic decision boundary $g(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} + \mathbf{u}^\top\mathbf{x} + c$, and positive definite A is upper bounded by*

$$
|p_\sigma^{robust}(\mathbf{x}) - p_\sigma^{mmse}(\mathbf{x})| \leq k\sigma^{C-1}\prod_{\substack{i=1\\ i\neq t}}^{C}\frac{|\lambda_{\max}^{A_i} - \lambda_{\text{mean}}^{A_i}|}{\|\mathbf{u}_i\|_2}
$$

for noise $\epsilon \sim \mathcal{N}(0, \sigma^2/d)$, in the limit of $d \to \infty$ and $N \to \infty$.

Proof. We proceed similarly to the proof made for the Taylor estimator, and without loss of generality, assume that $\mathbf{x} = 0$. Computing, p_σ^{mmse} we have $\mathbb{E}_\epsilon \nabla_{\mathbf{x}} g_i(\epsilon) = \mathbf{u}_i^\top$ and $\mathbb{E}_\epsilon\, g_i(\epsilon) = c + \mathbb{E}(\epsilon^\top A_i \epsilon) = c + \mathbb{E}(trace(\epsilon^\top A_i \epsilon)) = c + \mathbb{E}(trace(A_i \epsilon\epsilon^T)) = c + trace(A_i)\sigma^2/d = c + \sigma^2\lambda_{\text{mean}}^{A_i}$, resulting in

$$
\begin{aligned}
p_\sigma^{mmse}(0) &= P_\epsilon\left(\bigcup_i \hat{g}_i(\epsilon) > 0\right) \\
&= P_\epsilon\left(\bigcup_i \mathbf{u}_i^\top\epsilon + c > -\sigma^2\lambda_{\text{mean}}^{A_i}\right)
\end{aligned}
$$

Subtracting the two, we have

$$
\begin{aligned}
&|p_\sigma^{robust}(0) - p_\sigma^{mmse}(0)| \\
&\leq \left|P\left(\bigcup_i -\sigma^2\lambda_{\text{mean}}^{A_i} > \mathbf{u}_i^\top\epsilon + c > -\sigma^2\lambda_{\max}^{A_i}\right)\right| \\
&= \left|P\left(\bigcup_i -\sigma\frac{\lambda_{\text{mean}}^{A_i}}{\|\mathbf{u}_i\|_2} > \frac{\mathbf{u}_i^\top\epsilon + c}{\sigma\|\mathbf{u}_i\|_2} > -\sigma\frac{\lambda_{\max}^{A_i}}{\|\mathbf{u}_i\|_2}\right)\right|
\end{aligned}
$$

Similar to the previous proof, let $z_i = \mathbf{u}_i^\top\epsilon + c$ be a random variable, and that $z_i|_i$ is a tensor extension of z_i from our previous notation.

$$
\begin{aligned}
&|p_\sigma^{robust}(\mathbf{x}) - p_\sigma^{mmse}(\mathbf{x})| \\
&\leq \left| P\left(\bigcup_i -\lambda_{\text{mean}}^{A_i}\sigma^2 > z_i > -\lambda_{\max}^{A_i}\sigma^2 \right) \right| \\
&= \left| \int ... \int_{-\lambda_{\max}^{A_i}\sigma^2}^{-\lambda_{\text{mean}}^{A_i}\sigma^2} \text{pdf}(z_i|_i)\, \mathrm{d}z_i|_i \right| \\
&\leq \max_{z_i|_i} \text{pdf}(z_i|_i)\, \sigma^{C-1} \prod_i \frac{|(\lambda_{\max}^{A_i} - \lambda_{\text{mean}}^{A_i})|}{\|\mathbf{u}_i\|_2} \\
&= k\sigma^{C-1} \prod_i \frac{|\lambda_{\max}^{A_i} - \lambda_{\text{mean}}^{A_i}|}{\|\mathbf{u}_i\|2}
\end{aligned}
$$

where $k = \max_z \text{pdf}(z_i|_i) = (2\pi)^{-(C-1)/2} \det(\mathbf{U}\mathbf{U}^\top)^{-1/2}$ like in the Taylor case. Note that as the rows of $\mathbf{U}$ are normalized, $\det(\mathbf{U}) \leq 1$ and $\det(\mathbf{U}\mathbf{U}^\top) = \det(\mathbf{U})^2 \leq 1$.

□

We note that these bounds are rather pessimistic, as in high-dimensions $\epsilon^\top A_i \epsilon \sim \lambda_{\text{mean}}^{A_i} \leq \lambda_{\max}^{A_i}$, and thus in reality the errors are expected to be much smaller.

A.1.1 Approximating the Multivariate Gaussian CDF with mv-sigmoid

One drawback of the Taylor and MMSE estimators is their use of the *mvn-cdf*, which does not have a closed form solution and can cause the estimators to be slow for settings with a large number of classes C. In addition, the *mvn-cdf* makes these estimators non-differentiable, which is inconvenient for applications which require differentiating p_σ^{robust}. To alleviate these issues, we approximate the *mvn-cdf* with an analytical closed-form expression. As CDFs are monotonically increasing functions, the approximation should also be monotonically increasing.

To this end, it has been previously shown that the *univariate* Normal CDF ϕ is well-approximated by the sigmoid function [Hendrycks and Gimpel, 2016]. It is also known that when $\mathbf{U}\mathbf{U}^\top = I$, *mvn-cdf* is given by $\Phi(\mathbf{x}) = \prod_i \phi(\mathbf{x}_i)$, i.e., it is given by the product of the univariate normal CDFs. Thus, we may choose to approximate $\Phi(\mathbf{x}) = \prod_i \text{sigmoid}(\mathbf{x})$. However, when the inputs are small, this can be simplified as follows:

$$
\begin{aligned}
\Phi_I(\mathbf{x}) &= \prod_i \phi(\mathbf{x}_i) \approx \prod_i \frac{1}{1+\exp(-\mathbf{x}_i)} \\
&= \frac{1}{1 + \sum_i \exp(-\mathbf{x}_i) + \sum_{j,k} \exp(-\mathbf{x}_j - \mathbf{x}_k) + ...} \\
&\approx \frac{1}{1 + \sum_i \exp(-\mathbf{x}_i)} \quad (\text{for } \mathbf{x}_i \to \infty \ \forall i)
\end{aligned}
$$

We call the final expression the "multivariate sigmoid" (*mv-sigmoid*) which serves as our approximation of *mvn-cdf*, especially at the tails of the distribution. While we expect estimators using *mv-sigmoid* to approximate ones using *mvn-cdf* only when $\mathbf{U}\mathbf{U}^\top = \mathbf{I}$, we find experimentally that the approximation works well even for practical values of the covariance matrix $\mathbf{U}\mathbf{U}^\top$. Using this approximation to substitute *mv-sigmoid* for *mvn-cdf* in the p_σ^{taylor} and p_σ^{mmse} estimators yields the $p_\sigma^{\text{taylor\_mvs}}$ and $p_\sigma^{\text{mmse\_mvs}}$ estimators, respectively.

A.1.2 Relationship between mv-sigmoid, softmax, and the Taylor estimator

A common method to estimate the confidence of model predictions is to use the softmax function applied to the logits $f_i(\mathbf{x})$ of a model. We note that softmax is identical to *mv-sigmoid* when directly applied to the logits of neural networks:

$$\text{softmax}_t\left(f_i(\mathbf{x})\,\Big|_{i=1}^{C}\right) = \frac{\exp(f_t(\mathbf{x}))}{\sum_{i=1}^{C}\exp(f_i(\mathbf{x}))} = \frac{1}{1+\sum\limits_{\substack{i=1\\ i\neq t}}^{C}\exp(f_i(\mathbf{x})-f_t(\mathbf{x}))} = \text{mv-sigmoid}\left(g_i(\mathbf{x})\,\Big|_{\substack{i=1\\ i\neq t}}^{C}\right)$$

Recall that $g_i(\mathbf{x}) = f_t(\mathbf{x}) - f_i(\mathbf{x})$ is the decision boundary function. Note that this equivalence only holds for the specific case of logits. Comparing the expressions of softmax applied to logits above and the Taylor estimator, we notice that they are only different in that the Taylor estimator divides by the gradient norm, and uses the *mvn-cdf* function instead of *mv-sigmoid*. Given this similarity to the Taylor estimator, it is reasonable to ask whether softmax applied to logits (henceforth p_T^{softmax} for softmax with temperature T) itself can be a "good enough" estimator of p_σ^{robust} in practice. In other words, does p_T^{softmax} well-approximate p_σ^{robust} in certain settings?

In general, this cannot hold because softmax does not take in information about $\mathbf{U}\mathbf{U}^\top$, nor does it use the gradient information used in all of our estimators, although the temperature parameter T can serve as a substitute for σ in our expressions. In Appendix A.1, we provide a theoretical result for a restricted linear setting where softmax can indeed match the behavior of $p_\sigma^{\text{taylor\_mvs}}$, which happens precisely when $\mathbf{U}\mathbf{U}^\top = \mathbf{I}$ and all the class-wise gradients are equal. In the next section, we demonstrate empirically that the softmax estimator p_T^{softmax} is a poor estimator of average robustness in practice.

The softmax estimator 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.4. *For multi-class linear models $f(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} + b$, such that the decision boundary weight norms $\|\mathbf{u}_i\|_2 = k, \forall i \in [1, C], i \neq t$,*

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

Proof. 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*:

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

Next, by denoting $\mathbf{u}_i = \mathbf{w}_t - \mathbf{w}_i$, each row has equal norm $\|\mathbf{u}_i\|_2 = \|\mathbf{u}_j\|_2, \forall i, j, t \in [1, ...C]$ which implies:

$$\begin{aligned} p_\sigma^{\text{taylor\_mvs}}(\mathbf{x}) &= \text{mv-sigmoid}\left[\frac{g_i(\mathbf{x})}{\sigma\|\mathbf{u}_i\|_2}\,\Big|_{\substack{i=1\\ i\neq t}}^{C}\right] \\ &= \text{mv-sigmoid}\left[g_i(\mathbf{x})/T\,\Big|_{\substack{i=1\\ i\neq t}}^{C}\right] \quad (\because T = \sigma k) \\ &= p_T^{\text{softmax}}(\mathbf{x}) \end{aligned}$$

□

Lemma A.4 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 DATASETS

The MNIST dataset consists of images of gray-scale handwritten digits spanning 10 classes: digits 0 through 9. The FashionMNIST (FMNIST) dataset consists of gray-scale images of articles of clothing spanning 10 classes: t-shirt, trousers, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. For MNIST and FMNIST, each image is 28 pixels x 28 pixels. For MNIST and FMNIST, 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 spanning 10 classes: airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck. The CIFAR100 dataset consists of color images of common objects and animals spanning 100 classes: apple, bowl, chair, dolphin, lamp, mouse, plain, rose, squirrel, train, etc. For CIFAR10 and CIFAR100, each image is 3 pixels x 32 pixels x 32 pixels. For CIFAR10 and CIFAR100, the training set consists of 50,000 images and the test set consists of 10,000 images.

A.3 MODELS

For the MNIST and FMNIST, 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, we train a Vision Transformer model to perform 10-class and 100-class classification, respectively, by fine-tuning a Vision Transformer that was pre-trained on ImageNet (`https://huggingface.co/google/vit-base-patch16-224-in21k`) on each dataset. For these models, the test set consists of 100 images. We chose this number of datapoints so that p_σ^{mc} would run within a reasonable amount of time. We also train a ResNet18 model to perform 10-class and 100-class classification, respectively. The model architecture is described in [He et al., 2016]. For CIFAR10 and CIFAR100, we also 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 for each model is shown in Table 2.

EXPERIMENTS

Due to file size constraints, Section A.4 can be found in the Supplementary material.

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

Table 2: Test set accuracy of models.

A.4 EXPERIMENTS

In this section, we provide the following additional experimental results:

1. Figure 5 shows results on the convergence of p_σ^{mc}. p_σ^{mc} takes a large number of samples to converge and is computationally inefficient.
2. Figure 6 shows results on the convergence of p_σ^{mmse}. p_σ^{mmse} takes only a few samples to converge and is more computationally inefficient than p_σ^{mc}.
3. Figure 7 shows the distribution of p_σ^{robust} as a function of σ. Consistent with theory in Section 3, (1) as noise increases, p_σ^{robust} decreases, and (2) p_σ^{mmse} accurately estimates p_σ^{mc}.
4. Table 3 presents estimator runtimes. Our analytical estimators are more efficient than the naïve estimator (p_σ^{mc}).
5. Figure 8 shows the accuracy of the analytical robustness estimators as a function of σ. p_σ^{mmse} and $p_\sigma^{\text{mmse\_mvs}}$ are the best estimators of p_σ^{robust}, followed closely by $p_\sigma^{\text{taylor\_mvs}}$ and p_σ^{taylor}, trailed by p_T^{softmax}.
6. Figure 9 shows the accuracy of the analytical estimators for robust models. For more robust models, the estimators compute p_σ^{robust} more accurately over a larger σ.
7. Figures 10 and 11 shows that *mv-sigmoid* well-approximates *mvn-cdf* over σ.
8. Figure 12 shows that p_T^{softmax} is not a good approximator of p_σ^{robust}.
9. Figure 13 shows the distribution of p_σ^{robust} among classes (measured by p_σ^{mmse}), revealing that models display robustness bias among classes.
10. Figures 14 and 15 show the application of p_σ^{mmse} and p_T^{softmax} to identification of robust and non-robust points. p_σ^{robust} better identifies robust and non-robust points than p_T^{softmax}.
11. Figures 16, 17, 18, and 19 show examples of noisy images with the level of noise analyzed in our paper. Overall, the noise levels seem visually significant.

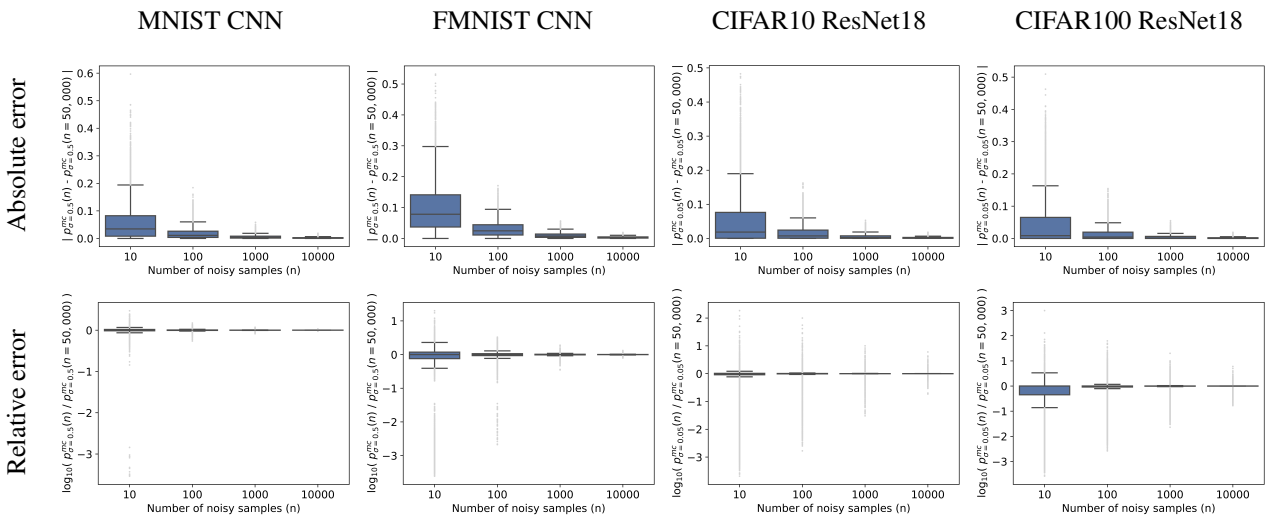


Figure 5: Convergence of p_σ^{mc}. In practice, p_σ^{mc} takes around $n = 10,000$ samples to converge and is computationally inefficient.

A.4.1 p_σ^{robust} identifies images that are robust to and images that are vulnerable to random noise

For each dataset, we train a simple CNN to distinguish between images with high and low p_σ^{mmse}. We train the same CNN to also distinguish between images with high and low p_T^{softmax}. The CNN consists of two convolutional layers and two fully-connected feedforward layers with a total of 21,878 parameters. For a given dataset, for each class, we take the images with the top-25 and bottom-25 p_σ^{mmse} values. This yields 500 images for CIFAR10 (10 classes x 50 images per class) and 5,000 images for CIFAR100 (100 classes x 50 images per class). We also perform the same steps using p_T^{softmax}, yielding

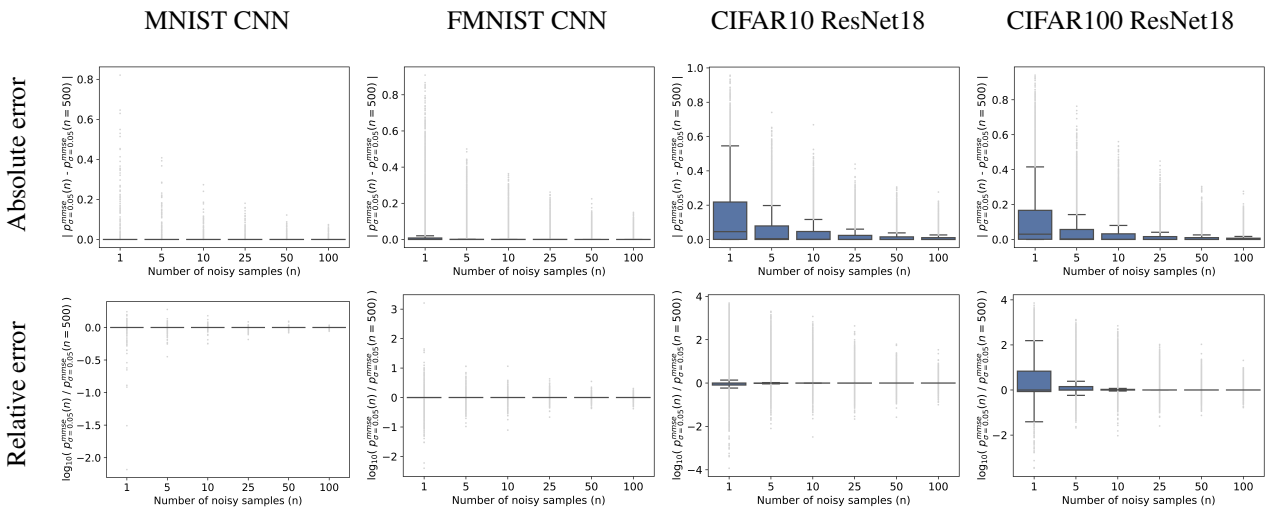
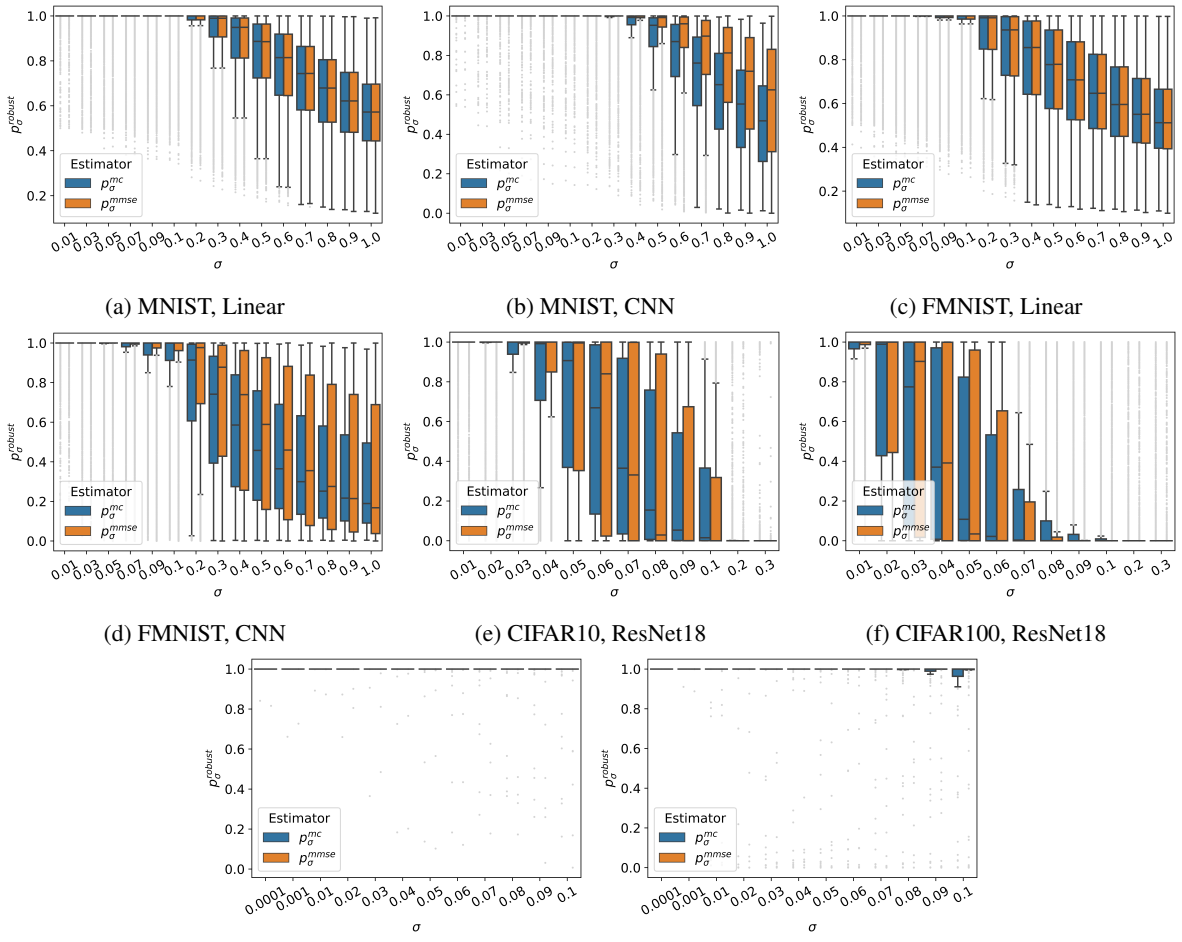


Figure 6: Convergence of p_σ^{mmse}. In practice, p_σ^{mmse} takes around $n = 5{-}10$ samples to converge and is more computationally efficient than p_σ^{mc}.

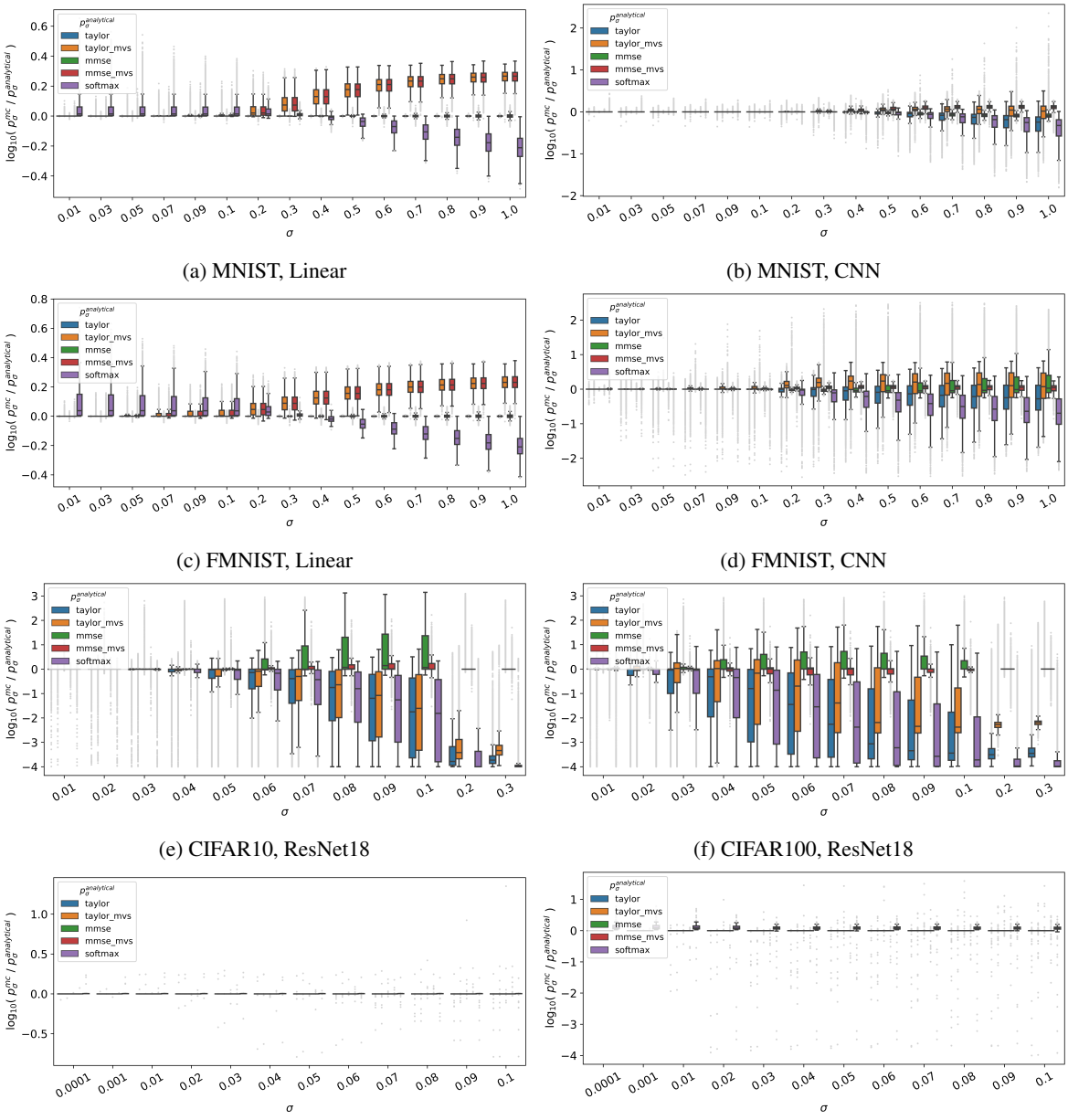


(a) MNIST, Linear (b) MNIST, CNN (c) FMNIST, Linear

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

(g) CIFAR10, Vision Transformer (h) CIFAR100, Vision Transformer

Figure 7: Distribution of p_σ^{robust} over σ. As noise increases, p_σ^{robust} decreases. In addition, p_σ^{mmse} accurately estimates p_σ^{mc}.



(a) MNIST, Linear

(b) MNIST, CNN

(c) FMNIST, Linear

(d) FMNIST, CNN

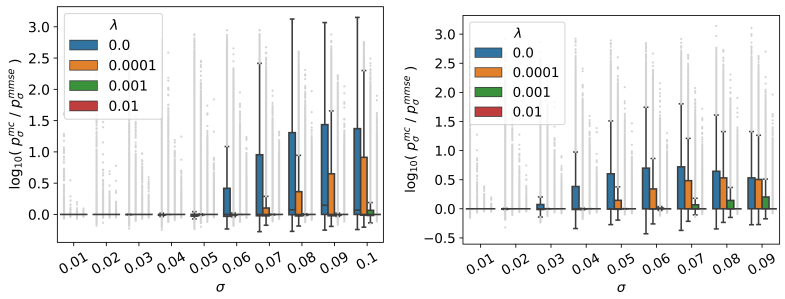
(e) CIFAR10, ResNet18

(f) CIFAR100, ResNet18

(g) CIFAR10, Vision Transformer

(h) CIFAR100, Vision Transformer

Figure 8: Accuracy of p_σ^{robust} estimators over σ. The smaller the noise neighborhood σ, the more accurately the estimators compute p_σ^{robust}. p_σ^{mmse} and $p_\sigma^{\text{mmse\_mvs}}$ are the best estimators of p_σ^{robust}, followed closely by $p_\sigma^{\text{taylor\_mvs}}$ and p_σ^{taylor}, trailed by p_T^{softmax}.



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

Figure 9: Accuracy of p_σ^{robust} estimators over σ for robust models. For more robust models, the estimators compute p_σ^{robust} more accurately over a larger σ.

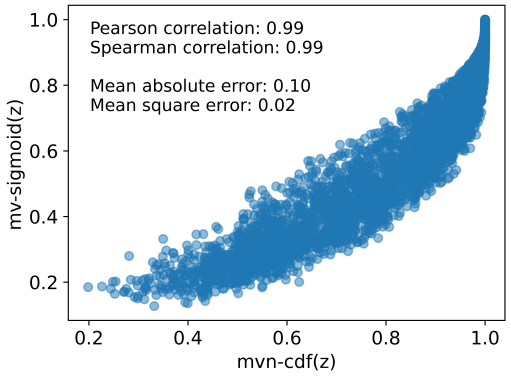


Figure 10: Correlation of *mvn-cdf(z)* and *mv-sigmoid(z)* for the CIFAR10 ResNet18 model. The formulation of z is described in Section 4.1. In practice, *mv-sigmoid* approximates *mvn-cdf* well.

another 500 images for CIFAR10 and another 5,000 images for CIFAR100. For each dataset, the train/test split is 90%/10% of points.

Then, we compare the performance of the two models. For CIFAR10, the test set accuracy for the p_σ^{mmse} CNN is 0.92 while that for the p_T^{softmax} CNN is 0.58. For CIFAR100, the test set accuracy for the p_σ^{mmse} CNN is 0.74 while that for the p_T^{softmax} CNN is 0.55. The higher the test set accuracy of a CNN, the better the CNN distinguishes between images. Thus, the results indicate that p_σ^{robust} better identifies images that are robust to and vulnerable to random noise than p_T^{softmax}.

We also provide additional visualizations of images with the highest and lowest p_σ^{robust} and images with the highest and lowest p_T^{softmax}.

A.4.2 Softmax probability is not a good proxy for average-case robustness

To examine the relationship between p_σ^{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.4 (Figure 12). For a non-robust model, p_σ^{robust} and p_T^{softmax} are not strongly correlated (Figure 12a). As model robustness increases, the two quantities become more correlated (Figures 12b and 12c). However, even for robust models, the relationship between the two quantities is mild (Figure 12c). That p_σ^{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_σ^{robust}.

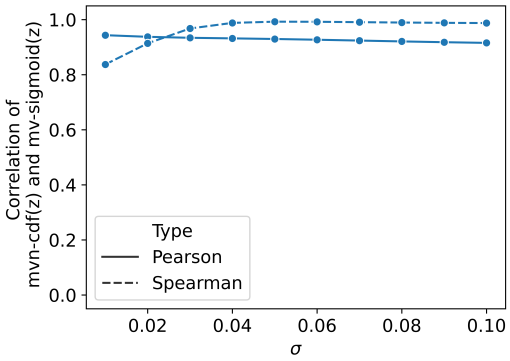
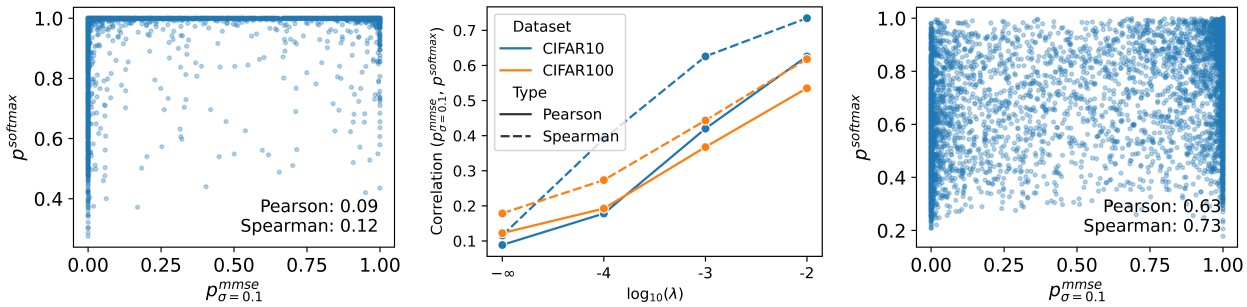


Figure 11: mv-sigmoid's approximation of mvn-cdf over σ. mv-sigmoid well-approximates mvn-cdf over σ.



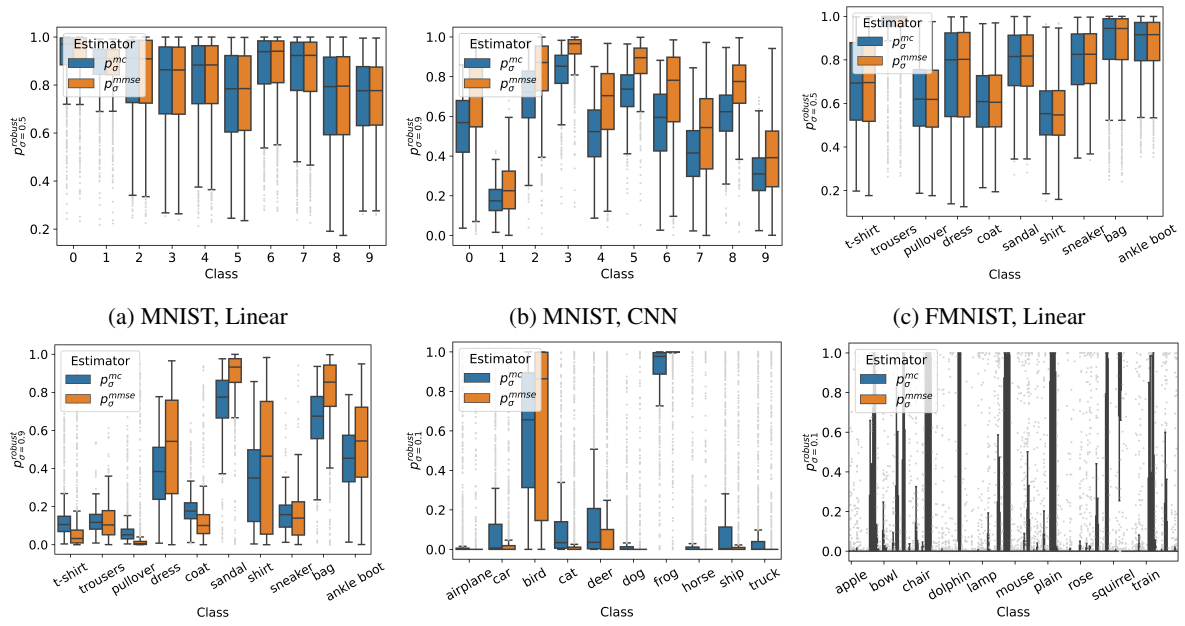
(a) CIFAR10
Non-robust model ($\lambda = 0$)

(b) CIFAR10 and CIFAR100
Varying model robustness

(c) CIFAR10
Robust model ($\lambda = 0.01$)

Figure 12: Relationship between p_σ^{robust} and p_T^{softmax} for CIFAR10 and CIFAR100 ResNet18 models. (a) For a non-robust model, p_σ^{robust} and p_T^{softmax} are not strongly correlated. (b) As model robustness increases, the two quantities become more correlated. (c) However, even for robust models, the relationship between p_σ^{robust} and p_T^{softmax} is mild. Together, 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.

While working on the paper, we hypothesized that p_σ^{robust} (e.g., p_σ^{mmse}) might be correlated with model accuracy. However, we did not find this in practice. Instead, what we find is that p_σ^{robust} succeeds in identifying canonical data points of a class, and does so much better than p_T^{softmax}. We first assess this finding through visual inspection, finding that images with higher p_σ^{robust} tend to be more canonical and clear images, and that this distinction is less apparent for p_T^{softmax} (Figures 3 and 14). We then use a model to classify these images as an additional, more objective assessment of this pattern (as discussed in Section 4.3).



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

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

Figure 13: Local robustness bias among classes. 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.

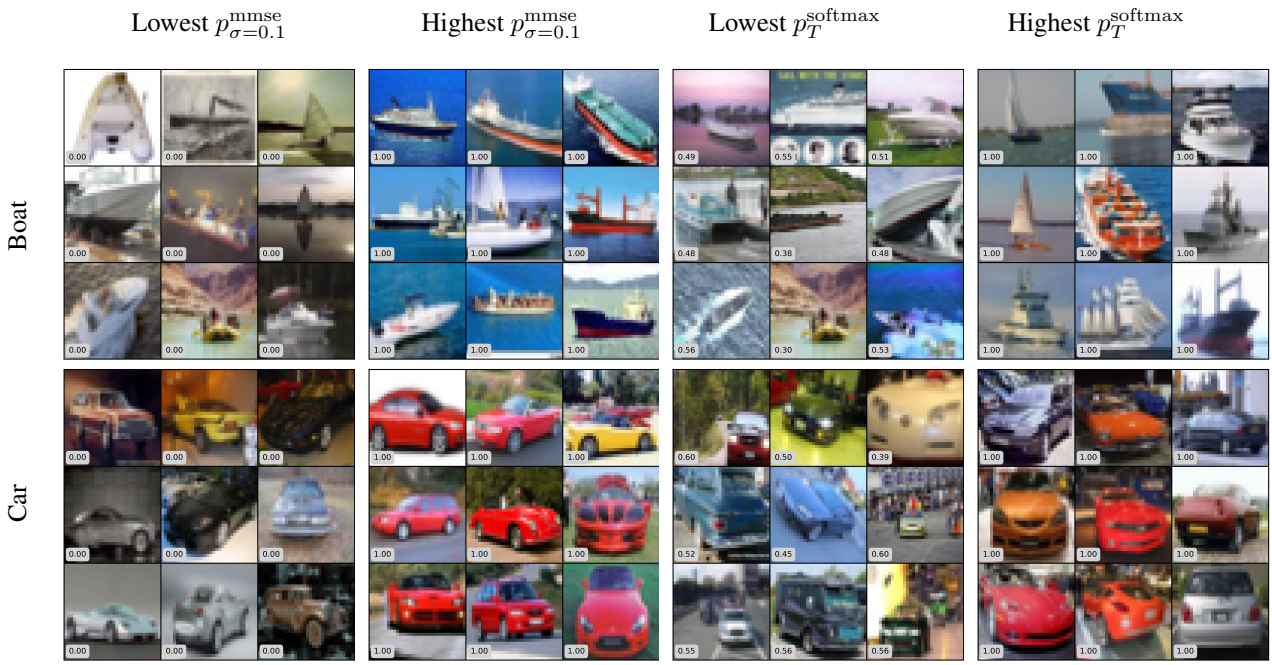


Figure 14: Additional images with the lowest and highest p_σ^{robust} and p_T^{softmax} values among CIFAR10 classes. Images with high p_σ^{robust} tend to be brighter and have stronger object-background contrast (making them more robust to random noise) than those with low p_σ^{robust}. The difference between images with high and low p_T^{softmax} is less clear. Thus, p_σ^{robust} better captures the model's local robustness with respect to an input than p_T^{softmax}.

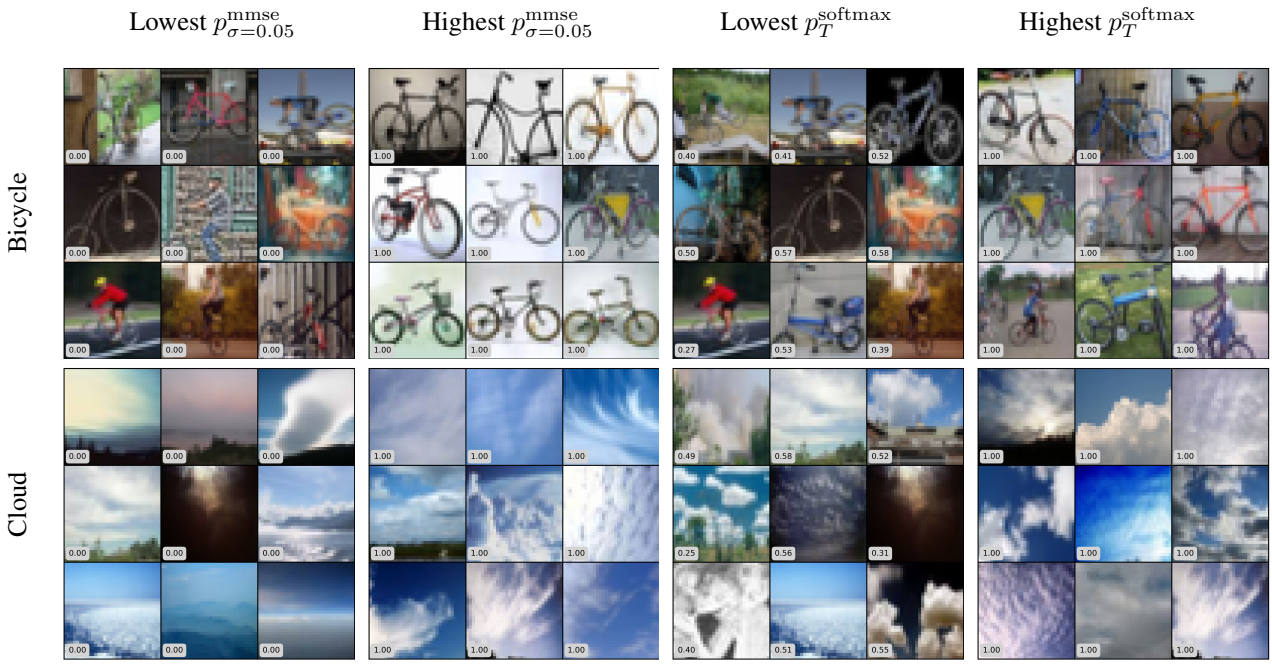


Figure 15: Images with the lowest and highest $p^{\text{robust}}_{\sigma}$ and p^{softmax}_{T} values among CIFAR100 classes. Images with high $p^{\text{robust}}_{\sigma}$ tend to be brighter and have stronger object-background contrast (making them more robust to random noise) than those with low $p^{\text{robust}}_{\sigma}$. The difference between images with high and low p^{softmax}_{T} is less clear. Thus, $p^{\text{robust}}_{\sigma}$ better captures the model's local robustness with respect to an input than p^{softmax}_{T}.

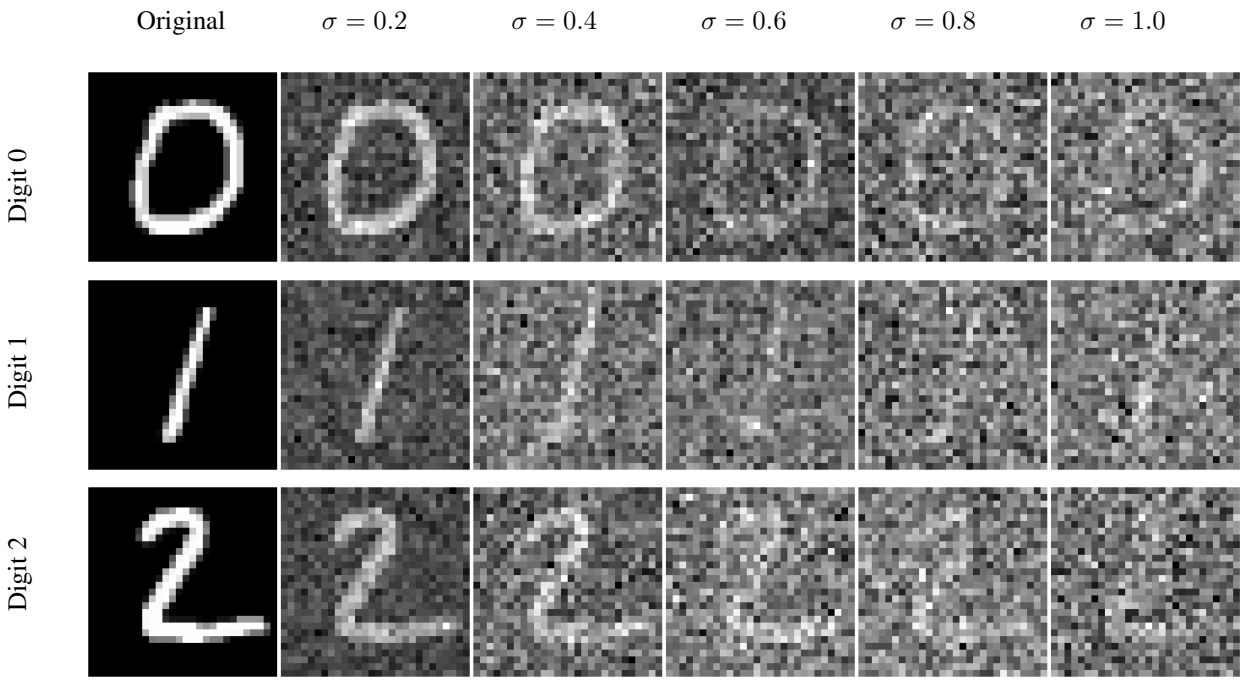


Figure 16: Examples of noisy images for MNIST.

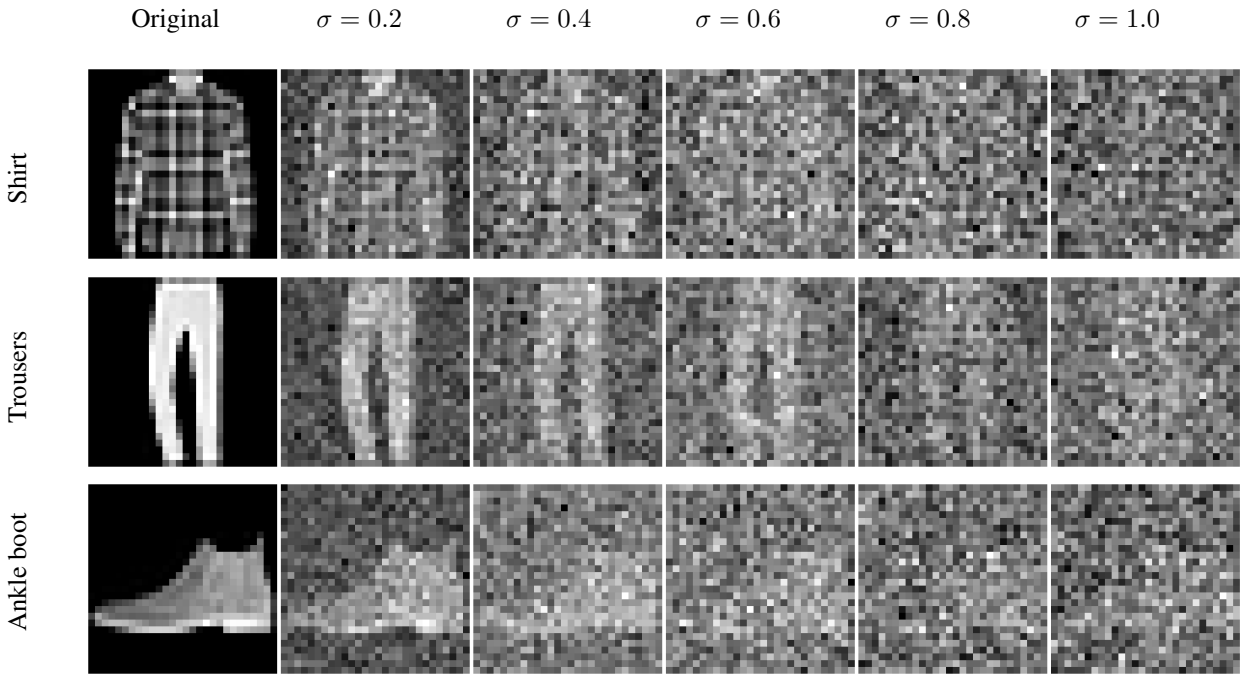


Figure 17: Examples of noisy images for FMNIST.

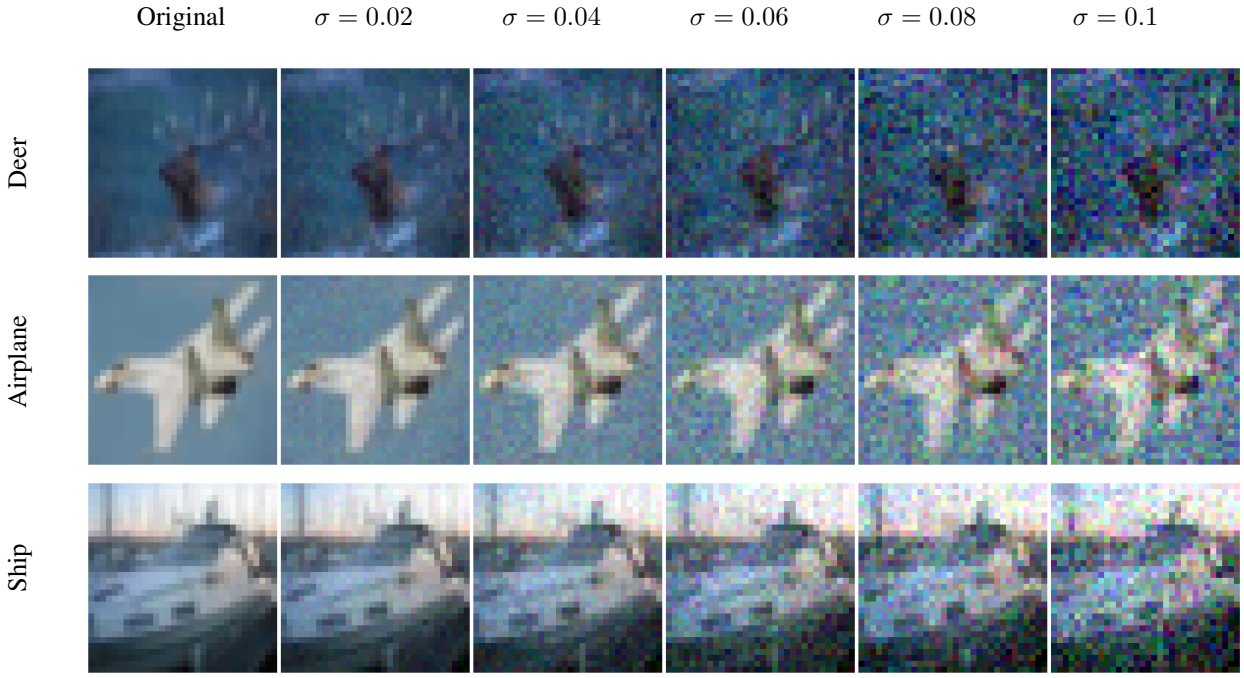


Figure 18: Examples of noisy images for CIFAR10.

		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. Runtimes are in the format of hour:minute:second. 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}).

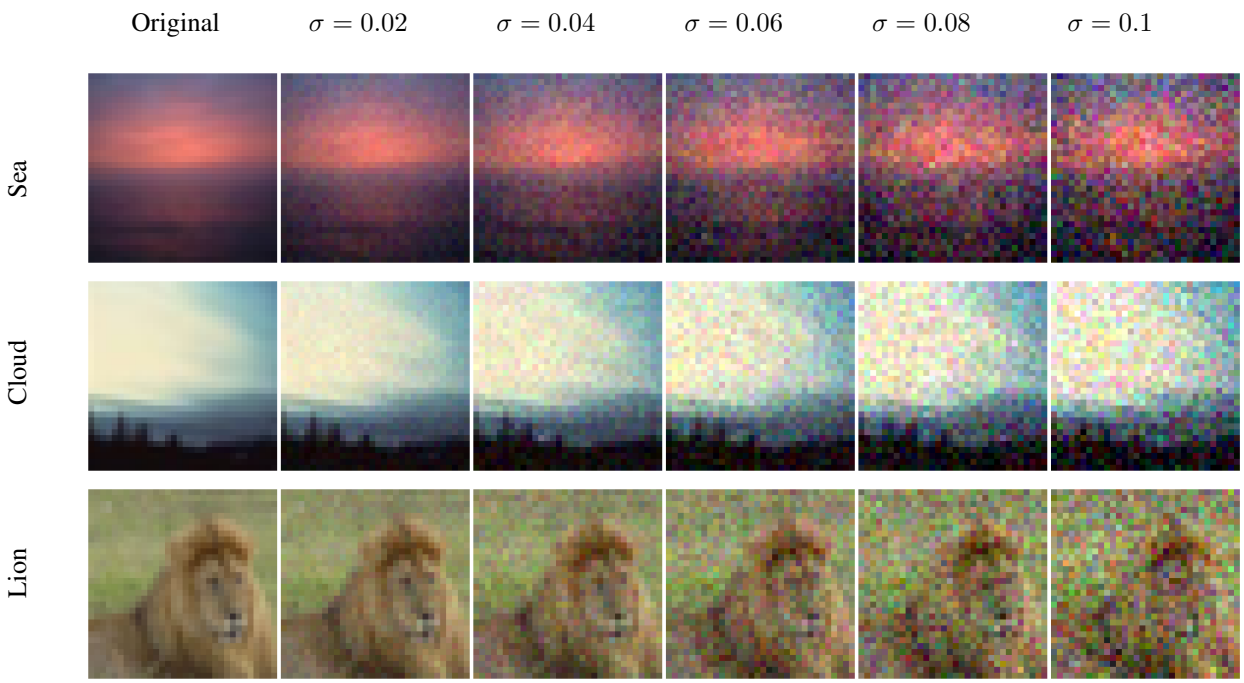


Figure 19: Examples of noisy images for CIFAR100.