PIGEON: Flying Beyond Strong Neuron Activation Coverage

Juheon Kang (G202548001)
2715wngjs@uos.ac.kr

Hyunseo Shin (G202448003)
hseo98@uos.ac.kr

Eunkyung Choi (G202448018)
rmarud202@uos.ac.kr

June 19, 2025

Abstract

Neuron Boundary Coverage (NBC) has been widely adopted to evaluate the test adequacy of deep neural networks. However, NBC suffers from key limitations, such as its sensitivity to activation outliers and lack of awareness of activation distributions. In this work, we propose **Pigeon**, a probabilistic refinement of NBC by incorporating confidence-based thresholds derived from neuron-wise statistics. Our variants offer more robust and distribution-aware assessments of internal neuron behavior. Empirical results across 147 model-dataset pairs demonstrate stronger correlation with established robustness metrics and reveal subtle model behaviors that traditional metrics often overlook. Our code is publicly available at this GitHub link.

1 Introduction

Deep Neural Networks (DNNs) have demonstrated impressive performance across a wide range of tasks, yet their deployment in safety-critical applications demands rigorous testing and validation. The non-linear and high-dimensional nature of DNNs limits the effectiveness of traditional software testing methodologies, necessitating alternative evaluation strategies.

Inspired by classical software engineering, where code coverage serves as a proxy for test adequacy, researchers have introduced neuron coverage metrics to quantify how thoroughly a network's internal states are exercised by input data. These structural metrics aim to provide insight into model behavior and potential vulnerabilities.

Among these, Neuron Coverage (NC), k-Multisection Neuron Coverage (KMNC), and Neuron Boundary Coverage (NBC) are widely used. However, previous studies have revealed that such metrics often exhibit weak and inconsistent correlation with adversarial robustness, raising concerns about their reliability for safety evaluation.

We revisit Strong Neuron Activation Coverage (SNAC), a variant of NBC that tracks whether a neuron's activation exceeds its training-time maximum. While SNAC is particularly relevant for capturing behaviors triggered by adversarial attacks (e.g., FGSM, Carlini & Wagner), its reliance on a single maximum value makes it sensitive to outliers and noisy spikes.

To address these limitations, we propose confidence-thresholded variants of SNAC based on neuron-wise activation distributions. By grounding thresholds in statistical confidence intervals, our method more accurately reflects the rarity of over-activations and shows improved correlation with robustness metrics. Through extensive experiments on adversarially retrained models, we demonstrate that our refined coverage metrics offer more reliable insights into model robustness.

2 Related Work

2.1 Neuron Coverage Metrics

Metric Definitions
Various neuron coverage metrics have been proposed to assess the testing adequacy of deep neural networks (DNNs). Neuron Coverage (NC) [6] measures the proportion of neurons activated beyond a predefined threshold. k-Multisection Neuron Coverage (KMNC) [5] extends this idea by partitioning

each neuron's activation range into k intervals to capture finer activation diversity. Neuron Boundary Coverage (NBC) [5] and its variant, Strong Neuron Activation Coverage (SNAC), focus on whether activations exceed training-time lower or upper bounds, respectively.

Limitations and Robustness
Despite their intuitive appeal, recent studies have questioned the robustness relevance of these metrics. Harel-Canada et al. [4] and Yang et al. [9] show that higher coverage scores do not guarantee improved fault detection and often correlate with unnatural input behaviors. Dong et al. [2] further demonstrate weak and inconsistent correlation between neuron coverage (including NBC) and adversarial robustness, suggesting that coverage is more sensitive to input diversity than actual model vulnerabilities.

We build upon Dong et al.'s analysis [2], which critically examines NC, KMNC, and NBC using rigorous experimental protocols. Their test configurations and evaluation methodology directly inform our work. Motivated by their findings, we propose confidence-thresholded SNAC variants that consider neuron-wise activation distributions, aiming to overcome the statistical fragility of maximum-based thresholds.

2.2 Robustness Metrics

Robustness metrics such as Lipschitz constants [8] and CLEVER scores [7] aim to quantify model sensitivity to input perturbations.

Lipschitz constant The global Lipschitz constant L provides an upper bound on the output variation for any pair of inputs:

$$\|f(x) - f(x')\| \leq L \cdot \|x - x'\|$$

Lower values of L indicate greater robustness.

CLEVER Score CLEVER(Cross-Lipschitz Extreme Value for nEtwork Robustness) improves upon global estimates by computing a local Lipschitz constant around a given input x. Letting $f_c(x)$ denote the logit for the true class c and $f_k(x)$ for any $k \neq c$, the CLEVER score is defined as:

$$\text{CLEVER}(x) = \frac{f_c(x) - \max_{k \neq c} f_k(x)}{L_{\text{local}}}$$

Here, L_{local} is estimated via sampling perturbations δ within an ℓ_p-ball of radius R:

$$L_{\text{local}} \approx \max_{\|\delta\| \leq R} \frac{\|f(x+\delta) - f(x)\|}{\|\delta\|}$$

Higher CLEVER scores indicate that larger perturbations are required for misclassification, suggesting stronger robustness. These robustness metrics provide valuable baselines for evaluating the effectiveness of coverage-based metrics.

2.3 Adversarial Attack

Fast Gradient Sign Method (FGSM) [3] generates adversarial examples by adding perturbations along the gradient direction. The Carlini & Wagner (C&W) attack [1] uses optimization to craft stronger, less perceptible perturbations. These attacks reveal the vulnerability of DNNs to small input changes.

3 Preliminaries

3.1 Neuron Boundary Coverage (NBC)

Neuron Boundary Coverage (NBC) quantifies the extent to which neuron activations fall into extreme boundary regions, as defined by the activation statistics observed during training. For each neuron n, let low_n and high_n be its minimum and maximum activation values, respectively. The boundary regions are defined as:

- Upper region: $[\text{high}_n, \infty)$

- Lower region: $(-\infty, \text{low}_n]$

Let UCN be the set of neurons whose activations exceed high_n on any test input, and LCN be the set whose activations fall below low_n. NBC is then defined as:

$$NBC = \frac{|UCN| + |LCN|}{2 \times |N|} \tag{1}$$

where $|N|$ is the total number of neurons in the network. NBC captures how frequently the network enters activation regimes it has not seen during training.

3.2 Strong Neuron Activation Coverage (SNAC)

SNAC is a simplified variant of NBC that focuses solely on upper-bound violations. It measures the proportion of neurons activated beyond their maximum training value:

$$SNAC = \frac{|UCN|}{|N|} \tag{2}$$

This metric reflects how many neurons are pushed into abnormally high activation states, which is often correlated with model fragility under input perturbations, particularly when using ReLU activations.

3.3 Why Measure Neuron Over-Activation?

Excessive neuron activation can indicate anomalous model behavior triggered by distributional shifts or adversarial perturbations. Coverage metrics such as NBC and SNAC aim to detect such behaviors by identifying activations that deviate from training-time norms.

While Dong et al. [2] observed that structural neuron coverage shows limited correlation with robustness, their analysis does not negate the usefulness of such metrics. Instead, it underscores the need for more statistically grounded formulations—a direction we pursue by refining SNAC with confidence-based thresholds.

3.4 Adversarial Attacks Considered

To evaluate the robustness relevance of our metric, we examine two canonical adversarial attacks:

Fast Gradient Sign Method (FGSM) FGSM [3] is a single-step attack that perturbs an input x in the direction of the gradient of the loss function:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y)) \tag{3}$$

where ϵ controls the perturbation magnitude, and J is the loss function with respect to true label y. FGSM is efficient and often leads to sharp neuron activations due to its direct alignment with gradient directions.

Carlini & Wagner (C&W) Attack The C&W attack [1] formulates adversarial example generation as an optimization problem:

$$\arg\min_{x'} \|x' - x\|_p + \lambda \cdot f(x', t) \tag{4}$$

where $f(x', t)$ encourages misclassification into a target label t, λ balances distortion and success, and p denotes the chosen norm (L_0, L_2, or L_∞). C&W is known for producing high-confidence, low-visibility perturbations that can bypass many standard defenses.

4 Distribution-Aware Neuron Coverage

To overcome the limitations of existing neuron coverage metrics such as NBC and SNAC, which rely solely on fixed activation boundaries, we introduce **Pigeon**, Distribution-Aware Neuron Coverage. This metric characterizes each neuron's activation behavior based on its empirical distribution, enabling a more statistically grounded notion of coverage.

4.1 Pre-Activation Distribution

Pigeon uses the *pre-activation* values of neurons—i.e., the inputs to nonlinear functions such as ReLU—rather than post-activation outputs. This is because ReLU eliminates negative values, restricting the observable distribution to only non-negative activations. As a result, distributional information is lost. Pre-activation values preserve the full dynamic range of a neuron's response, enabling more accurate modeling of activation behavior using statistical methods. We assume that the distribution of pre-activation values for each neuron approximately follows a Gaussian distribution. This is a reasonable approximation in networks using batch normalization or standardized initializations (e.g., LeNet).

4.2 Confidence Interval Estimation

Given a set of profiling inputs, we compute the empirical mean μ_i and standard deviation σ_i of the pre-activation values z_i for each neuron i:

$$\mu_i = \mathbb{E}[z_i], \quad \sigma_i = \sqrt{\mathbb{E}[(z_i - \mu_i)^2]}.$$

Since ReLU eliminates negative outputs, we focus on the upper tail of the pre-activation distribution. For each neuron i, we define the one-sided confidence bounds as follows:

- 95% upper bound (one-sided Gaussian): $\mu_i + 1.645\sigma_i$
- 99% upper bound (one-sided Gaussian): $\mu_i + 2.326\sigma_i$
- 99.99% upper bound (one-sided Gaussian): $\mu_i + 3.891\sigma_i$

These thresholds define the boundary for statistically rare (high) activations for each neuron.

4.3 Definition

Given a test dataset, a neuron i is considered *covered* if at least one test input x results in a pre-activation $z_i(x)$ exceeding its upper confidence bound. Formally:

$$\text{Pigeon}_i(x) = \begin{cases} 1 & \text{if } z_i(x) > \mu_i + k\sigma_i, \\ 0 & \text{otherwise} \end{cases}$$

where k = 1.645 for 95% confidence, k = 2.326 for 99% confidence, or k = 3.891 for 99.9% confidence.

The overall Pigeon is computed as the proportion of neurons that are covered by at least one test input:

$$\text{Pigeon} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\exists x \in \mathcal{D}_{\text{test}} : \text{Pigeon}_i(x) = 1\right],$$

where N is the total number of neurons and $\mathbb{I}[\cdot]$ is the indicator function.

This approach allows us to measure how well test inputs cover unusual or atypical neuron behaviors, thereby improving the sensitivity of coverage metrics in detecting corner-case activations compared to NBC.

4.4 Interpretation

The Pigeon metric reflects how frequently test inputs elicit statistically rare pre-activation values across neurons. A high or low Pigeon score can indicate different underlying characteristics of the model, training data, or test data distribution.

High Pigeon score:

- The training data used for the model may have a narrow or well-normalized distribution, leading to tightly bounded pre-activation statistics.
- The model may exhibit reduced robustness, with many neurons prone to over-activation when exposed to unfamiliar or out-of-distribution inputs.
- The test dataset may differ significantly from the training distribution, potentially containing inputs that stimulate atypical neuron behavior.

Low Pigeon score:

- The model may have been trained on a dataset with broad or diverse input distributions, covering a wide range of neuron activations during training.
- The model may be more robust, exhibiting stable and bounded neuron responses to test inputs.
- The test dataset may be well-normalized or closely aligned with the training distribution.

In practice, Pigeon scores can serve as lightweight indicators of distributional mismatch or robustness degradation.

5 Experiments

Our overall experiment pipeline is illustrated in Figure 1.

5.1 Adversarial Sample Generation via SNAC and FGSM-Based Neuron Boundary Perturbation

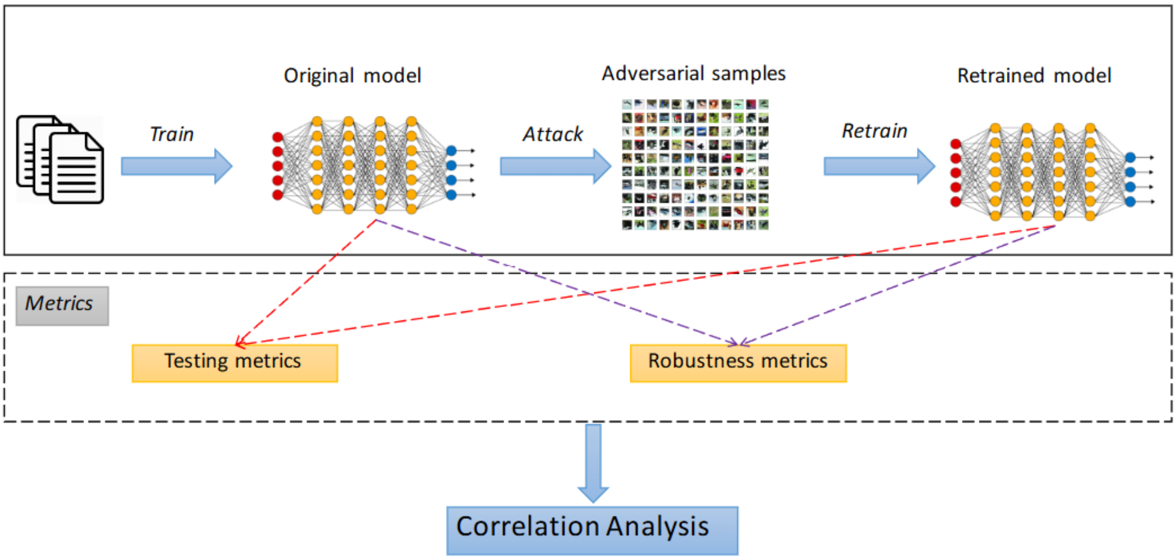


Figure 1: Overview of the experiment pipeline.

5.1.1 Objective

In this experiment, we investigate the behavior of internal neuron activations in a trained neural network and systematically generate adversarial inputs that exceed previously observed neuron activation upper boundaries. This follows the framework of SNAC, which evaluates how extensively individual neurons are activated during training and guides adversarial generation into unexplored activation regions.

5.1.2 Motivation

We hypothesize that once an input achieves maximal activation for a given neuron, slight perturbations using FGSM can often increase its activation beyond the previous maximum. This enables us to expose the model to novel activation states and serves both as a white-box testing method and a robustness evaluation technique.

Moreover, since SNAC is computed by counting the number of neurons whose activations exceed their previously observed maximum values, the SNAC score can be artificially inflated by generating additional boundary-crossing adversarial samples. In other words, by controlling the proportion of adversarial inputs that cross neuron boundaries, the overall SNAC score can be arbitrarily adjusted, thereby revealing its inherent vulnerability as a coverage metric.

5.1.3 Methodology

The experiment consists of the following steps:

1. **Model Setup and Hook Registration:** We utilize a pretrained LeNet model trained on the MNIST dataset. Forward hooks are registered to extract activation values from internal layers (`conv1`, `conv2`, `fc1`, `fc2`, `fc3`) during inference.

2. **Maximum Neuron Activation Collection:** The entire training set is processed through the model. For each neuron in the target layers, we record the maximum activation value observed across all samples, which serves as its SNAC boundary.

3. **Adversarial Sample Generation:** For each neuron, we identify the input sample that originally produced its maximum activation. Starting from these samples, we iteratively apply a gradient-based attack to further increase the neuron activation.

   Specifically, we optimize the following objective for neuron i at layer l:

   $$\mathcal{L} = -a_i^{(l)}(x)$$

   where $a_i^{(l)}(x)$ denotes the activation value of neuron i at layer l given input x. This encourages the neuron activation to increase as much as possible.

   The attack updates follow an FGSM-like rule:

   $$x_{t+1} = \text{clip}\left(x_t + \alpha \cdot \text{sign}\left(\nabla_x \mathcal{L}\right)\right)$$

   where α is the step size and the perturbation is projected within the ϵ-ball around the original input. The perturbed inputs are clipped to remain within the normalized data range of MNIST.

4. **Evaluation of Boundary Crossing:** After generating the adversarial examples, we verify whether the neuron activations successfully exceed their original SNAC boundaries. The boundary-crossing success rate is then computed as our evaluation metric.

5.2 Adversarial Experiment

5.2.1 Dataset Construction

We constructed 21 datasets based on the MNIST dataset by applying FGSM and C&W adversarial attacks to the LeNet model. Following the experimental setup of Dong et al. [2], we adopted their recommended hyperparameters for both attack methods to ensure comparability and reproducibility. The resulting datasets are summarized in Table 1, and the detailed attack parameters are provided in the Appendix 3.

5.2.2 Model Retraining

We trained seven models based on different dataset configurations as summarized in Table 2. Model **a** corresponds to the original model trained solely on the MNIST dataset without any adversarial examples. The remaining models (**b** to **g**) were obtained by retraining model **a** on adversarially-augmented datasets with different configurations and fine-tuning epochs.

5.2.3 Neuron Activation Statistics and SNAC Evaluation Results

For each of the seven trained models described in Section 2, we extracted neuron-wise activation statistics using their corresponding training datasets. We computed the following statistics for all neurons across the five layers (`conv1`, `conv2`, `fc1`, `fc2`, `fc3`):

- The maximum activation value.
- The mean activation value.
- The standard deviation of activations.
- Confidence-based boundaries computed as $\mu + z \cdot \sigma$, with $z \in \{1.645, 2.326, 3.891\}$ corresponding to 95%, 99%, and 99.99% one-sided confidence levels, respectively.

ID	Dataset Description
(1)	Original MNIST Train Set
(2)	Original MNIST Test Set
(3)	Full MNIST Set = (1) + (2)
(4)	FGSM Successful Adversarial on Train Set
(5)	FGSM Successful Adversarial on Test Set
(6)	FGSM Adversarial Full Set = (4) + (5)
(7)	c&w Successful Adversarial on Train Set
(8)	c&w Successful Adversarial on Test Set
(9)	c&w Adversarial Full Set = (7) + (8)
(10)	(1) + (4)
(11)	(2) + (5)
(12)	(3) + (6)
(13)	(1) + (7)
(14)	(2) + (8)
(15)	(3) + (9)
(16)	(4) + (7)
(17)	(5) + (8)
(18)	(6) + (9)
(19)	(1) + (4) + (7)
(20)	(2) + (5) + (8)
(21)	(3) + (6) + (9)

Table 1: Summary of constructed datasets.

Model	Training Dataset	Fine-tuning Epochs
a	Dataset (1)	-
b	Dataset (10)	5
c	Dataset (10)	10
d	Dataset (13)	5
e	Dataset (13)	10
f	Dataset (19)	5
g	Dataset (19)	10

Table 2: Model training configurations.

To improve computational efficiency and reduce memory usage during activation extraction, we employed a streaming cache mechanism where only the cumulative sum and sum of squares of activations were stored during forward passes. After processing the entire dataset, the mean and variance of each neuron were computed directly from these cached statistics without retaining the full activation history. This approach significantly reduces storage requirements while allowing exact calculation of both mean and standard deviation.

Using these statistics, we evaluated four neuron coverage variants: Using these statistics, we evaluated four variants of neuron coverage:

- **SNAC (baseline):** The number of neurons whose activations exceed their respective maximum activation observed during training (i.e., maximum-based boundary used in prior work).
- **SNAC_95 (ours):** The number of neurons exceeding the 95% confidence boundary based on $\mu + 1.645\sigma$.
- **SNAC_99 (ours):** The number of neurons exceeding the 99% confidence boundary based on $\mu + 2.326\sigma$.
- **SNAC_9999 (ours):** The number of neurons exceeding the 99.99% confidence boundary based on $\mu + 3.891\sigma$.

5.2.4 Comprehensive Coverage and Robustness Evaluation

To evaluate the effectiveness of our proposed SNAC-based metrics, we performed extensive experiments across multiple models and datasets. Specifically, we evaluated all possible combinations of 7 trained models and 21 constructed datasets, resulting in a total of 147 model-dataset pairs.

For each pair, we computed:

- The original **SNAC** metric (using maximum activation boundaries).
- Our proposed confidence interval-based metrics: **SNAC_95**, **SNAC_99**, and **SNAC_9999**.
- Four widely used robustness metrics: **Lipschitz constant**, and **CLEVER scores** under L_1, L_2, and L_∞ norms, denoted as **CL1**, **CL2**, and **CLi**, respectively.

To compute these metrics, we extracted activations from all layers using forward hooks, and applied our SNAC evaluation based on the cached neuron statistics (mean, standard deviation, confidence intervals, and maximums). The Lipschitz constant was estimated by multiplying the spectral norms of all weight matrices layer-wise. For CLEVER score computations, we used the implementation provided by the `art` library, averaging the scores across 10 randomly selected samples for each dataset.

After collecting all metric results across the 147 model-dataset pairs, we analyzed the pairwise correlations between the SNAC metrics and the robustness metrics. The results are summarized as correlation heatmaps, as shown in Figure 2.

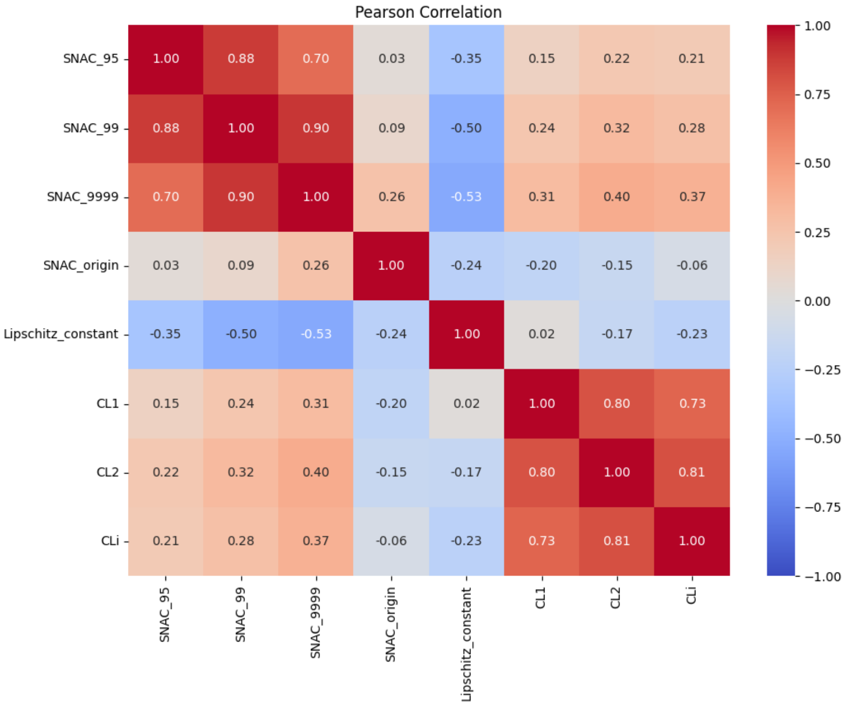


Figure 2: Correlation heatmap between neuron coverage metrics and robustness metrics across all model-dataset combinations. Other correlations are at the Appendix. Note that SNAC$_{origin}$ refers to the original SNAC definition, while SNAC$_{95}$, SNAC$_{99}$, and SNAC$_{9999}$ correspond to variants of the Pigeon method using one-sided confidence bounds at the 95%, 99%, and 99.99% levels, respectively.

6 Results and Analysis

6.0.1 Adversarial Sample Generation via SNAC and FGSM-Based Neuron Boundary Perturbation

Through this experiment, we achieved a boundary-crossing success rate of about 60%. This result demonstrates that even a simple FGSM-based algorithm can easily exploit the vulnerabilities of the existing SNAC metric, revealing its lack of robustness. Therefore, in this study, we propose a new metric that improves upon SNAC to address these limitations.

6.0.2 Adversarial Experiment Results

We analyzed the correlations between the SNAC-based neuron coverage metrics and various robustness metrics across all 147 model-dataset combinations. The pearson correlation results are visualized as a heatmap in Figure 2 and other correlation results are in Appendix A.1

Overall, we observe several noteworthy patterns:

- The original SNAC metric (`SNAC_origin`) shows weak correlation with all robustness metrics, with correlation coefficients ranging from -0.24 to -0.06. This suggests that the maximum-based SNAC boundary may not reliably reflect model robustness.
- In contrast, the proposed confidence interval-based metrics (`SNAC_95`, `SNAC_99`, `SNAC_9999`) exhibit stronger and more consistent correlations with robustness metrics. For instance, `SNAC_9999` achieves up to 0.40 correlation with `CL2` and 0.37 with `CLi`.
- The Lipschitz constant shows negative correlations with all SNAC variants, especially with `SNAC_9999` (-0.53). This is expected since larger Lipschitz constants often indicate less robustness.
- Notably, our confidence-based SNAC metrics exhibit significantly higher correlation with both Lipschitz constants and CLEVER scores compared to the original SNAC, indicating better alignment with robustness characteristics.

These results demonstrate that replacing the maximum activation boundary with statistically-derived confidence thresholds provides a more informative and stable indicator of neural network robustness.

7 Conclusion

In this study, we revisited neuron coverage as a tool for robustness evaluation in neural networks and proposed confidence-threshold-based extensions to the traditional SNAC metric. By analyzing seven retrained LeNet models under various adversarial training conditions and applying our coverage metrics to 147 model-dataset pairs, we demonstrated that higher-threshold SNAC variants (SNAC_95, SNAC_99, SNAC_9999) generally yield stronger correlations with established robustness metrics.

Our findings suggest that replacing the traditional max-based SNAC with our distribution-aware variant, **PIGEON**, can provide more meaningful insights into model robustness. Furthermore, the proposed coverage metrics may serve as principled baselines for defining per-neuron upper activation bounds, which could inform future work in activation-level constraint design or formal verification frameworks.

Overall, confidence-thresholded neuron coverage offers a lightweight and generalizable diagnostic tool for robustness analysis. We hope this work encourages further research on adaptive coverage strategies and principled metric design for safety-critical machine learning systems.

8 Limitations and Future Work

8.1 Deviation from Gaussian Assumption in Neuron Activations

Our coverage metric implicitly assumes that neuron activations approximately follow a Gaussian distribution, which simplifies statistical treatment and coverage computation. However, recent theoretical findings suggest that this assumption becomes increasingly inaccurate as network depth increases. The cumulant expansion framework provides a principled way to quantify and understand such deviations from Gaussianity. Although cumulant-based correction methods could be introduced to mitigate this issue, we have not yet incorporated them into our current metric.

Gaussianity and Cumulants
A random variable X is Gaussian if and only if all cumulants of order $n \geq 3$ vanish:

$$\kappa_n(X) = 0 \quad \text{for all } n \geq 3.$$

Thus, Gaussianity implies that the distribution is fully characterized by its first two cumulants (mean κ_1 and variance κ_2).

Cumulant Expansion Formula
For a transformed random variable $Y = g(X)$ where X has a known distribution, the expectation of $g(X)$ can be approximated via a cumulant expansion:

$$\mathbb{E}[g(X)] = \sum_{n=0}^{\infty} \frac{\kappa_n(X)}{n!} g^{(n)}(\mu),$$

where $g^{(n)}(\mu)$ denotes the n-th derivative of g evaluated at the mean μ of X. This expression shows how higher-order cumulants of X affect the transformed expectation $\mathbb{E}[g(X)]$. In deep networks, $g(\cdot)$ represents nonlinear activation functions (e.g., ReLU), which induce and propagate higher-order cumulants across layers.

Width-Induced Gaussianity

In wide layers, pre-activations $z = \sum_{i=1}^{N} w_i a_i + b$ are formed by summing a large number of weakly correlated activation values. By the Central Limit Theorem and the additivity of cumulants, we have:

$$\kappa_n(z) \sim \mathcal{O}(N^{1-n/2}) \quad \text{for } n \geq 3.$$

Hence, as $N \to \infty$, we obtain $\kappa_n(z) \to 0$ for all $n \geq 3$, and z becomes increasingly Gaussian.

Depth-Induced Non-Gaussianity

In contrast, each nonlinear layer transformation of the form $a^{(l)} = f(W^{(l)}a^{(l-1)} + b^{(l)})$ introduces distortion via $f(\cdot)$, generating new non-zero cumulants. These accumulate with depth, and since the composition of nonlinear functions is not cumulant-preserving, we observe a compounding effect:

$$\kappa_n^{(L)} = \mathcal{F}_n\left(\kappa_3^{(L-1)}, \kappa_4^{(L-1)}, \ldots\right), \quad n \geq 3.$$

Even if $\kappa_n^{(1)} \approx 0$, deeper layers amplify higher-order cumulants significantly, leading to substantial deviation from Gaussianity.

Future Work

As future work, we aim to develop an extended coverage metric that explicitly accounts for the deviation from Gaussianity induced by both network depth and width. By modeling the contribution of higher-order cumulants to activation distributions, such a metric could more accurately reflect the statistical behavior of deep neural networks. This direction would allow us to quantify and correct for the structural sources of distributional error in a principled manner.

8.2 Discussion on Coverage Metric Sensitivity

Structural Limitations of Coverage Metrics

While the proposed SNAC_95, SNAC_99, and SNAC_9999 metrics provide a more refined perspective on model robustness than the traditional maximum-based SNAC, they remain structurally sensitive to the size of the test dataset. Since coverage is defined by whether a neuron's activation exceeds a threshold at least once across all test samples, the probability of such exceedance increases with the number of inputs. This causes coverage scores to rise not necessarily due to changes in model behavior, but as a statistical artifact of larger data volume. As a result, comparing coverage values across datasets of different sizes becomes inherently unfair.

Fixed Threshold Semantics

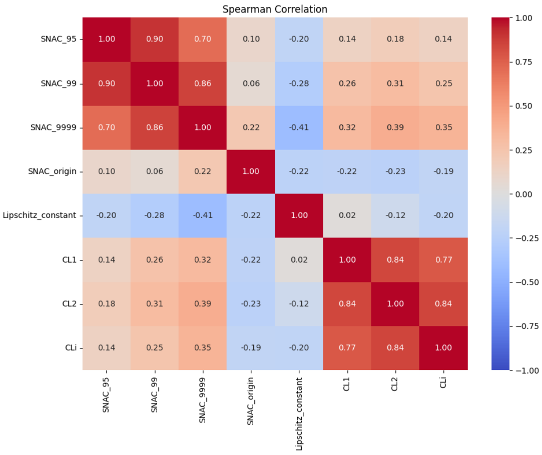
The use of fixed statistical thresholds (e.g., $z = 1.645,\ 2.326,\ 3.891$) assumes that these values have consistent semantic meaning across different models and datasets. However, variations in data distribution and internal activation dynamics mean that the same z-value can correspond to different practical levels of rarity. This limits the generalizability of fixed-threshold-based coverage metrics, especially when comparing across architectures or data domains.

Concluding Note

Although higher thresholds (e.g., SNAC_9999) showed stronger correlation with robustness metrics in our experiments, their semantic meaning depends heavily on data scale and rarity assumptions. We encourage future experiments to treat these thresholds as tunable hyperparameters, and to consider dataset-specific calibration strategies where appropriate.

A Appendix

A.1 Other Correlation Heatmaps Between Neuron Coverage Metrics and Robustness Metric



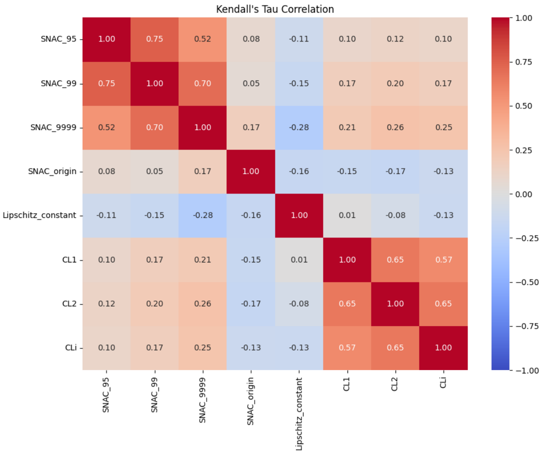


Figure 3: (a) Spearman Correlation Heatmap

Figure 4: (b) Kendall Correlation Heatmap

A.2 Details on Adversarial Settings

Table 3: Attack parameters and success rates reported by Dong et al. [2] on MNIST.

Dataset	Attack Method	Model	Parameter	Success Rate
MNIST	FGSM	LeNet	0.2, 0.3, 0.4	0.99
	C&W	LeNet	9, 10, 11	0.99

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[2] Yizhen Dong, Peixin Zhang, Jingyi Wang, Shuang Liu, Jun Sun, Jianye Hao, Xinyu Wang, Li Wang, Jinsong Dong, and Ting Dai. An empirical study on correlation between coverage and robustness for deep neural networks. In *2020 25th International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 73–82, 2020.

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

[4] Fabrice Harel-Canada, Lingxiao Wang, Muhammad Ali Gulzar, Quanquan Gu, and Miryung Kim. Is neuron coverage a meaningful measure for testing deep neural networks? In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 851–862, 2020.

[5] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, et al. Deepgauge: Multi-granularity testing criteria for deep learning systems. In *Proceedings of the 33rd ACM/IEEE international conference on automated software engineering*, pages 120–131, 2018.

[6] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. In *proceedings of the 26th Symposium on Operating Systems Principles*, pages 1–18, 2017.

[7] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018.

[8] Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86:391–423, 2012.

[9] Zhou Yang, Jieke Shi, Muhammad Hilmi Asyrofi, and David Lo. Revisiting neuron coverage metrics and quality of deep neural networks. In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 408–419. IEEE, 2022.