DEEPGRE: GLOBAL ROBUSTNESS EVALUATION OF DEEP NEURAL NETWORKS

Tianle Zhang[1], *Jiaxu Liu*[1], *Yanghao Zhang*[1], *Ronghui Mu*[1] *and Wenjie Ruan*[1]

[1]University of Liverpool, UK

ABSTRACT

Robustness measurements on deep neural networks (DNNs) have gained significant attention, especially in safety-critical applications. Numerous studies have been devoted to assessing the robustness of classifiers by averaging local robustness over a fixed set of data samples, such as test set. However, the local statistics may not provide an accurate representation of the actual global robustness over the entire underlying unknown data distribution. To address this challenge, this paper proposes a new framework, namely DeepGRE, for global robustness estimates of adversarial perturbation in combination with generative models and existing local robustness evaluation methods. Besides, DeepGRE employs Quasi-Monte Carlo approach to produce estimates of global robustness with low variance, making the assessments more reliable and statistically sound, since randomness is introduced by all samples drawn from a generative model. From a theoretical perspective, this work naturally provides an upper bound between true global robustness and estimated global robustness based on Lipschitz continuity, and also derives a statistical guarantee on the difference between true and empirical estimates with respect to sample complexity.

***Index Terms*—** Global Robustness, Deep Neural Network, Generative Models

1. INTRODUCTION

As Deep Neural Networks (DNNs) make remarkable success, the demand for reliable neural network components, especially in safety-critical scenarios, has become increasingly urgent. However, due to the lack of symbolic models and formal specifications, achieving safety certification for DNNs poses challenges. A key specification for DNNs is their robustness against input perturbations [1, 2, 3, 4].

Recently, remarkable efforts have been devoted to empirically evaluating the robustness of DNNs through adversarial examples, such as FGSM [1], PGD [5, 6] and C&W [7]. These approaches aim to design the most sophisticated and formidable attacks possible, and then measure empirical robustness based on the adversarial performance. For instance, Auto-Attack [8], a state-of-the-art attack based on an ensemble of advanced white-box and black-box adversarial perturbation methods, has become a benchmark for empirical robustness. However, these approaches can only falsify robustness claims and do not provide any theoretical guarantee on their results. Concurrently, there is a push to develop a certified score for adversarial robustness, signifying a quantifiable level of attack-proof certification. Some research, from the verification community, has focused on certified robustness performance against arbitrary attackers with rigorous guarantees, as opposed to empirical adversarial accuracy [9, 10, 11]. Furthermore, certain inherent statistics of models, such as the global Lipschitz constant [12] and the CLEVER score [13, 14], have proven to be effective metrics for assessing the level of robustness by disentangling the correlation between these measures and robustness.

Despite numerous methods for evaluating adversarial robustness, most of them focus on *local* or *point-wise* safety risks for single, specific inputs [15, 16, 1, 9]. The problem of *global robustness* remains a largely unexplored challenge. Ideally, global robustness is defined as the expectation of point-wise robustness across the whole data distribution [17], implying transparency of the data distribution and the ability to draw an unlimited number of samples from the true distribution for reliable robustness evaluation. However, in reality, the data distribution is unknown and difficult to characterize. As a result, current works simply define *global* robustness as aggregated local, pointwise robustness problems over *finite sample points* (in the *test dataset*) [9, 10]. The sampling process for these test data could be biased and not representative of the actual global robustness of the underlying data distribution, leading to the risk of incorrect or skewed robustness benchmarks.

In this paper, we propose a novel method named DeepGRE (**G**lobal **R**obustness **E**valuation for **Deep** Neural Networks) with *provable guarantees*. Its contributions lie in three aspects. Firstly, the *global robustness* is defined as the expected point-wise robustness over data distribution. Given unknown actual data distribution, DeepGRE employs a *proxy* generative model to capture overall robustness *w.r.t.* the entire data space. Secondly, we utilize a low-discrepancy sequence to produce more reliable global robustness statistics, which exhibit provable reduced variance when compared with traditional sampling techniques. Furthermore, we provide a theoretical upper bound on its estimation of the true global robustness using generative models, accompanied by statistical guarantees on Monte Carlo estimates.

2. GLOBAL ROBUSTNESS VIA GENERATIVE MODELS

2.1. Problem Formulation

Consider a classifier hypothesis f_θ with its parameter $\theta \in \Theta$, modeled as a differentiable mapping $\mathcal{X} \to \Delta^m$, where Δ^m denotes the probability simplex in $\mathbb{R}^m$. For brevity, we will omit θ when there is no ambiguity. Let $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ represent an input and its corresponding label, respectively. For a specific input $x \in \mathbb{R}^n$, $f(x) = \{c_1, c_2, \ldots, c_m\} \in \mathbb{R}^m$ represents the confidence values for m classification labels.

Let $o : \mathcal{X} \to \mathbb{R}$ be a Lipschitz continuous function statistically evaluating the network and comparing the robustness of homogeneous networks. A robustness score is the output of o providing rather vague estimates of robustness or formal guarantees of deterministic or probabilistic nature. In any case o assesses robustness only on single inputs $x \in \mathcal{X}$, and a higher robustness score $o(h, x)$ means better local robustness of a classifier h on x.

Definition 1 (Local Robustness)**.** Given two homogeneous networks h_1 and h_2, we assert that h_1 is strictly more robust than h_2 with respect to an evaluation function o and a given input x, if $o(h_1, x) > o(h_2, x)$.

Definition 2 (Global Robustness)**.** The global robustness of a hypothesis h is quantified as follows,

$$\mathcal{O}(h) = \mathbb{E}_{x \sim P_d}[o(h, x)] = \int_{x \sim P_d} o(h, x) p(x) dx. \qquad (1)$$

where p is the density probability governing the real data distribution P_d.

2.2. GRE via Generative Models

*All proofs can be found in the arxiv version.

Recall that a generative model that employs a randomly sampled vector $z \in \mathbb{R}^d$ drawn from the *latent code distribution* $\mathcal{Z}$ to yield a data sample $G(z)$. If $z \sim \mathcal{Z}$,We can infer that for a proficiently trained generative model, $G(z) \sim \mathcal{X}$. Subsequently, we introduce our principal theorem that estimates the global robustness of a classifier h, gauged by the data distribution given by $G(\cdot)$, and provides its theoretical analysis.

Definition 3 (Estimated Global Robustness)**.** Given a generator G capable of generating a sample $G(z)$ with $z \sim \mathcal{Z}$, the estimated global robustness of h is formally defined as

$$\widehat{\mathcal{O}}(h) = \mathbb{E}_{x \sim \widehat{P_d}}[o(h, x)] = \mathbb{E}_{z \sim \widehat{P_z}}[w(h, z)]. \qquad (2)$$

wherein $w = o \cdot G$ represents the concatenated function.

Assuming that dataset S consists of m independent and identically distributed (*i.i.d.*) samples from real distribution P_d, which is defined over a compact set $\mathcal{X} \subset \mathbb{R}^n$, and m i.i.d. samples drawn from the noise distribution P_z, which is defined over a compact set $\mathcal{Z} \subset \mathbb{R}^d$. We denote $\widehat{P}_d$ and $\widehat{P}_z$ are the empirical distributions derived from set S, respectively.

Assumption 1. *The function o is K_o-Lipschitz continuous with respect to its input on a compact domain and is upper bounded by a constant $C \geq 0$.*

Assumption 2. *Each generator $G \in \mathcal{G}$ is K_g-Lipschitz continuous with respect to its input z over a compact set $\mathcal{Z} \subset \mathbb{R}^d$ with diameter at most B_z.*

These assumptions are reasonable and satisfied by various generative models. Furthermore, we have $K = K_o K_g$, thereby upper bounding the Lipschitz constant of the concatenated evaluation function $w = o \cdot G$. Firstly, since $\widehat{P_z}$ is an empirical version of P_z, Thm. 1, the generalization bound for $\sup_{G \in \mathcal{G}} \left| \mathbb{E}_{z \sim P_z} w(h, z) - \mathbb{E}_{z \sim \hat{P}_z} w(h, z) \right|$ can be derived.

Theorem 1. *Given the Asm. 1 and 2, m i.i.d. samples are from latent code distribution p_z defined over a compact set $\mathcal{Z} \subset \mathbb{R}^d$, and $B_z = \sup_{z, z' \in \mathcal{Z}} \|z - z'\|_\infty$. For any $\delta \in (0, 1]$, $\lambda \in (0, B_z]$, with probability of at least $1 - \delta$, we have*

$$\sup_{G \in \mathcal{G}} \left| \mathcal{O}(h) - \widehat{\mathcal{O}}(h) \right| \leq K\lambda + \frac{C}{\sqrt{m}} \sqrt{\lceil B_z^n \lambda^{-n} \rceil \log 4 - 2 \log \delta}.$$

Thm. 1 ensure that the upper bounds on $\left| \mathcal{O}(h) - \widehat{\mathcal{O}}(h) \right|$ hold true for particular G in the family of generators.

2.3. Quasi-Monte Carlo Evaluation

As defined in Def. 3, the true global robustness is estimated with the expectation of concatenated function $w(\cdot)$, integrating a local robustness statistical function $o(\cdot)$ and a well-trained generator $G(\cdot)$. However, the estimate of true global robustness introduces randomness when diverse samples obtained from the generator $G(z)$ are employed for evaluation, referring to Thm. 2.

Theorem 2. *Given a local robustness function o of interest, and global robustness represented as an integral of a concatenated function $w = o \cdot G$, we have $\mathcal{O}(h) = \int w(z) p(z) dz$. Using N i.i.d. samples, a Monte Carlo estimate of the integral yields $\hat{\mathcal{O}} = \mathcal{O} + \xi$ with $\mathbb{E}[\xi] = 0$ and $var(\xi) = \frac{C(w)}{N}$, where $C(w)$ is $\int (w - \mathcal{O}(h))^2 p(z) dz$.*

That means we can identify an integrator that produces estimates of global robustness with low variance. A connection between better discrepancy and more accurate integration, *i.e.,* global robustness evaluation, can be directly derived from the Koksma-Hlawka inequality [18], *i.e.,*

$$\left| \mathcal{O}(h) - \hat{\mathcal{O}} \right| \leq V[w] D_N^* = V[o \cdot G] D_N^*. \qquad (3)$$

Here, $V[o \cdot G]$ depends on the concatenated function needed to be integrated and can be challenging to determine, while D_N^* denotes the discrepancy of the sampling sequence. For Sobol sequences, the discrepancy is $O\left((\log N)^d N^{-1}\right)$, where d is the number of dimensions. That means that when generating a global robustness estimate using a generator fed with low-discrepancy sequences like Sobol sequences, the error tends

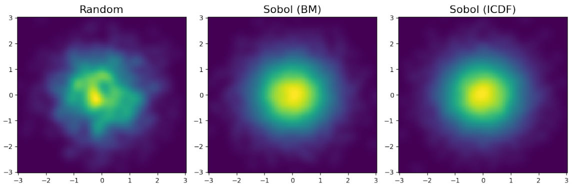


Fig. 1. 2D density plots of random points vs Sobol points.

to be lower compared to random sequences for the same number of points. Moreover, Quasi-Monte Carlo method with low discrepancy sequences such as Sobol and Halton sequences has been proven to achieve convergence up to 5 times faster than traditional Monte Carlo methods with lower error rates.

As a consequence of these insights, we leverage the Quasi-Monte Carlo (QMC) method with a low-discrepancy sequence, specifically the Sobol sequence, to compute the estimated global robustness associated with a given generator rather than classic Monte Carlo. It is able to obtain more accurate estimates. However, it is important to note that sequences like Sobol are deterministic in nature. To reintroduce randomness into the QMC process, one way is to scramble the base digits of the sequence [19]. The resulting sequence will still have a QMC structure and the expectation of the integral remains unchanged even after introducing this randomness.

2.4. Sample Distribution for GM

Recall that a generative model takes a random vector sampled from a zero-mean normal Gaussian distribution as input to generate a data sample $x = G(z)$, while low-discrepancy sequences are commonly designed to produce points within the unit hypercube. To ensure working a direct drop-in replacement for current generators using $\mathcal{N}(0, 1)$ as the prior for z, intuitively there are two ways to transform a uniform distribution to a standard normal distribution while preserving the low-discrepancy property of the generated points after the transformation.

The *Inverse Cumulative Distribution Function (ICDF)* provides the value of a random variable such that the probability of it being less than or equal to that value is equal to the given probability. Specifically, $z' = \sqrt{2}\,\mathrm{erf}^{-1}(2z-1), \quad z \sim \mathcal{U}(0, 1)$, where $Q(z)$ is the ICDF and erf is the Gauss error function. In our context, since our low-discrepancy sequence generates $\mathcal{U}[0, 1]$, we can treat them as probabilities and use $Q(z)$ to transform them into $\mathcal{N}(0, 1)$.

For the *Box-Muller transformation*, given $z \in (0, 1)^d$ where d is an even number, let z^{even} be the even-numbered components of z and z^{odd} be the odd-numbered components of u. By using this transformation, we can obtain the desired normal distribution, *i.e.*, $z_0 = \sqrt{-2\ln\left(z^{\text{even}}\right)}\cos\left(2\pi z^{\text{odd}}\right)$, $z_1 = \sqrt{-2\ln\left(z^{\text{even}}\right)}\sin\left(2\pi z^{\text{odd}}\right)$, and then $z' = (z_0, z_1)$.

We show the comparison of the sampled Gaussian distributions presented in Fig. 1. The figure demonstrates that Sobol sequences exhibit superior quality compared with classic Monte Carlo sampling. This improved quality is visually evident in the higher uniformity and dispersion of Sobol points across the density plots.

2.5. Anytime Statistical Guarantees on MC Estimates

Our aim is to estimate the global robustness $\mathcal{O}(h)$ of models h through expectations $\widehat{\mathcal{O}}(h)$ of statistical evaluation functions o under some distribution p (either equal or close to the data distribution). In our case, the estimated global robustness is evaluated using the Quasi-Monte Carlo estimate, which involves computing the mean of generated samples $S = \{G(z_i)\}_{i=1}^{n}$ from a given generator G and its sampling distribution. This estimate can be expressed as

$$\widehat{\mathcal{O}}_S(h) = \frac{1}{n}\sum_{i=1}^{n} o\left(h, G\left(z_i\right)\right) = \frac{1}{n}\sum_{i=1}^{n} w\left(h, z_i\right). \quad (4)$$

In what follows, we present another main theorem to deliver a run-time probabilistic guarantee on the difference between the Monte Carlo estimate $\widehat{\mathcal{O}}_S(h)$ and the true expectation $\widehat{\mathcal{O}}(h)$ in terms of sample numbers. Since the generator can continually and iteratively draw samples, this theorem provides pragmatic means to deliver statistical guarantees at *anytime*. Here the number of samples n is treated as a stopping time J that, being a random variable, depends on the process rather than being chosen in advance, or independent of the underlying process.

Theorem 3 (Anytime statistical guarantee on estimates)**.** *Let h be a hypothesis, and o denote statistically robustness evaluation function upper bounded by constant C (Asm. 1). Given a generator G with random latent code z, let $z_1, z_2, \ldots$ be i.i.d samples of z. Consider a stopping time t as a random variable on $\mathbb{N} \cup \{\infty\}$ such that $P\left[t < \infty\right] = 1$. Then, for any stopping time t and a given $\delta \in \mathbb{R}_+$,*

$$\mathrm{P}\left[\left|\widehat{\mathcal{O}}_S(h) - \widehat{\mathcal{O}}(h)\right| \le \varepsilon(\delta, t)\right] \ge 1 - \delta \quad (5)$$

holds, where $\varepsilon(\delta, t) = C \cdot \sqrt{\frac{0.6\cdot\log(\log_{1.1} t+1)+1.8^{-1}\cdot\log(24/\delta)}{t}}$. This means that the anytime evaluation $\widehat{\mathcal{O}}_S(h)$ of global robustness at time t is $\varepsilon(\delta, t)$-close to the true value $\widehat{\mathcal{O}}(h)$ with probability at least $1 - \delta$.

3. EXPERIMENTS

In order to evaluate the proposed method, an assessment is conducted involving various trained neural networks on public data sets CIFAR-10 and ImageNet, which we describe in more detail in the Appendix.

3.1. Experiment Setup

Generative Models. Our experiments focus mainly on Generative Adversarial Networks [20] as they are one of the most popular deep generative models today. We ran our evaluations on SAGAN [21], SNGAN [22], and BigGAN [23].

DeepGRE Implementation. The proposed framework can integrate with any local robustness evaluation methods, as GeepGRE is an estimate of integrating local robustness over a sampling distribution. In our experiments, we leverage the CLEVER score, a local robustness metric based on extreme value theory, to compute global robustness statistics, which are then scaled between [0,100].

Compute Resources. All the experiments are executed on a system equipped with a 32-Core AMD EPYC 7452 CPU and an NVIDIA A100 40GB GPU.

3.2. Evaluation with Different Sampling Schemes

Recall Sec. 2.4 introduces that an integrator with lower discrepancy sampling sequences will be able to produce a lower variance estimate of global robustness. Tab. 3.2 compares 20 runs each of global robustness statistics for a variety of models on CIFAR-10, estimated using 10,000 i.i.d. normal samples or a Sobol sequence with either Box-Muller transform or ICDF. It is clear from the table that robustness evaluation should always use a low-discrepancy sequence, because evaluating with Sobol$_{BM}$ and Sobol$_{ICDF}$ gives better and reliable evaluation results with lower standard deviations.

		Cui_WRN-28-10	Xu_WRN-28-10	Wang_WRN-28-10	Sehwag_R18
SAGAN	Normal	59.94$_{\pm 0.6539}$	58.35$_{\pm 0.5998}$	59.42$_{\pm 0.5171}$	55.23$_{\pm 0.5347}$
	Sobol$_{BM}$	**60.09**$_{\pm 0.6387}$	58.43$_{\pm 0.4052}$	59.28$_{\pm \mathbf{0.3837}}$	55.27$_{\pm \mathbf{0.3613}}$
	Sobol$_{ICDF}$	59.95$_{\pm \mathbf{0.4966}}$	**58.50**$_{\pm \mathbf{0.3622}}$	**59.45**$_{\pm 0.4681}$	**55.33**$_{\pm 0.3646}$
SNGAN	Normal	61.50$_{\pm 0.9732}$	**64.59**$_{\pm 2.2887}$	60.93$_{\pm 0.9670}$	**56.55**$_{\pm 0.4001}$
	Sobol$_{BM}$	**64.19**$_{\pm 0.8933}$	63.62$_{\pm 0.7236}$	63.16$_{\pm 0.7811}$	56.16$_{\pm 0.1911}$
	Sobol$_{ICDF}$	63.31$_{\pm \mathbf{0.8300}}$	63.41$_{\pm \mathbf{0.4345}}$	**64.15**$_{\pm \mathbf{0.5979}}$	56.13$_{\pm \mathbf{0.1860}}$
BIGGAN	Normal	61.56$_{\pm 0.9239}$	61.58$_{\pm 1.4360}$	59.89$_{\pm 0.6915}$	55.92$_{\pm 0.4336}$
	Sobol$_{BM}$	**63.64**$_{\pm 0.7911}$	62.02$_{\pm 0.6376}$	62.75$_{\pm \mathbf{0.3165}}$	56.56$_{\pm \mathbf{0.1809}}$
	Sobol$_{ICDF}$	62.49$_{\pm \mathbf{0.6778}}$	**62.75**$_{\pm \mathbf{0.3892}}$	**63.31**$_{\pm 0.4942}$	**56.67**$_{\pm 0.2589}$

Table 1. DeepGRE statistics of different models evaluated on Normal and Sobol sequences over 20 runs. Bolded values indicate the best score or standard deviation.

3.3. Quantitative Results of Experiments

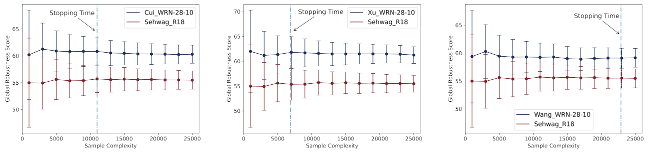
We assess the effectiveness of our method in two aspects: robustness comparison of networks and its associated sample complexity.

To begin, we compare the networks' robustness on ImageNet dataset using DeepGRE global robustness statistics, RobustBench Accuracy (RB Acc.), and AutoAttack Accuracy (AA Acc.). Specifically, DeepGRE statistics are computed with 500 generated samples. Auto-Attack accuracy is obtained with Auto-Attack on 500 generated samples as well. RobustBench accuracy is the adversarial accuracy evaluated with AutoAttack on the test set. Notably, SAGAN is used to generate images in conjunction with Sobol (ICDF) sequence.

The comparative analysis of robustness across six different networks, based on the aforementioned robustness metrics, is demonstrated in Tab. 3.3. It's evident that the DeepGRE statistics align coherently with the results from attack-dependent robustness evaluations.

Model	RB Acc. (%)	AA Acc. (%)	DeepGRE
LiuConvNeXtL	59.56	54.10	24.80
Singh_L-ConvStem	57.70	53.30	20.79
Liu_SwinB	56.16	52.20	21.35
PengRobust	48.94	43.60	15.58
SalmanDo_50_2	38.14	31.80	11.34
WongFast	26.24	18.60	9.87

Table 2. Robustness evaluation on ImageNet *w.r.t.* DeepGRE, RobustBench, and Auto-Attack.



(a) Sehwag_R18 *v.s.* Cui_WRN-28-10 (b) Sehwag_R18 *v.s.* Xu_WRN-28-10 (c) Sehwag_R18 *v.s.* Wang_WRN-28-10

Fig. 2. The relationship between the statistical bounds of DeepGRE with confidence level (99.99%) and sample complexity

Additionally, we investigate the sample complexity associated with the task of comparing network robustness. Thm. 3 provides an efficient and adaptive way of assessing robustness in terms of the sample number needed for statistical guarantees. It enables our models' evaluation to iteratively draw samples. Upon termination, the sample number becomes a stopping time, depending on the ongoing process. Specifically, given a confidence level δ, the sampling process is halted if one network's global robustness lower bound is surpasses the upper bound of the other network.

We present a comparison involving three pairs of networks on CIFAR-10 dataset with a varying number of generated data samples (using SAGAN and Sobol sequence with the Box-Muller transformation). According to Fig. 2, we have 90% confidence that the Sehwag_R18 is less robust than the others, *i.e.*, Cui_WRN-28-10 / Xu_WRN-28-10 / Wang_WRN-28-10), using around 11,000, 7,000 and 23,000 samples, respectively. The results underscore the feasibility and efficiency of our approach with the anytime global robustness evaluation in Thm. 3.

4. CONCLUSION

In this study, we proposed DeepGRE, a novel framework for global robustness evaluation. This framework seamlessly combines current local robustness methods with cutting-edge generative models, and employs a low-discrepancy sequence for reduced variance robustness statistics. A limitation, however, is that inherent biases from generative models can influence these statistics relative to the underlying data distribution. Future endeavors will focus on attaining unbiased global robustness, exploring bias correction via importance sampling and refining training methodologies accordingly.

5. REFERENCES

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[2] Battista Biggio, Giorgio Fumera, and Fabio Roli, "Security evaluation of pattern classifiers under attack," *IEEE Transactions on Knowledge and Data Engineering*, pp. 984–996, 2017.

[3] Tianle Zhang, Wenjie Ruan, and Jonathan E. Fieldsend, "Proa: A probabilistic robustness assessment against functional perturbations," in *ECML-PKDD*, 2022, pp. 154–170.

[4] Peipei Xu, Wenjie Ruan, and Xiaowei Huang, "Quantifying safety risks of deep neural networks," *Complex & Intelligent Systems*, pp. 1–18, 2022.

[5] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio, "Adversarial machine learning at scale," in *ICLR*, 2017.

[6] Yanghao Zhang, Fu Wang, and Wenjie Ruan, "Fooling object detectors: Adversarial attacks by half-neighbor masks," *CoRR*, 2021.

[7] Nicholas Carlini and David A. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE S&P*. 2017, pp. 39–57, IEEE Computer Society.

[8] Francesco Croce and Matthias Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *ICML*, 2020, pp. 2206–2216.

[9] Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska, "Reachability analysis of deep neural networks with provable guarantees," in *IJCAI*, 2018, pp. 2651–2659.

[10] Wenjie Ruan, Min Wu, Youcheng Sun, Xiaowei Huang, Daniel Kroening, and Marta Kwiatkowska, "Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance," in *IJCAI*, 2019, pp. 5944–5952.

[11] Peipei Xu, Fu Wang, Wenjie Ruan, Chi Zhang, and Xiaowei Huang, "Sora: Scalable black-box reachability analyser on neural networks," in *ICASSP*, 2023, pp. 1–5.

[12] Klas Leino, Zifan Wang, and Matt Fredrikson, "Globally-robust neural networks," in *ICML*, Marina Meila and Tong Zhang, Eds., 2021, pp. 6212–6222.

[13] 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," in *ICLR*, may 2018.

[14] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Aurelie Lozano, Cho-Jui Hsieh, and Luca Daniel, "On extensions of clever: A neural network robustness evaluation algorithm," in *GlobalSIP*, nov 2018.

[15] Souradeep Dutta, Susmit Jha, Sriram Sankaranarayanan, and Ashish Tiwari, "Output range analysis for deep neural networks," *CoRR*, vol. abs/1709.09130, 2017.

[16] Jörn-Henrik Jacobsen, Jens Behrmann, Nicholas Carlini, Florian Tramèr, and Nicolas Papernot, "Exploiting excessive invariance caused by norm-bounded adversarial robustness," *CoRR*, vol. abs/1903.10484, 2019.

[17] Zaitang Li, Pin-Yu Chen, and Tsung-Yi Ho, "GREAT score: Global robustness evaluation of adversarial perturbation using generative models," *CoRR*, vol. abs/2304.09875, 2023.

[18] Art B Owen, "Quasi-monte carlo sampling," *Monte Carlo Ray Tracing: Siggraph*, vol. 1, pp. 69–88, 2003.

[19] Art B Owen, "Randomly permuted (t, m, s)-nets and (t, s)-sequences," in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. Springer, 1995, pp. 299–317.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[21] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena, "Self-attention generative adversarial networks," in *ICML*, 2019, pp. 7354–7363.

[22] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida, "Spectral normalization for generative adversarial networks," in *ICLR*, 2018.

[23] Andrew Brock, Jeff Donahue, and Karen Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *ICLR*, 2019.

[24] Huan Xu and Shie Mannor, "Robustness and generalization," *Mach. Learn.*, vol. 86, no. 3, pp. 391–423, 2012.

[25] Ari Heljakka, Arno Solin, and Juho Kannala, "Towards photographic image manipulation with balanced growing of generative autoencoders," in *IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 3109–3118.

[26] Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun, "Adversarially regularized autoencoders," in *ICML*, 2018, pp. 5897–5906.

6. APPENDIX

6.1. Learning Algorithm

Consider a learning problem specified by a function/hypothesis class $\mathcal{H}$, an instance set $\mathcal{X}$ with diameter at most B, and a evaluation function $o : \mathcal{H} \times \mathcal{X} \rightarrow \mathbb{R}$ which is bounded by a constant C. Given a distribution P_x defined on $\mathcal{X}$, the quality of a function $h(x)$ is measured by its expected evaluation statistic $F(P_x, h) = \mathbb{E}_{x \sim P_x}[o(h, x)]$. Since P_x is unknown, we need to rely on a finite training sample $S = \{x_1, \ldots, x_m\} \subset \mathcal{X}$ and often work with the empirical evaluation $F\left(\widehat{P}_x, h\right) = \mathbb{E}_{x \sim \widehat{P}_x}[o(h, x)] = \frac{1}{m}\sum_{x \in S} o(h, x)$, where $\widehat{P}_x$ is the empirical distribution defined on S. A learning algorithm $\mathcal{A}$ will pick a function $h_m \in \mathcal{H}$ based on input S, i.e., $h_m = \mathcal{A}(\mathcal{H}, S)$. Let $\mathcal{X} = \bigcup_{i=1}^{N} \mathcal{X}_i$ be a partition of $\mathcal{X}$ into N disjoint subsets. We use the following definition about robustness of an algorithm.

Definition 4 (Robustness)**.** An algorithm $\mathcal{A}$ is (N, ϵ)-robust, for $\epsilon : \mathcal{X}^m \rightarrow \mathbb{R}$, if the following holds for all $S \in \mathcal{X}^m$: $\forall s \in S, \forall x \in \mathcal{X}, \forall i \in 1, \ldots, N$, if $s, x \in \mathcal{X}_i$ then $|o(\mathcal{A}(\mathcal{H}, S), s) - o(\mathcal{A}(\mathcal{H}, S), x)| \leq \epsilon(S)$.

Basically, a robust algorithm will be a hypothesis which ensures that the evaluation difference of two similar data instances should be the same. A small change in the input leads to a small change in the output of the given hypothesis. In other words, the robustness ensures that each testing sample which is close to the training dataset will have a similar evaluation with that of the closest training samples. Therefore, the hypothesis $\mathcal{A}(S)$ will generalise well over the areas around S.

Theorem 4 ([24])**.** *If a learning algorithm $\mathcal{A}$ is (N, ϵ)-robust and the training data S is an i.i.d. sample from distribution P_x, then for any $\delta \in (0, 1]$ we have the following with probability at least $1-\delta$: $\left|F\left(P_x, \mathcal{A}(\mathcal{H}, S)\right) - F\left(\widehat{P}_x, \mathcal{A}(\mathcal{H}, S)\right)\right| \leq \epsilon(S) + C\sqrt{(N \log 4 - 2\log\delta)/m}$.*

This theorem formally makes the important connection between robustness of an algorithm and generalisation. If an algorithm is robust, then its resulting hypotheses can generalise. One important implication of this result is that we should ensure the robustness of a learning algorithm. However, it is nontrivial to do so in general.

Let us have a closer look at robustness. $\epsilon(S)$ in fact bounds the amount of change in the loss with respect to a change in the input given a fixed hypothesis. This observation suggests that robustness closely resembles the concept of Lipschitz continuity.

Definition 5 (Lipschitz Continuity)**.** Given two metric spaces (X, dX) and (Y, dY), where dX and dY are the metrics on the sets X and Y respectively, a function $f : X \rightarrow Y$ is called Lipschitz continuous if there exists a real constant $K \geq 0$ such that, for all $x_1, x_2 \in X$:

$$d_Y\left(f\left(x_1\right), f\left(x_2\right)\right) \leq K d_X\left(x_1, x_2\right) \tag{6}$$

K is called the Lipschitz constant for the function f.

Therefore, we establish the following connection between robustness and Lipschitz continuity.

Lemma 5. *Given any constant $\lambda > 0$, consider a function $f : \mathcal{H} \times \mathcal{X} \rightarrow \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^n$ is compact, $B = \operatorname{diam}(\mathcal{X}) = \max_{x, x' \in \mathcal{X}} \|x - x'\|_\infty$, $N = \lceil B^{n_x}\lambda^{-n_x}\rceil$. If for any $h \in \mathcal{H}$, $o(h, x)$ is K-Lipschitz continuous w.r.t input x, then any algorithm $\mathcal{A}$ that maps $\mathcal{X}^m$ to $\mathcal{H}$ is $(N, L\lambda)$-robust.*

Combining Theorem 4 and Lemma 5, we make the following connection between Lipschitz continuity and generalization.

Theorem 6. *If a loss $o(h, x)$ is K-Lipschitz continuous w.r.t input x in a compact set $\mathcal{X} \subset \mathbb{R}^n$, for any $h \in \mathcal{H}$, and $\widehat{P}_x$ is the empirical distribution defined from m i.i.d. samples from distribution P_x, then $\sup_{h \in \mathcal{H}}\left|F\left(P_x, h\right) - F\left(\widehat{P}_x, h\right)\right|$ is upper-bounded by*

1. *$L\lambda + C\sqrt{\left(\lceil B^{n_x}\lambda^{-n_x}\rceil \log 4 - 2\log\delta\right)/m}$ with probability at least $1 - \delta$, for any constants $\delta \in (0, 1]$ and $\lambda \in (0, B]$.*
2. *$(LB + 2C)m^{-\alpha/n}$ with probability at least $1 - 2\exp\left(-0.5m^\alpha\right)$, for any $\alpha \leq n/(2 + n)$.*

6.2. Generative Models

From a statistical standpoint, let x denote the observable variable and let y represent the corresponding label. The learning objective for a generative model is to model the conditional probability distribution $P(x|y)$.

Among all generative models, generative adversarial networks (GANs) have garnered considerable attention in recent years due to their ability to generate realistic and indistinguishable high-quality images. Fundamentally, GANs consist of two neural networks—the discriminator and the generator G—and are trained to enable G to generate elements that mimic a target true data distribution P_d, even in the simplest case without sample labels. Given a training dataset of real-world data samples, the generator aims to capture the true data distribution, while the discriminator strives to discern whether the data samples originate from the generator or real data. The objective of the generator is to minimize the divergence between the discriminator's outputs for true versus generated samples, while the objective of the discriminator is to accurately classify the true versus fake samples. With certain enhancements, GANs may also be capable of reconstruction—finding a latent representation z for a given original vector $x \in \mathcal{X}$ such that $G(z)$ closely approximates x (e.g., according to some norm in the original space).

Furthermore, an autoencoder ($\mathcal{N}^E, \mathcal{N}^D$), where $\mathcal{N}^E$ and $\mathcal{N}^D$ are feed-forward neural networks denoted as the encoder and the decoder respectively, is another notable generative model. Its goal is to compress (encode) its inputs $x \in \mathcal{X}$ into low-dimensional latent vectors z, such that approximate decompression (decoding, reconstruction) can be achieved: $\mathcal{N}^D(z)$ closely approximates x. A generative autoencoder, such as those in [25, 26], is an autoencoder whose decoder is additionally trained to sample from the original distribution —effectively, a generative autoencoder performs both the tasks of an autoencoder and a GAN.

In addition to GANs and autoencoders, diffusion models (DMs) are also gaining traction. DMs consist of two stages: the forward diffusion process and the reverse diffusion process. In the forward process, the input data is gradually perturbed by Gaussian noises until it eventually resembles an isotropic Gaussian distribution. In the reverse process, DMs reverse the forward process, implementing a sampling process from Gaussian noises to reconstruct the true samples.

In summary, generative models are capable of data generation from low-dimensional vectors. Data generation is achieved by applying G to a low-dimensional vector $z \in \mathbb{R}^d$ sampled from the latent code distribution $\mathcal{Z}$ (typically, $\mathcal{N}(0, I)$). If $z \sim \mathcal{Z}$, then for a well-trained generator G, we may assume that $G(z) \sim \mathcal{X}$. Often, the dimension of $\mathcal{Z}$ is made smaller than the dimension of $\mathcal{X}$.