6-30-2021

Exploring Robustness of Neural Networks through Graph Measures

Asim Waqas

Ghulam Rasool
Rowan University, rasool@rowan.edu

Hamza Farooq

Nidhal Carla Bouaynaya
Rowan University, bouaynaya@rowan.edu

Exploring Robustness of Neural Networks through Graph Measures

Asim Waqas*
Department of Electrical and Computer Engineering
Rowan University
waqasa8@students.rowan.edu

Ghulam Rasool
Department of Electrical and Computer Engineering
Rowan University
rasool@rowan.edu

Hamza Farooq
Department of Neurology
University of Minnesota

Nidhal C. Bouaynaya
Department of Electrical and Computer Engineering
Rowan University
bouaynaya@rowan.edu

Abstract

Motivated by graph theory, artificial neural networks (ANNs) are traditionally structured as layers of neurons (nodes), which learn useful information by the passage of data through interconnections (edges). In the machine learning realm, graph structures (i.e., neurons and connections) of ANNs have recently been explored using various graph-theoretic measures linked to their predictive performance. On the other hand, in network science (NetSci), certain graph measures including entropy and curvature are known to provide insight into the robustness and fragility of real-world networks. In this work, we use these graph measures to explore the robustness of various ANNs to adversarial attacks. To this end, we (1) explore the design space of inter-layer and intra-layers connectivity regimes of ANNs in the graph domain and record their predictive performance after training under different types of adversarial attacks, (2) use graph representations for both inter-layer and intra-layers connectivity regimes to calculate various graph-theoretic measures, including curvature and entropy, and (3) analyze the relationship between these graph measures and the adversarial performance of ANNs. We show that curvature and entropy, while operating in the graph domain, can quantify the robustness of ANNs without having to train these ANNs. Our results suggest that the real-world networks, including brain networks, financial networks, and social networks may provide important clues to the neural architecture search for robust ANNs. We propose a search strategy that efficiently finds robust ANNs amongst a set of well-performing ANNs without having a need to train all of these ANNs.

1 Introduction

One of the most significant features of today's Artificial Intelligence (AI) is the existence of two competing paradigms, the *symbolic approach* and the *connectionist approach* [1]. In this work, we

*corresponding author

Preprint. Under review.

explore Artificial Neural Networks (ANNs) through the prism of connectionist approach. Inspired by the recent work in neuroscience, where specific graph measures were used to quantify the robustness of brain networks to various insults [2], this paper endeavors to explore these measures for analyzing the robustness of various ANNs. We begin by generating random graphs (in the graph domain), converting these graphs to ANNs (deep learning (DL) domain), training these ANNs for various image classification tasks, and finally evaluating these trained ANNs under adversarial attacks. Our analysis of graph-theoretical measures and adversarial performance of trained ANNs reveals their mutual relationship. We propose an efficient algorithm that can help users choose a robust ANN from any set of ANN architectures without the need to go through the extensive cycle of training, validation, testing, and comparing all ANNs under consideration. For clarity, we use the term *ANN* for artificial neural networks, *graphs* for the unweighted directed acyclic graphs, and the term *network* for various networks as used in the Network Science (NetSci) domain.

A few questions that stem out of the connectionist approach are, (1) can we relate an ANN's function to its graph structure? (2) how functional aspects of an ANN change with its graph structure? and (3) if such a relationship exists, are there any characterizations that explain the relationship between the graph structure and function of an ANN, especially their performance under adversarial attacks? Researchers have recently reported the relationship between an ANN's predictive accuracy and its underlying graph structure [3, 4, 5, 6, 7, 8, 9]. However, most of these architectures are custom-designed for the specific tasks in the ANN domain. Many researchers have proposed ANNs with a better predictive performance by varying wiring (i.e., links) between neurons and the operations they perform [10, 11, 12]. However, these efforts are expensive in terms of time and computational resources. Recently, Xie et al. [13] proposed a novel method of exploring a diverse set of connectivity patterns (or graph structures) through random graph-theoretic models. Moreover, You et al. [14] showed that the graph structure and predictive performance of ANNs are closely related. The work proposed in [13], referred to as *RandWire*, and in [14], referred to as *Relational Graphs*, showed that exploring a constrained search space for Neural Architecture Search (NAS) leads to better performing ANNs, quantified using test datasets. However, in the realm of AI safety, there is a need to explore the structures (or architectures) of ANNs to find robust ANNs for safety critical applications.

It has been shown in Percolation Theory that the underlying network structure of any real-world system has a key role in its robustness to attacks [15]. Moreover, functional robustness of networks, quantified by Tannenbaum et al. [16], has been successfully implemented in finding robustness of networks in NetSci. For example, analysis of cancer cells [17], fragility of financial networks [18], matching network embedding space with structure of data [19], random graphs [20], social communities' identity [21], robustness of brain networks and tracking changes to age and Autism Spectrum Disorder (ASD) [2], and to explain cognitive impairment in Multiple Sclerosis (MS) patients [22]. These works provide a solid foundation and motivation for studying the robustness of ANNs using graph-theoretic measures.

As shown in Figure 1(a), graph theory has long been used in NetSci to study various real-world networks using graph-theoretic measures. Examples include, biological systems such as brain networks, economic systems such as financial networks, and social systems such as social networks. The graph measures include path length, graph connectivity, efficiency, degree measures, clustering coefficient, and many more. The graph spectral analysis may include graph curvature and entropy using the graph Adjacency and/or Laplacian matrices. As illustrated by Figure 1(b), in our settings, the architectures of ANNs are represented by graphs based on the three classical families of random graph models, Watts-Strogatz (WS), Erdős Rényi (ER), and Barabási-Albert (BA) [23, 24, 25, 13, 14]. Given these graph representations of the architectures of ANNs, we can study various graph-theoretic properties. We hypothesize that the graph properties quantifying the robustness of different networks in the NetSci domain, will provide us with the insight into the robustness of ANNs. We provide experimental evidence to support our hypothesis and present an algorithm that selects a robust ANN from the given design space without the need to search and train ANN based on the graphs from the entire design space exhaustively.

2 Related work

Our work is a cross-pollination effort among three fields; graph theory, neuroscience, and machine learning (i.e., ANNs) . It is well-established that the structure of an ANN is closely related to its predictive performance [3, 4, 5, 6, 7, 8, 9]. It has been shown that the model architecture as a prior is

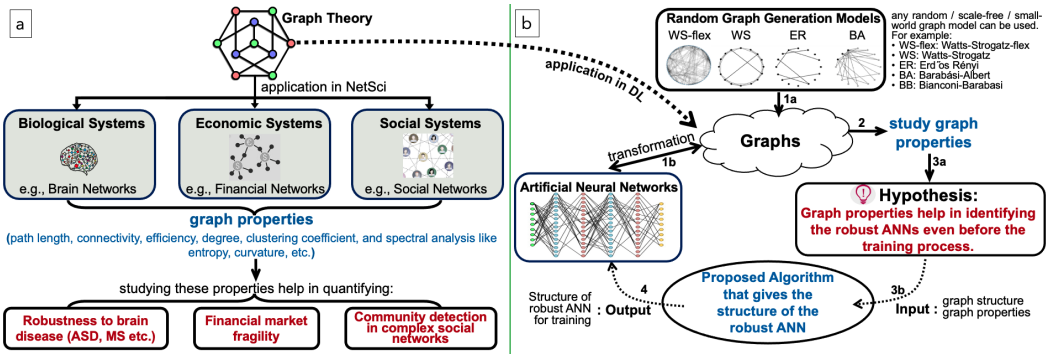


Figure 1: **Research Hypothesis. (a)** Graph theory has been applied in Network Science (NetSci) to study biological systems such as brain networks, economic systems such as financial networks, and social systems such as social networks. Graph theoretic measures (curvature, entropy, etc.) have been successfully employed to quantify robustness and fragility of these networks [17, 2, 22, 18, 21]. **(b) Step-1a** and **1b**: The graphs generated using classical families of random graph models (WS and WS-flex, ER, and BA) can be used to build Artificial Neural Networks (ANNs) through various transformations [13, 14]. **Step-2**: We can study various graph-theoretic properties of these random graphs. **Step-3a**: We **hypothesize** that the graph properties that quantify robustness of different networks in NetSci domain, can also help us in identifying the the ANNs that are robust to adversarial attacks. **Step-3b**: Such robust ANNs can be found through an algorithm, without the need to exhaustively train and search the entire ANNs' design space. The algorithm will take the graph properties as the input and, **step-4**: outputs the structure of the robust ANN for training.

non-trivial for downstream tasks [26]. Traditionally, we witness significant improvement in an ANN's predictive performance owing to the evolution of handcrafted wiring plans in the structure of ANNs. *RandWire* [13] showed that variants of random graph generators produce graphs of ANNs that have competitive accuracy as compared to their hand-crafted counterparts. *Relational graphs* depicted that the performance of an ANN is approximately a smooth function of graph properties (i.e., clustering coefficient and average path length). The authors also showed that the best-performing ANNs had graph structure similar to real biological brain networks [14]. Recently, *Robust Networks* (RobNets) investigated the patterns of ANN architectures that were resilient to adversarial attacks and found that densely connected patterns resulted in improved robustness [27]. However, most real-world networks are not dense but sparse, with relatively few edges between nodes [28].

Inspired by the biological neuronal networks of C. Elegans and the mouse visual cortex, the Deep Connectomics Networks (DCNs) were designed for the vision tasks [29]. Brain-inspired AI (BI-AI) systems have been shown to perform well on cognitive brain data, brain-computer interfaces, multi-sensory streaming data modelling in finance, environment and ecology [30]. Moreover, brain-inspired continual and incremental learning helps ANNs prevent catastrophic forgetting [31, 32].

Various graph theoretic properties have been studied to capture the functional robustness of networks [16]. Weighted graphs and their properties have been studied to distinguish cancer from normal cell networks [17]. In financial networks, the geometric feature called *curvature* has been shown to illustrate the market fragility [18]. In social networks, graph properties help in community identification in complex networks [21]. In neuroscience, graph properties quantify the robustness of brain networks, track changes caused by the age and Autism Spectrum Disorder (ASD), and to explain cognitive impairment in patients with Multiple Sclerosis (MS) [2, 22].

It has been reported that biologically-inspired ANNs are robust to adversarial attacks [33]. Neuro-inspired CNN with Feedback (CNN-F) is custom-designed CNN having dual pathways that provide robustness against pixel noise, occlusion, and blurring [34]. Biologically plausible mechanisms in primate visual ventral system have helped in improving the robustness of ANNs against adversarial perturbations [35]. VOneNets, a class of hybrid CNN vision models, have shown that precisely mimicking just one stage of the primate visual system in ANNs leads to better performance in ImageNet-level computer vision applications [36].

Among the traditional methods for improving the non-trivial task of adversarial robustness of ANNs [37, 38, 39], adversarial training [40] is the most propitious at finding the robust model from the given design-space of architectures. However, besides other drawbacks, adversarial training's biggest downside is the significantly large computational cost despite few improvements to speed-up the process [41, 42, 43, 44]. Therefore, the path to robust ANN design needs to be reconsidered to avoid the expensive training. In this work, we present one such path that is based on the successes in the fields of graph theory and NetSci.

3 Methodology

We now present our methodology, starting with the process of generating random graphs. Our motivation for using random graphs is based on the work of [45], i.e, a randomly organized ANN model analogizes more closely to a human brain . Moreover, brain networks are mostly random in terms of physical connections at the initial stages of construction which differs in individuals [46, 47].

3.1 Random Graph Generation

Connectivity patterns of ANNs represented as computational graphs is naturally captured through *inter-layer* and *intra-layer* connectivity. To minimize the human bias, we include one design example from each of these connectivity choices, (1) *RandWire* [13] belongs to inter-layer connectivity regime based on three classical families of random graph generators, i.e., WS, ER, and BA [23, 24, 25], (2) *Relational Graphs* are generated using intra-layer connectivity patterns through WS-flex model [14].

ER graph generator, represented by ER(P), generates graphs with N nodes having an edge with probability P. BA graph generator, represented by BA(M), iteratively and sequentially adds M new nodes (and non-duplicate edges) to the initial M nodes having no edges. WS graph generator, represented by WS(K, P), uses N initial nodes laid out in ring with each node connected to $K/2$ neighbors on each side. Moving in clockwise direction, each edge connecting two nodes is rewired with probability P, the loop is repeated $K/2$ times. For *RandWire*, we use generator notation $g(\theta, s)$, where g is the generator (ER, BA, WS), θ represents parameters (P, M, (K, P)), and s is the random seed [13].

WS-flex model generates *relational graphs* by generalizing WS model through relaxation of the same-degree constraint for all nodes. Parameterized by N nodes, K average degree, and P rewiring probability, these graphs are represented by WS-flex(N, K, P). It is important to note that ER, BA, and WS produce a subset of all possible N-node graphs with different underlying priors, whereas, WS-flex graph generator encompasses all the graphs generated by aforementioned classic random graph generators. As shown by [14], design space spanned by the global graph measure of *average path length* (L), and local graph measure of *clustering coefficient* (CC) smoothly spans all of the graphs generated by these graph generators. These two measures have been extensively used in prior works [23, 48, 49]. Therefore, our design space of graphs is spanned by *CC* and *L* as illustrated by Figure 2. We downsample and aggregate the 2.313M candidate WS-flex graphs into coarser bins of 3854, 441, and 53 graphs, where each bin has at least one representative graph. We also generated 28 graph samples from *RandWire* (20 from WS, 4 each from ER and BA). A total of 81 graphs are used in our experiments.

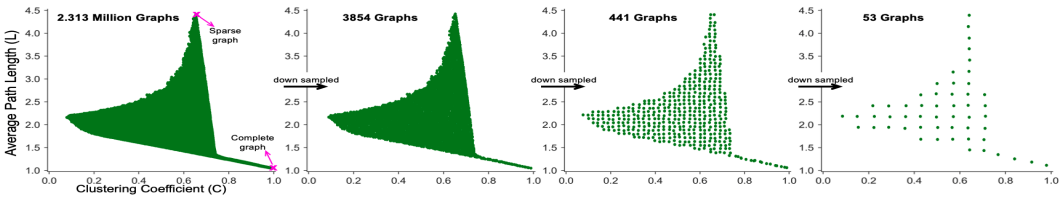


Figure 2: The design space of graphs is presented. The original 2.313 Million candidate graphs from WS-Flex graph generator are down sampled to 3854, 441, and finally 53 graphs.

3.2 Graph-Theoretical Properties

We consider following graph-theoretical measures in our analyses. **Average Path Length (L)** is a global graph measure defined as the average shortest path distance between any pair of nodes of the graph. It depicts the efficiency of the graph with which information or communication is transferred through the nodes [50]. Smaller L means the graph is globally efficient and the information is exchanged effectively across the whole network and vice versa.

Clustering Coefficient (C) is a measure of local connectivity of a graph. For a given node i in a graph, the probability that all its neighbors are also neighbors to each other is called clustering coefficient, and it ranges between 0 and 1. More densely interconnected the neighborhood of a node, higher is its measure of C. A large value of C is linked with the resilience against random network damage [51]. The small-worldness of networks is also assessed by the clustering coefficient [52].

Graph Spectral Measures focus on eigenvalues and eigenvectors of the associated graph Adjacency and Laplacian matrices. The measures of our particular interest are *topological entropy* and *curvature* of graphs.

Topological Entropy or simply *entropy* (H_G) of graph G having adjacency matrix A_G is the logarithm of the spectral radius of A_G, i.e., logarithm of the maximum of absolute values of the eigenvalues of A_G [53].

$$H_G = \log(\lambda_{A_G}) \tag{1}$$

Ollivier–Ricci Curvature (ORC) or simply *curvature* is the discrete analog of the Ricci curvature [54, 55]. Let (X, d) be a geodesic (a curve representing the shortest path between two points on a surface or in a Riemannian manifold) metric space having family of probability measures $\{p_x : x \in X\}$. Then ORC $\kappa_{ORC}(x, y)$ along the geodesic connecting x and y is,

$$\kappa_{ORC}(x, y) = 1 - \frac{W_1(p_x, p_y)}{d(x, y)} \tag{2}$$

where W_1 is the earth mover's distance (Wasserstein-1 metric), and d is the geodesic distance on the space. Curvature is directly proportional to robustness of the network. Larger the curvature, faster will be the return to the original state after perturbation. Smaller curvature means slow return, which is also called fragility [2].

Robustness is the rate at which a dynamic system returns to its original state after perturbation. Fluctuation Theorem [56] states that, given random perturbations to network, change in robustness ΔR is positively correlated to change in system entropy ΔH,

$$\Delta H \times \Delta R > 0. \tag{3}$$

Tannenbaum et al. [16] showed that entropy ΔH and curvature $\Delta\kappa_{ORC}$ are also positively correlated,

$$\Delta H \times \Delta\kappa_{ORC} > 0. \tag{4}$$

From equation (3) and (4), Tannenbaum et al. [16] drew the analogy that graph curvature and robustness are also positively correlated,

$$\Delta\kappa_{ORC} \times \Delta R > 0. \tag{5}$$

Having generated graphs and studied their properties in graph theory domain, we now transition to the DL domain for analyzing the corresponding ANNs.

3.3 Transforming Graphs' Structures to ANNs' Architectures

For transforming graphs into ANNs, we define *edge* as the directed transfer of tensor between nodes it connects, whereas *node* depicts the troika of aggregation, transformation, and distribution. Important to note here is that the transformation encapsulates ReLU-Conv-BN operations [57] with 3x3 separable Conv [58]. Stages, strides, and rate of change in channel count have been kept the same as [13] for the ANNs generated through *RandWire* (WS, ER, BA) graphs. For generating different ANNs from *relational graphs* (WS-flex), we kept the message exchange, node feature, and aggregation function definitions same as [14]. We generated 5-layer Multilayer Perceptron (MLP) with 512 hidden units and 6-layer CNN with 3x3 convolutions for each of the 53 WS-flex graphs. For

the *RandWire* graphs, we generated 5-layer CNNs with 3x3 convolutions for each of the 28 graphs. To expand on the model categories, we then generated ResNet-18 networks from each of the 53 WS-flex graphs. Total 187 ANNs were generated (53x3 for WS-flex, 28 for RandWire graphs).

Training, Validating and Testing ANNs: We used CIFAR-10 and CIFAR-100 datasets [59] for training 53 MLPs, 53 CNNs from *WS-flex*, and 28 CNNs from *RandWire*. Our experiments also included Tiny ImageNet dataset with ResNet-18 [60] for 53 WS-flex ANNs. For verifying the hypothesis on varying number of nodes, we kept n=64 nodes for WS-flex ANNs and n=32 nodes with C=109 channels for *RandWire* ANNs. For each instance of ANNs, a random seed value s=5 was used to train and evaluate on CIFAR-10 and CIFAR-100 datasets and the average validation accuracy was recorded for all five random seeds. The hyperparameters for training and evaluating MLPs, CNNs, and ResNet-18s were kept the same as [14, 13], except the batch size of 1024 and 100 epochs for CNNs. Complete set of hyperparameters is given in the supplementary section. Due to limitations in computational resources and large training times, we did not carryout experiments on ImageNet dataset [61].

Testing Under Adversarial Attacks: After training the aforementioned ANNs, we subjected them to testing with adversarial examples; for a valid input x_1 and a target class y_1, it is possible to find x_1' through imperceptible non-random perturbation to x_1 which changes ANN's prediction to some other y_2; such x_1' is called an *adversarial example*. From the commonly employed *White-box* attacks [40, 62, 63, 64, 65], where model architecture, parameters, and training data is known to the attacker, we used Fast Gradient Sign Method (FGSM) [62, 63] and Projected Gradient Descent (PGD) [40] as adversarial attacks in our experiments with noise ϵ=[0.05, 0.1, 0.2, 0.3] for FGSM and ϵ=8/255 and α=1/255 for PGD.

ANN Performance and Graph Measures: We compare the ANNs' predictive performance with graph-theoretic measures of their corresponding graphs. We calculated the graphs-theoretic measures L, C, H, and κ_{ORC} given in section 3.2 for each of the 81 graphs. For the corresponding ANNs of these graphs, we recorded their predictive performance under adversarial as well as non-adversarial conditions. As per our hypothesis, certain ANNs should report higher predictive performance under adversarial conditions.

Algorithm 1 Robust Model Selection

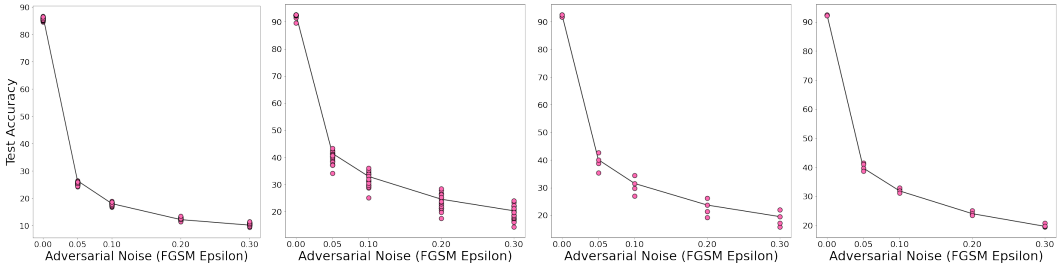
Input: data x, task t, nodes n, number of graphs to be selected α
repeat
  for (τ graphs $\in$ n-node design space) **do**
    find all graphs $G_\tau(n)$
    calculate $\kappa_{OR}(G_\tau)$, $H(G_\tau)$ (graph measures)
  end for
  for $i \leftarrow 1$ **to** α **do**
    G_i = max($G_\tau(\kappa_{OR}, H)$) (graphs having highest curvature, entropy measures)
    $[RG_\alpha] \leftarrow G_i$ (add to the list of robust graphs)
  end for
until (α **robust graphs** (RG_α) **found**)
{required number of graphs found}
repeat
  convert robust graphs to neural networks, $RG_\alpha \rightarrow NN_\alpha$
  for $j \leftarrow 1$ **to** α **do**
    train, validate $NN_j(x, t)$
    calculate adversarial accuracy ($AdvAcc(NN_j)$
  end for
  select $Robust\ NN = max(AdvAcc(NN_j))$
until (**Robust Neural Network found**)

Search for Robust ANNs in the Graph Domain: We propose a search method for robust ANNs based on the graph properties of ANNs and avoiding exhaustive search. Algorithm 1 finds the architecture of an ANN that is robust under adversarial conditions without the need to undergo the train-validate-test loop for all the given models/ choices of architectures. The input parameters of the algorithm include dataset x, task t, number of nodes n, and the number of ANNs to be considered

α. For all possible (τ) number of graphs in the given design space of n-node graphs, calculate curvature (κ_{ORC}) and entropy (H). Select α number of graphs having highest κ_{ORC} and H values. Convert these α graphs to ANNs. All these α ANNs have higher robustness than the rest of ($\tau - \alpha$) ANNs. Train α ANNs for the downstream task t and select the ANN with the highest adversarial accuracy. Selection of parameter α is at the user's discretion as per availability of resources such as computational power and time, generally $\alpha \leq$10.

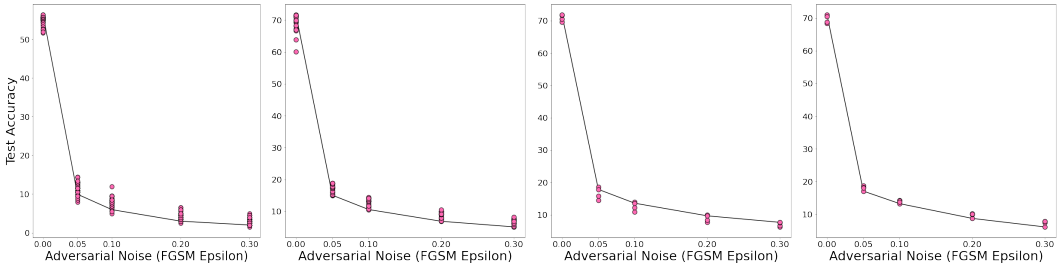
4 Results

The test accuracy of trained ANNs under different adversarial conditions is compared with their clean accuracy. We also compare κ_{ORC} and H graph measures with adversarial performance of ANNs.



(a) WS-flex CNNs (b) WS CNNs (c) BA CNNs (d) ER CNNs

Figure 3: **FGSM attack results for CNN on CIFAR-10**. CNNs from the four graph models (WS-flex, WS, BA, ER) trained and tested on CIFAR-10 under varying adversarial noises, i.e., ϵ=[0, 0.05, 0.1, 0.2, 0.3], where ϵ=0 means *no adversarial attack*. Black line depicts the best performing CNN at ϵ=0 when subjected to higher ϵ values. This CNN is no more the best network as ϵ increases.



(a) WS-flex CNNs (b) WS CNNs (c) BA CNNs (d) ER CNNs

Figure 4: **FGSM attack results for CNN on CIFAR-100**. Our CNNs trained and tested on CIFAR-100 under varying adversarial noises, i.e., ϵ=[0, 0.05, 0.1, 0.2, 0.3]. Black line depicts the best performing CNN at ϵ=0 when subjected to higher ϵ values which is not the best for higher ϵ.

CIFAR-10/100 Datasets: The first set of experiments is for the CIFAR-10 [59] dataset using CNNs belonging to the WS-flex, WS, BA, and ER graph models. The test accuracy of these CNNs is plotted against various levels of adversarial noise (ϵ) for the FGSM attacks in Figure 3. As seen in Figure 3(a), the difference between highest and lowest test accuracy of CNN for WS-flex models under non-adversarial conditions is 1.98%, whereas, this difference increases to 2.11% at highest value of $\epsilon = 0.3$. This spread is more visible in case of WS CNNs (3.1% at $\epsilon = 0$ to 9.6% at $\epsilon = 0.3$), BA CNNs (0.87% at $\epsilon = 0$ to 6.27% at $\epsilon = 0.3$), and ER CNNs (0.37% at $\epsilon = 0$ to 1.35% at $\epsilon = 0.3$), Figure 3(b)(c)(d). This means that the performance of CNNs in each category of models is almost the same in the absence of any adversarial noise but they perform very differently under adversarial attacks. The second set of experiments is on CIFAR-100 [59] dataset using CNNs belonging to the WS-flex, WS, BA, and ER graph models. Test accuracy of these CNNs is plotted against various levels of adversarial noise (ϵ) for the FGSM attacks in Figure 4. For these CNNs, the difference between highest and lowest test accuracy of CNN for all categories of models does not show the behavior of spreading out at higher ϵ values because of the increase in the dataset complexity whereas

model architectures are the same as tested on CIFAR-10 dataset. We notice in our results on both the datasets that the best performing CNN under non-adversarial conditions does not remain the best as we increase the adversarial noise from $\epsilon = 0 \rightarrow 0.3$. This can be seen by tracing the black line in the respective plots of Figure 3 and 4. This shows that our best CNN under non-adversarial conditions is not the most robust one in each model category. Now let us compare ANNs' adversarial accuracies with graph measures to assess their robustness.

Graph Measures and Adversarial Performance: Graph curvature and entropy values plotted against adversarial performance of ANNs is given in Figure 5 for WS-flex CNNs on CIFAR-10, Figure 6 for WS and BA CNNs on CIFAR-10, Figure 7 for WS-flex CNNs on CIFAR-100, and Figure 8 for WS, BA, and ER CNNs on CIFAR-100. Each datum represents a CNN, color-coded according to its curvature and entropy measures. As seen in the plots, the CNNs having higher curvature and entropy values come out to be among the best CNNs under higher adversarial conditions for all categories of models and lie in the top right quadrant of each plot. These CNNs bearing higher curvature-entropy values prove to be consistently robust.

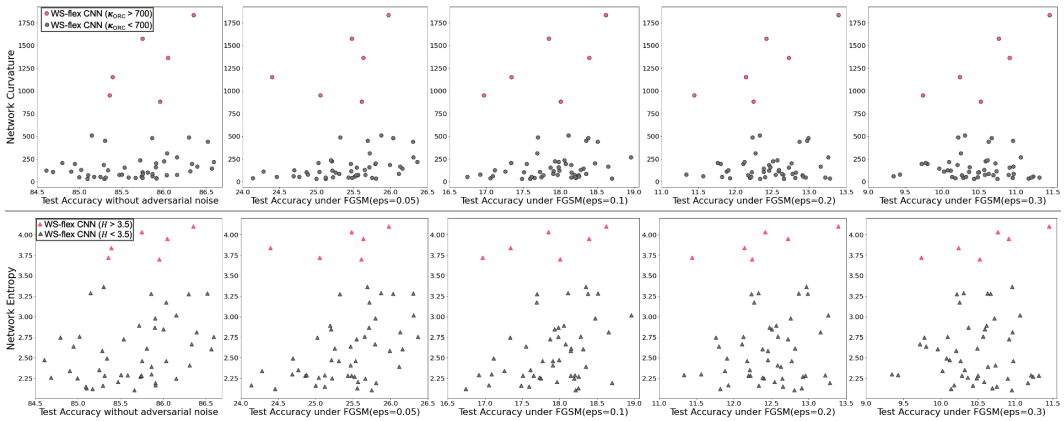


Figure 5: Graph measures *vs.* adversarial performance for WS-flex CNNs on CIFAR-10. Each datum is a CNN. Top row **(Curvature *vs.* Test Accuracy under FGSM attack)**. Bottom row **(Entropy *vs.* Test Accuracy under FGSM attack)**. CNNs with higher graph values (κ_{ORC}>700, H>3.5) perform well under non-adversarial (ϵ=0) as well as under adversarial attacks (ϵ=0.05, 0.1, 0.2, 0.3). At ϵ=0.3, the best CNN is the one with the highest κ_{ORC} and H values. The robust CNNs (having higher κ_{ORC}, H, and accuracies) lie in the top right portion of each plot.

Further Experiments and Failure Cases: We extended our experiments to include Projected Gradient Descent (PGD) attacks [40] for CIFAR-10 and CIFAR-100 datasets. We also experimented with Tiny ImageNet dataset [60] for the 53 WS-flex ANNs. These results are given in the supplementary section. In our experiments with ER CNNs on CIFAR-10 dataset, we noticed that these CNNs have negligible accuracy spread across all values of ϵ and so the robustness measures are not profoundly variable. This is possibly because each edge's wiring in ER(P) is independent. We give these plots in the supplementary section.

Analysis of the Search Algorithm: Based on our results, we analyze our search algorithm 1 for the robust ANN. Let us analyze this algorithm on our experiments with WS-flex CNNs on CIFAR-10 dataset. Given the dataset x=*CIFAR10*, task t=*image classification*, n=*64* number of nodes, and α=*6* ANNs to be selected out of the total τ=*2.313* Million graphs. We calculate H and κ_{ORC} given by Eq. 1 and 2 respectively. The α=*6* graphs with highest values of H and κ_{ORC} are $G(\kappa_{ORC}$,H)={(1835.88, 4.10), (1577.94, 4.03), (1365.30, 3.95), (1154.22, 3.83), (951.0, 3.72), (884.40, 3.70)}. Under highest value of ϵ=0.3 for FGSM attack, the predictive performance of corresponding 5-layer MLPs is *Test Accuracy*={21.33, 21.40, 21.23, 21.14, 21.03, 20.99} respectively which is higher than all the ANNs under evaluation. The best ANN at ϵ=0 has $G(\kappa_{ORC}, H)$=(77.83, 2.2) which is no more among the best ANNs at higher ϵ values. Plots for WS-flex MLPs on CIFAR-10 are given in supplementary section.

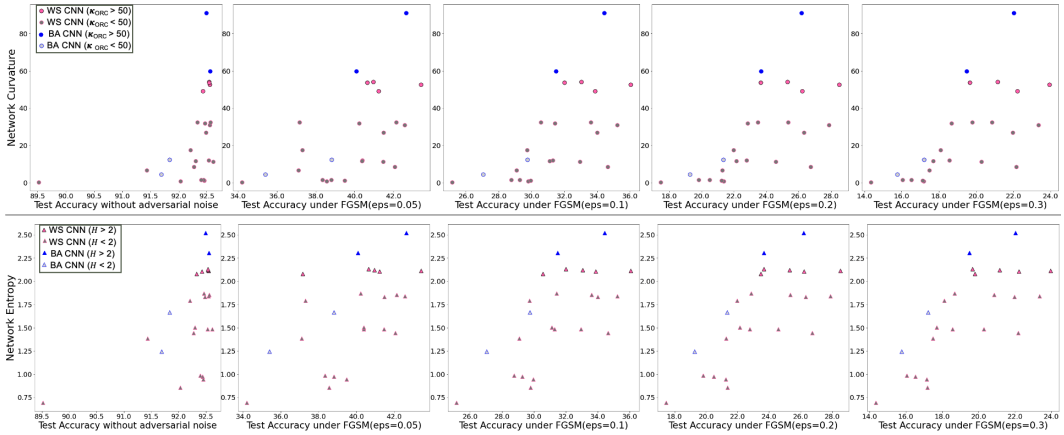


Figure 6: Graph measures *vs.* adversarial performance for WS and BA CNNs on CIFAR-10. Each datum is a CNN. Refer to the figure legends for each category of network. Top row **(Curvature *vs.* Test Accuracy under FGSM attack)**. Bottom row **(Entropy *vs.* Test Accuracy under FGSM attack)**. Among the best performing CNNs at all levels of adversarial noise (ϵ), the ones with higher curvature and entropy values are consistent and they lie in the top right portion of each plot.

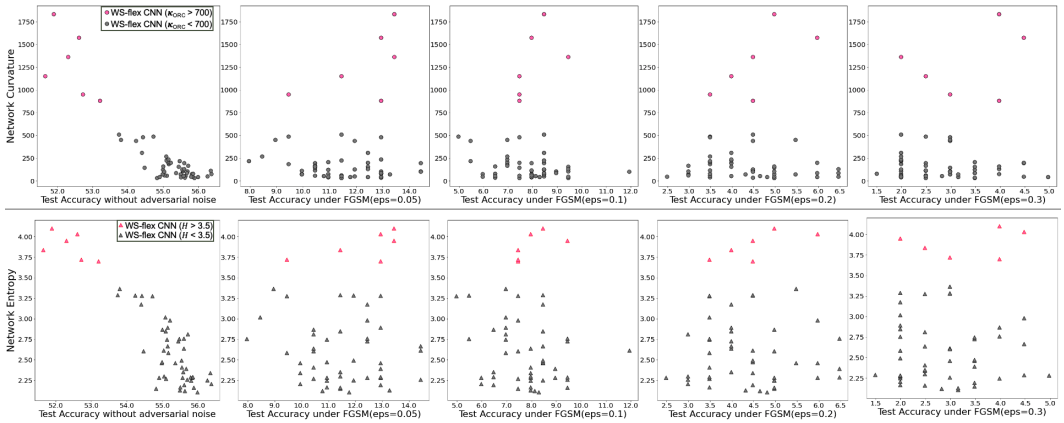


Figure 7: Graph measures *vs.* adversarial performance for WS-flex CNNs on CIFAR-100. Each datum is a CNN. Top row **(Curvature *vs.* Test Accuracy under FGSM attack)**. Bottom row **(Entropy *vs.* Test Accuracy under FGSM attack)**. CNNs with lower curvature and entropy values perform well under non-adversarial (ϵ=0), but as (ϵ) increases, the CNNs with higher curvature and entropy values (κ_{ORC}>700, H>3.5) perform better and these CNNs lie in the top right portion of each plot.

5 Broader Impact

Our work offers a novel approach to quantify robustness of ANNs through graph curvature and entropy. We present an algorithm to efficiently find the most robust ANNs by avoiding the search mechanisms that involve exhaustive training and evaluation. We believe that this work will have a high impact on the acceptability of safety-critical AI applications.

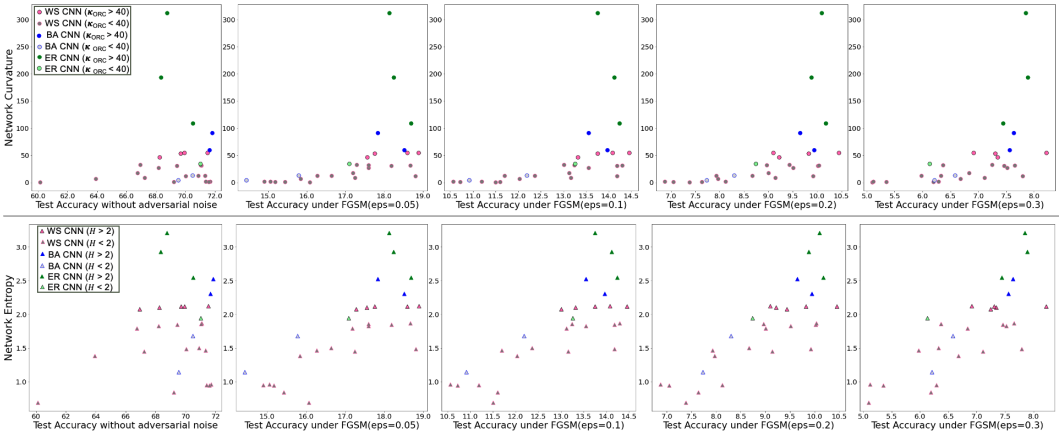


Figure 8: Graph measures *vs.* adversarial performance for WS, BA, ER CNNs on CIFAR-100. Each datum is a CNN. Refer to the figure legends for each category of network. Top row **(Curvature *vs.* Test Accuracy under FGSM attack)**. Bottom row **(Entropy *vs.* Test Accuracy under FGSM attack)**. Among the best performing CNNs at all levels of adversarial noise (ϵ), the ones with higher curvature and entropy values are consistent and they lie in the top right portion of each plot.

References

[1] Ron Sun. Artificial intelligence: Connectionist and symbolic approaches. 1999.

[2] Hamza Farooq, Yongxin Chen, Tryphon T Georgiou, Allen Tannenbaum, and Christophe Lenglet. Network curvature as a hallmark of brain structural connectivity. *Nature communications*, 10(1):1–11, 2019.

[3] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[4] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[5] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[10] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[11] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[12] Chris Ying, Aaron Klein, Eric Christiansen, Esteban Real, Kevin Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *International Conference on Machine Learning*, pages 7105–7114. PMLR, 2019.

[13] Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. Exploring randomly wired neural networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1284–1293, 2019.

[14] Jiaxuan You, Jure Leskovec, Kaiming He, and Saining Xie. Graph structure of neural networks. In *International Conference on Machine Learning*, pages 10881–10891. PMLR, 2020.

[15] Albert-László Barabási et al. *Network science*. Cambridge university press, 2016.

[16] Allen Tannenbaum, Chris Sander, Liangjia Zhu, Romeil Sandhu, Ivan Kolesov, Eduard Reznik, Yasin Senbabaoglu, and Tryphon Georgiou. Ricci curvature and robustness of cancer networks. *arXiv preprint arXiv:1502.04512*, 2015.

[17] Romeil Sandhu, Tryphon Georgiou, Ed Reznik, Liangjia Zhu, Ivan Kolesov, Yasin Senbabaoglu, and Allen Tannenbaum. Graph curvature for differentiating cancer networks. *Scientific reports*, 5(1):1–13, 2015.

[18] Romeil S Sandhu, Tryphon T Georgiou, and Allen R Tannenbaum. Ricci curvature: An economic indicator for market fragility and systemic risk. *Science advances*, 2(5):e1501495, 2016.

[19] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2018.

[20] Yong Lin, Linyuan Lu, and Shing-Tung Yau. Ricci curvature of graphs. *Tohoku Mathematical Journal, Second Series*, 63(4):605–627, 2011.

[21] Jayson Sia, Edmond Jonckheere, and Paul Bogdan. Ollivier-ricci curvature-based method to community detection in complex networks. *Scientific reports*, 9(1):1–12, 2019.

[22] Hamza Farooq, Christophe Lenglet, and Flavia Nelson. Robustness of brain structural networks is affected in cognitively impaired ms patients. *Frontiers in neurology*, 11:1542, 2020.

[23] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[24] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[25] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[26] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[27] Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. When nas meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 631–640, 2020.

[28] Gerardo Iñiguez, Federico Battiston, and Márton Karsai. Bridging the gap between graphs and networks. *Communications Physics*, 3(1):1–5, 2020.

[29] Nicholas Roberts, Dian Ang Yap, and Vinay Uday Prabhu. Deep connectomics networks: Neural network architectures inspired by neuronal networks. *arXiv preprint arXiv:1912.08986*, 2019.

[30] Nikola K Kasabov. *Time-space, spiking neural networks and brain-inspired artificial intelligence*. Springer, 2019.

[31] Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020.

[32] Kaushalya Kumarasinghe, Nikola Kasabov, and Denise Taylor. Brain-inspired spiking neural networks for decoding and understanding muscle activity and kinematics from electroencephalography signals during hand movements. *Scientific reports*, 11(1):1–15, 2021.

[33] Aran Nayebi and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202*, 2017.

[34] Yujia Huang, Sihui Dai, Tan Nguyen, Pinglei Bao, Doris Y Tsao, Richard G Baraniuk, and Anima Anandkumar. Brain-inspired robust vision using convolutional neural networks with feedback. *NeurIPS 2019 Workshop*, 2019.

[35] Manish V Reddy, Andrzej Banburski, Nishka Pant, and Tomaso Poggio. Biologically inspired mechanisms for adversarial robustness. *arXiv preprint arXiv:2006.16427*, 2020.

[36] Joel Dapello, Tiago Marques, Martin Schrimpf, Franziska Geiger, David D Cox, and James J DiCarlo. Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. *BioRxiv*, 2020.

[37] Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018.

[38] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018.

[39] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[40] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[41] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

[42] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5636–5643, 2020.

[43] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[44] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *arXiv preprint arXiv:2007.02617*, 2020.

[45] Alan Mathison Turing. Intelligent machinery, 1948.

[46] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[47] Benoit Siri, Mathias Quoy, Bruno Delord, Bruno Cessac, and Hugues Berry. Effects of hebbian learning on the dynamics and structure of random networks with inhibitory and excitatory neurons. *Journal of Physiology-Paris*, 101(1-3):136–148, 2007.

[48] Olaf Sporns. Graph theory methods for the analysis of neural connectivity patterns. In *Neuroscience databases*, pages 171–185. Springer, 2003.

[49] Danielle Smith Bassett and ED Bullmore. Small-world brain networks. *The neuroscientist*, 12 (6):512–523, 2006.

[50] Mite Mijalkov, Ehsan Kakaei, Joana B Pereira, Eric Westman, Giovanni Volpe, and Alzheimer's Disease Neuroimaging Initiative. Graph measures, 2017. URL `http://braph.org/manual/graph-measures/`.

[51] C.J.Stam. Connected brains: introduction to graph theory, 2020. URL `https://home.kpn.nl/stam7883/graph_introduction.html`.

[52] Naoki Masuda, Michiko Sakaki, Takahiro Ezaki, and Takamitsu Watanabe. Clustering coefficients for correlation networks. *Frontiers in neuroinformatics*, 12:7, 2018.

[53] Yongxin Chen, Tryphon Georgiou, Michele Pavon, and Allen Tannenbaum. Robust transport over networks. *IEEE transactions on automatic control*, 62(9):4675–4682, 2016.

[54] Yann Ollivier. Ricci curvature of metric spaces. *Comptes Rendus Mathematique*, 345(11): 643–646, 2007.

[55] Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.

[56] Lloyd A Demetrius. Boltzmann, darwin and directionality theory. *Physics reports*, 530(1):1–85, 2013.

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[58] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[59] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[60] Kaggle. Tiny imagenet. `https://www.kaggle.com/c/tiny-imagenet/overview`, 2021. Accessed: 05-25-2021.

[61] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[63] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[64] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[65] 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.

APPENDIX

A Further Experiments and Results

Here we provide additional results of our experiments on CIFAR-10 and Tiny ImageNet datasets. The 53 graphs from WS-flex models converted into 53 Multilayer Perceptron networks (MLPs) are trained on CIFAR-10 dataset and tested under FGSM and PGD attack conditions. Furthermore, these 53 WS-flex graphs transformed into 53 ResNet18 ANNs are trained on Tiny ImageNet dataset and tested under FGSM and PGD attack conditions.

A.1 WS-flex MLPs on CIFAR-10 dataset

As seen in Figure 9(a), the 53 WS-flex MLPs trained on CIFAR-10 dataset have comparable test accuracies for all values of FGSM ϵ, i.e., the difference between highest and lowest test accuracy of these MLPs at each level of adversarial noise is almost same (1.59% at $\epsilon = 0$, 0.92% at $\epsilon = 0.05$, 1.07% at $\epsilon = 0.1$, 1.46% at $\epsilon = 0.2$, and 1.68% at $\epsilon = 0.3$). However, when these 53 MLPs are subjected to PGD attack, this difference becomes significantly high (8.5% under PGD attack). This means that the performance of these MLPs is almost the same in the absence of any adversarial noise and under FGSM attacks but they perform very differently under the PGD adversarial attacks. We opine that this happens because of two reasons; firstly the complexity level of MLPs is very low as compared to other ANNs in our experiments, secondly the iterative nature of PGD attacks have amplified effect on the performance of simple-structured ANNs like MLPs. Now let us compare these MLPs' adversarial accuracies with graph measures to assess their robustness.

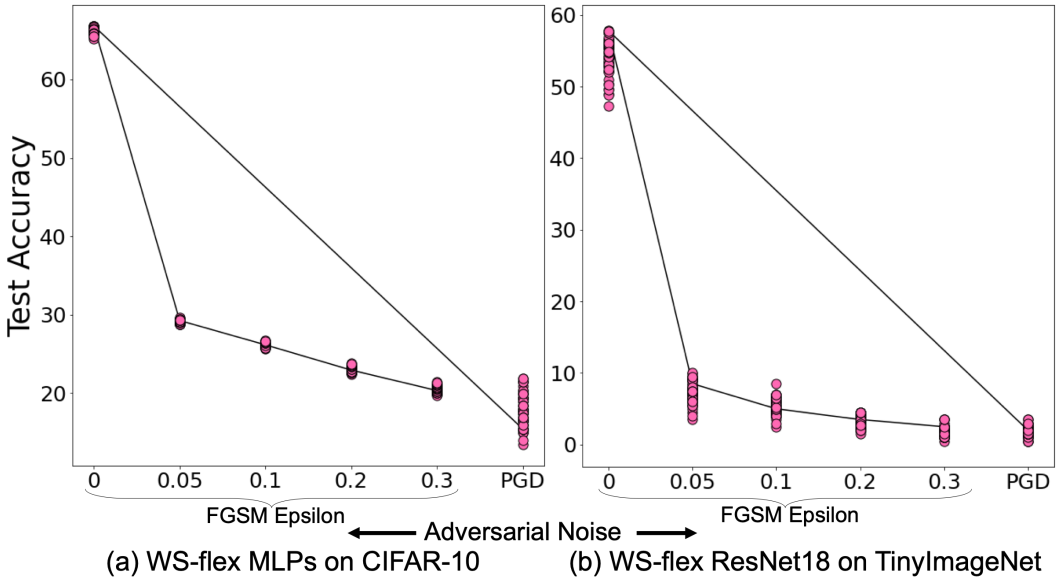


Figure 9: **(a) Adversarial attack results for WS-flex MLPs on CIFAR-10**. 53 WS-flex MLPs trained and tested on CIFAR-10 under varying FGSM adversarial noises (ϵ=[0, 0.05, 0.1, 0.2, 0.3]) and PGD attack. Black line depicts the best performing MLP at ϵ=0 (no adversarial attack). When subjected to higher ϵ values, this best MLP (at ϵ=0) does not remain the best.
(b) Adversarial attack results for WS-flex ResNet18 on TinyImageNet. 53 WS-flex ResNets trained and tested on TinyImageNet under varying FGSM adversarial noises (ϵ=[0, 0.05, 0.1, 0.2, 0.3]) and PGD attack. Black line depicts the best performing ResNet18 at ϵ=0 (no adversarial attack). When subjected to higher ϵ values, this best ResNet (at ϵ=0) is still among the best.

Graph Measures and Adversarial Performance: Graph curvature and entropy values plotted against adversarial performance of our 53 WS-flex MLPs is given in Figure 10 for CIFAR-10 dataset. Each datum represents an MLP, with pink color for MLPs having curvature $\kappa_{ORC} > 700$, and entropy $H_G > 3.5$. As seen in the plots, MLPs with high curvature and entropy values come out to be among

the best MLPs under higher adversarial conditions and lie in the top right quadrant of plots. These MLPs bearing higher curvature-entropy values prove to be robust among the 53 MLP samples.

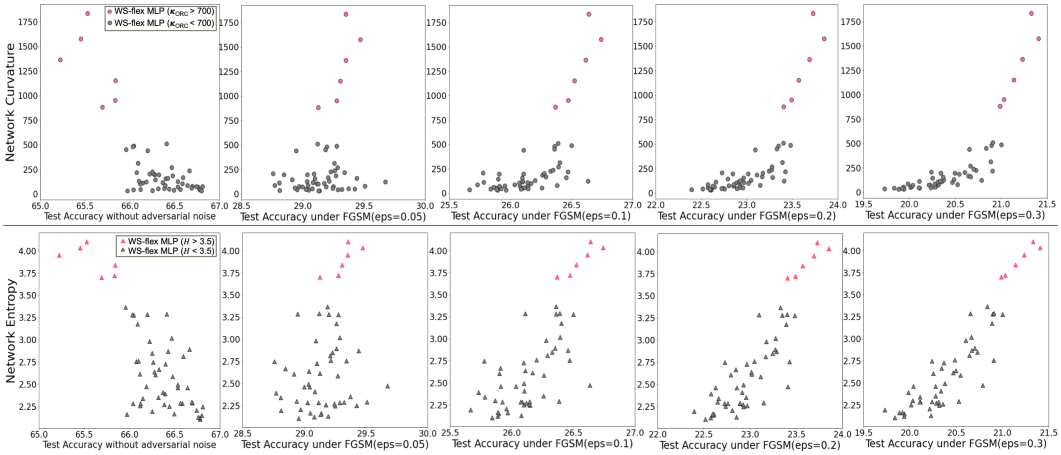


Figure 10: Graph measures *vs.* adversarial performance for WS-flex MLPs on CIFAR-10. Each datum is an MLP. Top row **(Curvature *vs.* Test Accuracy under FGSM attack)**. Bottom row **(Entropy *vs.* Test Accuracy under FGSM attack)**. MLPs with higher graph values (κ_{ORC}>700, H_G>3.5), depicted by pink-colored datums do not perform well under non-adversarial conditions (ϵ=0) as compared to other MLPs. However, as FGSM adversarial attack noise increases (ϵ=[0.05, 0.1, 0.2, 0.3]), the MLPs having higher graph measures out-perform other MLPs. At ϵ=0.3, the best MLP is the one with the highest κ_{ORC} and H values. These robust MLPs lie in the top right quadrant of the higher ϵ plots.

A.2 WS-flex ResNet18 on Tiny ImageNet

We also experimented with Tiny ImageNet dataset for the 53 WS-flex ANNs. As seen in Figure 9(b), the 53 WS-flex ResNets trained on Tiny ImageNet dataset have varying test accuracies under non-adversarial conditions ($\epsilon = 0$). This is due to the complexity of both the dataset and the models being trained on this dataset. As values of FGSM ϵ increases, the ResNets tend to have comparable performance. Moreover, by tracing the black line we notice that the ResNet which was the best under non-adversarial conditions, continue to be among the best performing ResNets under higher adversarial noise values. We will see in subsequent plots that this ResNet has higher values of curvature and entropy. Now let us look at the curvature and entropy plots for these ResNets to assess their robustness.

Graph Measures and Adversarial Performance: Graph curvature and entropy values plotted against adversarial performance of our 53 WS-flex ResNet18 models is given in Figure 11 for Tiny ImageNet dataset. Each datum represents a ResNet18 model, with pink color for ResNets having curvature $\kappa_{ORC} > 700$, and entropy $H_G > 3.5$. As seen in the plots, the ResNets having higher curvature and entropy values are the best even before they are subjected to adversarial noise. And under adversarial conditions, these ResNets continue to prove themselves to be the robust choices regardless of the level of adversarial noise value.These robust ResNets lie in the top right quadrant of all the plots of Figure 11.

A.3 Projected Gradient Descent (PGD) Attacks

Projected Gradient Descent (PGD) attacks are also included for our experiments on CIFAR-10, CIFAR-100, and Tiny ImageNet datasets. The CIFAR results are shown in Figure 12. The 53 MLPs and 53 CNNs from WS-flex graph models were subjected to PGD attacks for CIFAR-10 and CIFAR-100 datasets respectively. Performance comparison of these ANNs without any adversarial attack versus the same ANNs under PGD attacks is shown in the Figure 12. We see that ANNs having higher curvature and entropy values are consistently among the best performing ANNs under PGD attacks not only for the different ANN types (MLPs and CNNs), but also for the different dataset

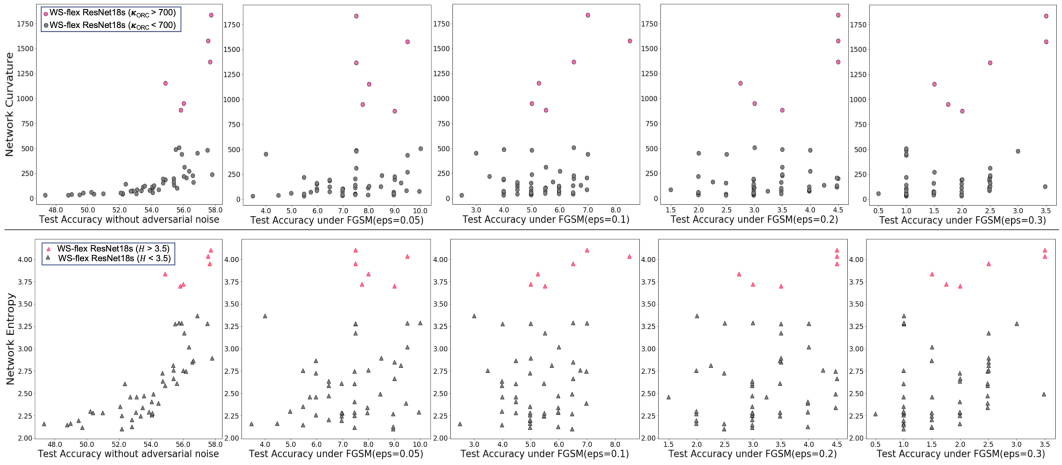


Figure 11: Graph measures *vs.* adversarial performance for WS-flex ResNet18s on Tiny ImageNet. Each datum is a ResNet18 ANN. Top row **(Curvature *vs.* Test Accuracy under FGSM attack)**. Bottom row **(Entropy *vs.* Test Accuracy under FGSM attack)**. ResNets with higher graph values (κ_{ORC}>700, H_G>3.5), depicted by pink-colored datums, are the best performing models under non-adversarial conditions (ϵ=0). These ResNets continue to be among the best as FGSM adversarial attack noise increases (ϵ=[0.05, 0.1, 0.2, 0.3]). These robust ResNets lie in the top right quadrant of each plot.

tasks (CIFAR-10 and CIFAR-100). So our algorithm for robust model selection presented in the main paper stands valid on the ANNs under PGD attacks as well.

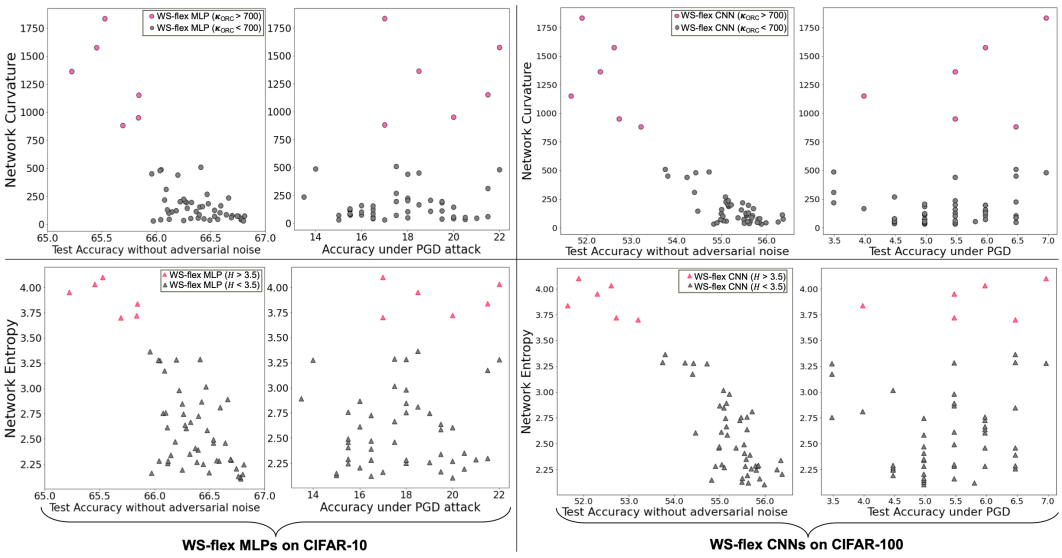


Figure 12: Comparison of graph curvature and entropy measures against performance of WS-flex ANNs under PGD attack. **(Left two columns): WS-flex MLPs on CIFAR-10.** **(Right two columns): WS-flex CNNs on CIFAR-100.** Top row depicts curvature *vs.* performance of ANNs without adversarial attack (ϵ=0) and under PGD attack. Bottom row depicts entropy *vs.* performance of ANNs without adversarial attack (ϵ=0) and under PGD attack. For both datasets, better performing ANNs under PGD attack are among those having higher curvature and entropy values.

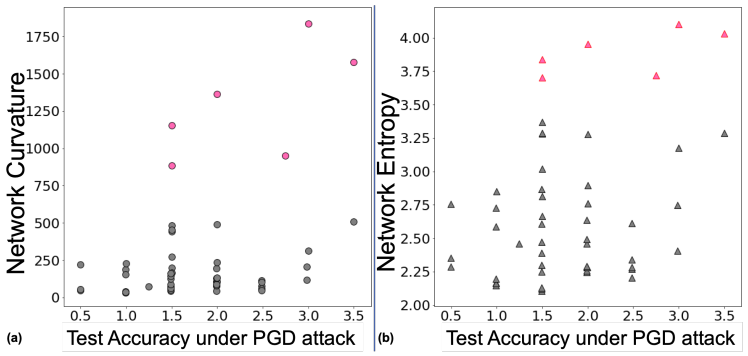


Figure 13: Comparison of graph curvature and entropy measures against performance of **WS-flex ResNet18s on Tiny ImageNet dataset under PGD attack**. **(a)**: Network Curvature *vs.* adversarial accuracy under PGD attack. **(b)**: Network Entropy *vs.* adversarial accuracy under PGD attack. The ResNet18 model having higher curvature and entropy values are among the best performing models under PGD attacks.

A.4 Exceptional Cases

In our experiments with ER CNNs on CIFAR-10 dataset, we noticed that these CNNs have similar accuracy spread ($\approx 2\%$) across all values of FGSM attack noise (ϵ=[0.05, 0.1, 0.2, 0.3]). Moreover, the graph-theoretic measures of curvature and entropy for these graph models are higher than their corresponding counter-parts from WS and BA models. The minimum values of graph measures in our sample set of ER models ($\kappa_{ORC} = 34.01$, $H_G = 1.95$) are close to the thresholds selected for performance comparison ($\kappa_{ORC} = 50$ and $H_G = 2$). This means that the resulting robustness measures of ER CNNs are not profoundly different from each other. We are of the opinion that this is possibly because each edge's wiring in ER(P) is independent and ER models inherently have higher curvature and entropy values. So we cannot clearly pick the robust CNN from the sample set of ER models, especially in case of smaller dataset tasks such as CIFAR-10.

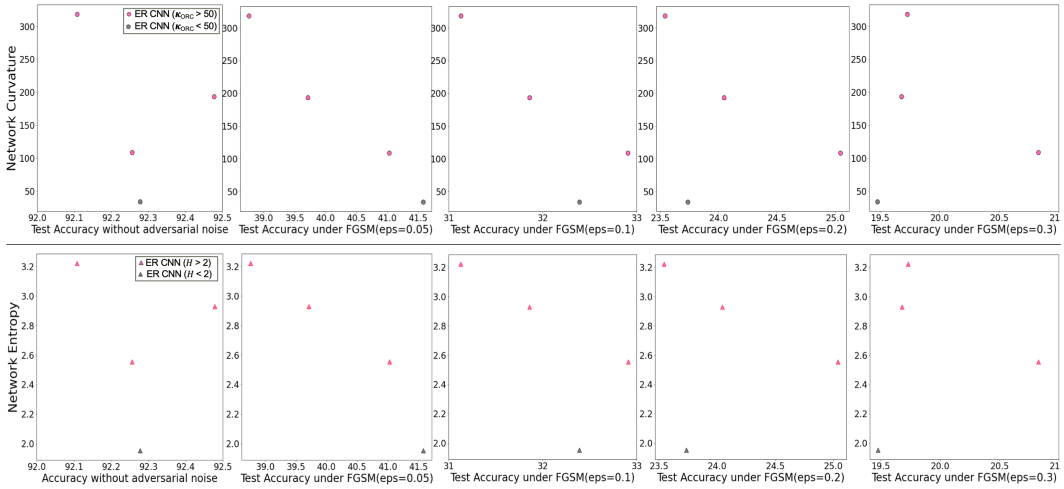


Figure 14: Graph measures *vs.* adversarial performance for ER CNNs on CIFAR-10. Each datum is a CNN. Pink-colored CNNs have higher curvature and entropy values as indicated by the legends. Top row **(Curvature *vs.* Test Accuracy under FGSM attack)**. Bottom row **(Entropy *vs.* Test Accuracy under FGSM attack)**. The curvature and entropy values of ER CNNs are greater than or almost equal to the corresponding threshold values ($\kappa_{ORC} = 50$ and $H_G = 2$). Hence there is no significant performance variance among all ER CNNs.

B Compute Resources and Wall Clock Times

Training an MLP transformed from the complete graph of WS-flex model on CIFAR-10 dataset approximately takes 7 minutes on NVIDIA TITAN RTX GPU. Each MLP was trained 5 times with random seed, consuming approximately 40 minutes for training. On the NVIDIA TITAN RTX GPU, training of all 53 MLPs on CIFAR-10 approximately took 3 days. For CIFAR-100 dataset, the 53 CNNs took approximately 5 days for training. The RandWire ANNs from WS, BA, and ER models take approximately 3 hours on the NVIDIA TITAN RTX GPU for training on both CIFAR-10 and CIFAR-100 datasets with random seed of 5. The 28 RandWire CNNs took approximately 5 days using the same TITAN RTX GPU. For Tiny ImageNet experiments on the 53 WS-flex ResNet18 ANNs, the baseline model took approximately 3 hours on TITAN RTX GPU whereas the most sparse ResNet18 took approximately 18 hours of training. Total training time for WS-flex ResNet18 models on Tiny ImageNet was approximately 22 days. All the aforementioned training times include the testing time for FGSM and PGD adversarial attacks. For tracking the experiments, visualisation of results, and hyperparameter tuning, we used the Weights and Biases (`https://wandb.ai/`) which is freely available performance visualization platform for machine learning tasks.

C Hyperparameters for Experiments

The relevant hyperparameters for our experiments are as shown in Figure 15. For the sake of procedural consistency and comparisons of results, the set of parameters other than the those mentioned in Figure 15 have been kept the same as in original experiments for *RandWire* and *Relational Graphs* by their respective authors.

Training Hyperparameters

Name of Hyperparameter	CIFAR-10			CIFAR-100		Tiny ImageNet
	WS-flex MLPs	WS-flex CNNs	RandWire CNNs	WS-flex CNNs	RandWire CNNs	WS-flex ResNet18
Nodes	64	64	32	64	32	64
Epochs	200	100	55	350	55	75
Batch Size	256	1024	128	32	128	256
Base Learning Rate	0.1	0.1	.0803	0.025	.0803	0.1
Learning Rate Policy	-- Cosine --					
Momentum	-- 0.9 --					
Weight Decay	0.0005	.01	0.00092	0.0005	0.00092	0.006
Drop out	-	-	-	FC: p=0.1	-	Conv: p=0.2 FC: p=0.5
Training Seeds	5	5	5	3	3	1

Figure 15: Training hyperparameters for all experiments.