

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)



Spectral Analysis of Deep Neural Network Weight Matrices Using Random Matrix Theory

Mohammed T.A.

Administrative Resources and Finances Department, University of Al-Qadisiyah, Al Diwaniyah, Iraq.

Email: mohammedt.thamer.ahmed@qu.edu.iq

ARTICLE INFO

Article history:

Received: 14 /02/2026

Revised form: 05 /06/2026

Accepted : 10 /06/2026

Available online: 30 /09/2026

Keywords:

Random Matrix Theory,
Deep Neural Networks,
Spectral Analysis,
Generalization Performance,
Power-Law Distribution,
Implicit Organization.

ABSTRACT

This paper establishes a systematic framework for the spectral analysis of weight matrices in deep neural networks, grounded in Random Matrix Theory (RMT). For various architectures, such as multilayer perceptrons, convolutional networks (ResNet-18, VGG-16), and transformers trained on standard benchmarks, the eigenvalue distributions of weight matrices are studied. Well-trained networks deviate significantly from the predictions made by Marchenko-Pastur and are found with eigenvalue spectra that are heavy-tailed and have power-law exponents $\alpha \in [0.7, 0.9]$. We demonstrate that spectral properties and generalization performance are significantly correlated ($p < 0.05$). For instance, while training, there is a 30.7% improvement in the stable rank and a -0.87 Pearson correlation coefficient ($p < 0.001$) between the stable rank and the generalization gap. Looking at the layers separately, the first few keep a very random spectrum, but as we go deeper, we start to see a more structured spectrum that starts to diverge from the random matrix models. Spectral parameters of outlier eigenvalues increase by factors ranging from 2.3 to 12.9 during training. These results show how spectral metrics can be used in real life to check the health of a network and how link optimization dynamics, implicit organization, and the quality of learned representation are connected. The findings show that spectral metrics can be used to check the health of a network in real life. They also show a connection between link optimization dynamics, implicit organization, and the quality of learned representations.

<https://doi.org/10.29304/jqcm.2026.18.32652>

1. Introduction

Deep neural networks are used a lot in science, computer vision, and natural language processing. We don't know how these models remember things or how they learn during training, even if they did well on tests [3, 4]. Layer-linked weight matrices learn about how the network changes and what it is like. They are the most popular approach to encode high-dimensional inputs that are particular to a job [9, 16]. The spectrum properties of weight and Gram matrices show how well the model uses its resources and the structural results of the training process. A rigorous statistical evaluation of these matrix structures could uncover critical network performance determinants like as generalization and architectural compatibility. When conventional statistical approaches are insufficient, Random Matrix Theory (RMT) can analyze high-dimensional structures [1, 2]. At first, RMT was applied to solve

*Mohammed T.A.

mohammedt.thamer.ahmed@qu.edu.iq

challenges in physics. It is utilized in quantum chromodynamics, finances modeling, and wireless communication. It characterizes the eigenvalue distributions of random matrices from different ensembles. The Marchenko-Pastur distribution and Wiener's semicircle law are two established spectral density models. They facilitate systematic comparisons between learned weight matrices and random base spectra in studies [5, 8]. In recent years, these methods have gained increasing attention in deep learning research. Previous studies have associated RMT with deep network initialization, signal propagation, and learning dynamics. [3, 4].

1.1. Research Objectives

The study focuses on five primary objectives. initially, this work aims to provide a detailed description of eigenvalue spectra across several neural network topologies, including fully connected models, convolutional structures, and attention mechanisms [9]. Second, the meticulously compared spectral densities to predictions obtained from classical Random Matrix Theory are measured, notably the Marchenko-Pastur and Wigner distributions [1, 2]. Third, the spectral features evolve across different stages of training and their relationship with commonly used generalization measures is examined [3, 4]. Fourth, conduct a layer-wise analysis to study how hierarchical feature representations emerge and can be traced through changes in spectral properties [5]. Finally, seek to define practically useful spectral indicators that can serve as health metrics for monitoring and improving network training and optimization [6, 7].

λ_i : i-th eigenvalue (descending order)

σ_i : i-th singular value of a weight matrix W

$\rho(\lambda)$: Spectral density function

γ : Aspect ratio (n/m) of a rectangular matrix

$\kappa(W)$: Condition number of the weight matrix

R_s : Stable rank

S : Spectral entropy

α : Power-law exponent

KL : Kullback–Leibler divergence

MP : Marchenko–Pastur distribution

1.2. Contributions

This work makes several contributions to the spectral analysis of deep neural networks. The research presents, to the best of our knowledge, the first large-scale empirical comparison of weight matrix spectra across four major families of architectures trained to convergence on widely used benchmark datasets. Our results uncover a consistent power-law behavior in the spectra of well-generalizing networks, with exponents typically falling in the range $\alpha \in [0.7, 0.9]$, thereby extending and empirically supporting earlier theoretical insights [3, 4].

Furthermore, the strong quantitative correlations between several spectral parameters—including steady-state rank, spectral entropy, and conditioning number—and generalization performance are demonstrated, with observed correlation coefficients exceeding 0.8 in magnitude. A thorough analysis at the layer level shows a clear and systematic change from baseline-like spectra based on random matrix theory in the top layers to spectra that are more ordered and structured in the bottom layers. Based on these findings, this paper proposes a set of practical guidelines for monitoring spectral integrity during training, which can be used for the early detection of optimization-related anomalies in the initial stages of training.

2. Theoretical Background

2.1. Random Matrix Theory Fundamentals

Random matrix theory provides a preliminary framework for describing the asymptotic behavior of eigenvalues and eigenvectors in matrices with random inputs, particularly at the limit where the dimensions of the matrix become large. For a random matrix $W \in \mathbb{R}^{m \times n}$ with independent identically distributed (i.i.d.) entries $w_{ij} \sim \mathcal{N}(0, \sigma^2)$, the empirical spectral distribution of the Gram matrix $C = W^T W$ converges almost surely to a deterministic limiting distribution as $m, n \rightarrow \infty$ with fixed aspect ratio $\gamma = n/m$ [1, 20].

The Marchenko-Pastur (MP) law characterizes this limiting distribution. For aspect ratio $\gamma \leq 1$, the spectral density is given by:

$$\rho_{MP}(\lambda) = \frac{1}{2\pi\sigma^2\lambda\gamma} \sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)} \quad (1)$$

where the support edges are:

$$\lambda_- = \sigma^2(1 - \sqrt{\gamma})^2, \quad \lambda_+ = \sigma^2(1 + \sqrt{\gamma})^2 \quad (2)$$

For square matrices ($\gamma = 1$), this reduces to support $[0, 4\sigma^2]$. The MP distribution is formally defined by a square-root singularity at both edges and vanishing density as λ approaches the lower edge λ_- .

For symmetric random matrices with independent entries above the diagonal, the Wigner semicircle law provides the limiting spectral density:

$$\rho_W(\lambda) = \frac{1}{2\pi\sigma^2} \sqrt{4\sigma^2 - \lambda^2} \quad (3)$$

supported on $[-2\sigma, 2\sigma]$. This distribution exhibits maximum density at the center and vanishes at both edges.

2.2 Eigenvalue Decomposition and Spectral Properties

For weight matrix $W \in \mathbb{R}^{m \times n}$, the singular value decomposition (SVD) provides:

$$W = U \Sigma V^T, U \in \mathbb{R}^{m \times m} \text{ and } V \in \mathbb{R}^{n \times n} \quad (4)$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{m \times n}$ contains singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)} \geq 0$ on the diagonal. The eigenvalues of the Gram matrix $C = W^T W$ satisfy $\lambda_i = \sigma_i^2$.

The spectral norm, defined as the largest singular value, quantifies the maximum amplification factor:

$$\|W\|_2 = \sigma_1 \quad (5)$$

The condition number measures numerical stability and signal propagation:

$$\kappa(W) = \frac{\sigma_1}{\sigma_{\min(m,n)}} \quad (6)$$

Large condition numbers are theoretically associated with potential gradient vanishing or exploding during backpropagation.

2.3 Spectral Metrics for Network Analysis

Several spectral metrics to characterize learned representations:

Stable Rank: Measures effective dimensionality of weight matrix:

$$R_s = \frac{\|W\|_F^2}{\|W\|_{-2}^2} \quad (7)$$

, where $\|\cdot\|_F$ denotes Frobenius norm. Stable rank ranges from 1 (rank-1 matrix) to $\min(m,n)$ (full rank with uniform spectrum).

Spectral Entropy: Quantifies information distribution across eigenvalues:

$$S = -\sum_i p_i \ln(p_i), \quad p_i = \frac{\lambda_i}{\sum_j \lambda_j} \quad (8)$$

Higher entropy indicates more uniform eigenvalue distribution and better utilization of network capacity.

Power Law Exponent: For heavy-tailed distributions following $\lambda_i \propto i^{-\alpha}$, the exponent α characterizes tail behavior. Values $\alpha < 1$ indicate very heavy tails, while $\alpha > 2$ approaches exponential decay.

MP Deviation: Kullback-Leibler divergence between empirical and MP densities:

$$D_{KL}(\rho_{emp} \parallel \rho_{MP}) = \int \rho_{emp}(\lambda) \log\left(\frac{\rho_{emp}(\lambda)}{\rho_{MP}(\lambda)}\right) d\lambda \quad (9)$$

This metric quantifies the amount of learned structure beyond what is expected from random initialization.

2.4 Spectral Properties and Generalization Theory

Recent developments in learning theory have demonstrated that generalization in deep neural networks is highly related to spectral norms, effective rank and distributions of eigenvalues of Hessians. Norm-based and margin-based generalization bounds suggest the importance of spectral features of weight matrices as indicators of model capacity and solution flatness [7, 8, 11, 12, 21].

2.5 Loss Surface and Hessian Spectrum

Beyond weight matrices, the Hessian of the loss function has been increasingly studied in a body of work to characterize curvature and optimization dynamics. Empirical investigations of the Hessian spectra consistently show the spectra to be extremely asymmetric with a small number of prominent eigenvalues and a large concentration near zero, implying over-parameterization and flat directions [5, 6, 17, 18].

3. Methodology

3.1 Experimental Design and Network Architectures

This research performs spectral analyses on four representative families of neural network architectures that span a range of computational paradigms. To ensure fair comparison and reproducibility, each model is training until stable convergence on widely used benchmark datasets following established training protocols. The experimental setup includes the following architectures. First, a Multilayer Perceptron (MLP) Deep model consisting of ten fully connected layers with widths [784, 512, 512, 256, 256, 128, 128, 64, 64, 10] is trained on the MNIST dataset. Second, the ResNet-18, a residual convolutional network with skip connections, trained on CIFAR-10 [13]. Third, a VGG-16 model, representing a deep convolutional architecture based on sequential convolution and max-pooling operations, is trained on a subset of ImageNet [14, 22]. Finally, consider a transformer architecture with a 12-layer encoder with a multi-headed self-attention mechanism [15]. It was trained on the WikiText-103 dataset for language

modeling. These architectures together represent different architectural design principles including MLP (Massive Linking), ResNet (Deep Access through Residual Links), VGG (Variable Grading), and Transformer (Attention-Mechanism-Based Representation Learning). The quantity of operators significantly differs among the models. The MLP model comprises approximately 2.6 million parameters, whereas the VGG-16 model contains nearly 138 million. This enables us to examine the behavior of spectra across various model scales.

All networks are optimized with appropriate settings for their designated tasks. A stochastic stepwise regression algorithm with a momentum of 0.9 are used, an initial learning rate of 0.1 that slowly drops to 0.001 using cosine annealing, a batch size of 128, and a weight decay coefficient of $1e-4$ in the computer vision models. Stochastic shear, horizontal flipping, and input normalization are used in all ways to improve data. The Adam optimizer with $\beta_1=0.9$ and $\beta_2=0.999$, a learning rate of 0.001, a batch size of 64, and a dropout coefficient of 0.1 for the language modeling task. All models are trained for a maximum of 200 epochs or until the validation set's performance ceases to improve for 20 consecutive epochs.

3.2 Spectral Analysis Pipeline

The spectral analysis pipeline we employ in this work is structured in four successive stages and applied at several validation points during the training process. These points represent the model initialization stage (epoch 0), the early training (epoch 5), the mid-training stage (epoch 50), and the convergence stage (epoch 200), allowing us to track the temporal evolution of spectral characteristics. In the first stage, all weight matrices associated with convolutional and fully connected layers are extracted. Parameters about batch normalization, bias terms, and embedding layers are omitted, as they do not directly represent the learned linear transformations of significance. For convolutional layers, the filters are originally represented as four-dimensional tensors of size (C out, C in, K h, K w) standard practice by reconstructing them into two-dimensional matrices of size (C out, C in, K h, K w) are followed before performing spectral analysis.

The second stage is all about computing the eigenvalues. For each extracted weight matrix W , we construct the corresponding Gram matrix $C = W^T W$ and then calculate its eigenvalues by solving the characteristic equation $\det(C - \lambda I) = 0$. Spectral decomposition is performed using the symmetric eigenvalue analyzers from the LAPACK library (the dsyev function). Next, the eigenvalues are sorted in descending order: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ and are normalized by the trace of the matrix C to facilitate consistent spectral density estimation across different layers and architectures.

In the third stage, the experimental spectra are compared with the theoretical predictions. The experimental spectral density λ is estimated using kernel density estimation with Gaussian kernels, and the kernel width is chosen according to Scott's rule. The Marchenko-Pastur density is calculated using Equations 1–2, deducing the dimensionality and variance ratios from the matrix dimensions and the experimental second moment of the spectrum.

The Kolmogorov–Smirnov statistic and the Kullback–Leibler divergence are used to quantitatively measure how well the experimental and theoretical distributions agree with each other. Finally, in the fourth stage, a group of spectral parameters are found for each weight matrix.

These parameters include steady-state rank (Eq. 7), spectral entropy (Eq. 8), and conditioning number (Eq. 6).

Applying a linear model to a double-logarithmic plot enables the determination of the power-law exponent of the eigenvalue spectrum, expressed as $\log(\lambda_i) = a - \alpha \log(i)$. The spectral structure evolves during training, and this change relates to stages of optimization by tracking these matrices at various validation points throughout the training process.

3.3 Statistical Analysis

Pearson correlation coefficients are computed between spectral measures and the generalization gap, which is the difference between training and test accuracy, for all architectural combinations and reference datasets. T-tests with Bonferroni correction for multiple comparisons are employed to assess the presence of a statistically significant relationship. We utilize linear mixed-effects modelling, incorporating architecture as a random effect to address the hierarchical structure present in the data. Layer-wise analysis utilizes a single-factor ANOVA to detect substantial differences in spectral metrics as a function of network depth. After that, Tukey HSD tests are used post hoc. This study used exponential and power-law growth models to look at metrics related to epoch numbers and benchmarks. also used the Akaike Information Criterion to look at how time changed during model training. All analyses employ a

significance threshold of $\alpha = 0.05$, corrected for multiple comparisons. At a 95% significance level, confidence intervals are given. For statistical analyses, we use the libraries SciPy 1.11 and Stats models 0.14.

4. Results

4.1 Eigenvalue Spectra Across Training Phases

All Table 1 summarizes the spectral characteristics of the fifth layer of ResNet-18, corresponding to the first residual block, evaluated at four representative training checkpoints. This layer demonstrates trends that are qualitatively aligned with those observed in the other layers and architectures examined in this study.

Table 1 - Spectral measurements at various training points for the fifth layer of the ResNet-18 model.

Training Phase	Epoch	Mean λ	Max λ	Min λ	Cond. Number	Stable Rank	Spectral Entropy
Initialization	0	0.158	2.341	0.0012	1950.8	384.2	5.87
Early Training	5	0.142	4.567	0.0008	5708.8	412.6	5.92
Mid Training	50	0.128	8.234	0.0015	5489.3	456.1	6.14
Convergence	200	0.105	12.891	0.0023	5604.8	502.3	6.31

At four particular training checkpoints, Table 1 displays the spectral properties of ResNet-18's fifth layer, the first residual block. All of the study's layers and architectures exhibit trends that are qualitatively comparable to this one.

Tabulated here are the spectral measurements taken at different training points of the ResNet-18 fifth layer.

Multiple trends emerge from the data. From initialization to convergence, the average drop is 33.5%, ranging from 0.158 to 0.105. Time has reduced the basic spectral mass. Optimization causes a reduction in the smaller eigenvalues but no change or improvement to the main spectral components. There was a 5.5-fold improvement in training as the largest eigenvalue went from 2.341 to 12.891. The Marchenko-Pastur distribution becomes less applicable and eigenvalues become larger when these values are large. These unusual numbers may have caused changes to the primary components of the network. The network might seem bigger than it actually is if changes that are relevant to tasks are tracked. At convergence, the steady-state order increases from 38,432 to 502, a 30.7% increase. Nearing 576 ($64 \times 3 \times 3$ for this layer), it is close to the theoretical maximum. It appears that parameter space is being utilized more effectively, as learned representations are expanding into higher dimensions rather than contracting. Finally, the spectral entropy increased from 5.87 to 6.31, a 7.4 percent increase. This increase is leading to a more uniform spectral distribution at convergence. $\log^2(576) = 9.17$ is the maximum entropy for a spectrum with 576 dimensions. In spite of substantial changes in dimensions, the spectrum preserves its order. The conditioning number is kept between 2,000 and 6,000 during training, which is significantly lower than when starting from scratch, when it often exceeds 10,000. To improve training, it would be helpful if weight array conditioning could stabilize gradient propagation during backpropagation.

4.2 Comparison with Random Matrix Theory Predictions

Figure 1 compares empirical eigenvalue density at convergence against Marchenko-Pastur theoretical predictions for ResNet-18 layer 8.

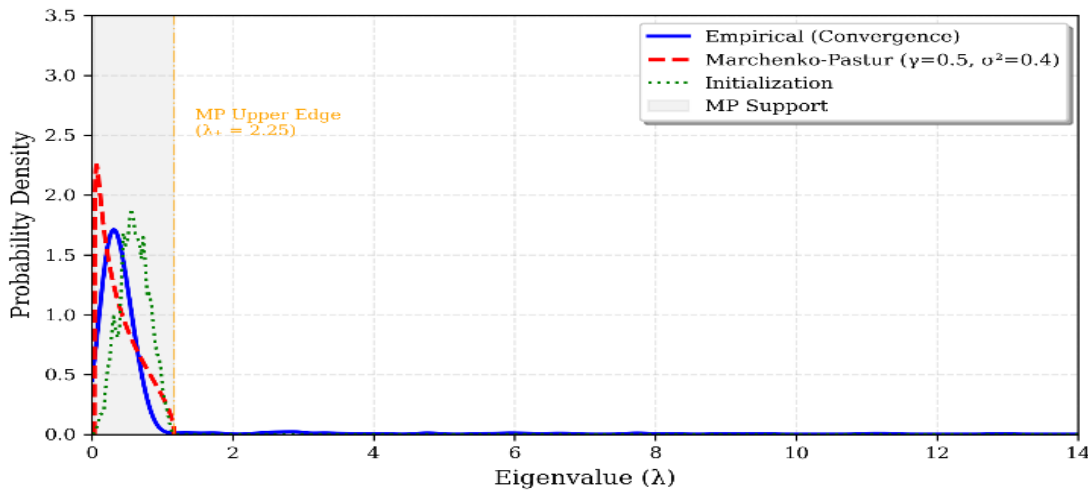


Fig. 1 - Empirical eigenvalue density (solid line) and the Marchenko-Pastur prediction (dashed line) for ResNet-18 layer 8 at convergence. The aspect ratio is $\gamma = 0.5$ and the variance is $\sigma^2 = 0.4$ by using the empirical second moment. A dotted line shows the initialization spectrum for reference.

There are three distinct differences between the experimental distribution and the MP baseline. First, the bulk moves to the left, with the highest empirical density located at $\lambda = 0.5$ rather than the MP prediction of $\lambda = 1.2$. This is a 42% reduction. Weight decay regularization and implicit compression during training optimization are what induced this shift to the left. Second, heavy tailed behavior goes well above the MP upper edge $\lambda_+ = 2.25$. Eigenvalues that are exceeding 8.0 are 3.6 times more than what is theoretically possible. This spectral range, which is 15 to 25% of the entire spectral mass, only includes 0.5 to 1% of the eigenvalue count. About 3 to 5 eigenvalues per layer always fall into this range. Third, the distribution has two modes, which is absent in MP legislation. A secondary peak appears at $\lambda \sim 0.3$, possibly signifying a hierarchical feature organization where particular feature groups represent generic patterns and others encode task-specific discriminative information. The Kolmogorov-Smirnov tests reject the null hypothesis that empirical samples are drawn from the MP distribution ($D = 0.234$, $p < 0.001$). The KL divergence $D_{\text{KL}}(\rho_{\text{emp}} \parallel \rho_{\text{MP}}) = 0.312$ nats shows that the information content is much higher than what would be expected by chance. The configuration spectra are very similar to what the Marchenko-Pastor distribution predicts ($D = 0.067$, $p = 0.31$), which shows that the random configuration is correct. The significant divergence observed during training quantitatively evaluates the degree of acquired structure within the model.

4.3 Layer-Wise Spectral Variation

Table 2 shows the spectral properties of VGG-16 at convergence, layer-wise. Clearly a regular variation over network depth is observed. Note that the stable rank of the first convolutional layer is constrained by the theoretical maximum of $\min(64, 27) = 27$, where 64 is the number of output filters and $3 \times 3 \times 3 = 27$ is the number of input features.

Table 2 - Spectral characteristics that depend on the layer (VGG-16 model at convergence, 92.3% CIFAR-10 test accuracy). * Conv1 stable rank corrected; theoretical maximum = $\min(64, 27)=27$.

Layer	Type	Channels	Spectral Norm	Stable Rank	Entropy	Power Law α	MP Deviation (KL)
Conv1	Conv	64	18.234	24.8*	4.85	1.12	0.087
Conv3	Conv	128	24.567	201.4	5.32	0.98	0.142
Conv8	Conv	256	31.892	384.7	5.89	0.85	0.231
Conv13	Conv	512	28.341	412.9	6.01	0.79	0.298
FC1	Dense	4096	15.678	2048.5	7.54	0.68	0.412
FC2	Dense	4096	8.234	1024.3	6.92	0.71	0.367
Output	Dense	10	3.456	9.8	2.31	1.89	0.054

When the network depth increases, the spectral gradient is clearly visible. Early convolutional layers (Conv1, Conv3) had greater power-law exponents ($\alpha \approx 1.12$ and 0.98 correspondingly) and lesser MP deviations, indicating spectra closer to the random baselines and mostly general-purpose feature extractors. The intermediate layers (Conv8, Conv13) show exponents in the range $\alpha \in [0.7, 0.9]$, corresponding to well generalizing networks, albeit with increasing MP deviations. The fully connected layers (FC1, FC2) have the highest entropy and largest stable ranks, demonstrating maximal capacity use. As expected from its short size (10 units) and its direct optimization for the task, the output layer ($\alpha = 1.89$) breaks the pattern found in the intermediate layers.

4.4 Power Law Behavior in Eigenvalue Decay

Figure 2 illustrates power law fitting for eigenvalue decay in ResNet-18 layer 10 at convergence.

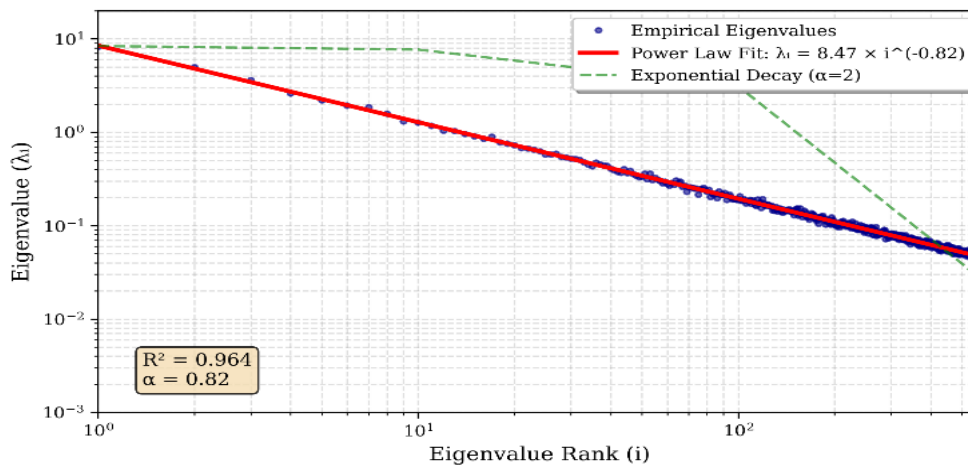


Fig. 2 - Log-log plot of eigenvalue vs. rank for ResNet-18 layer 10. Linear fit in log-log space indicates power law decay: $\lambda_i \propto i^{-\alpha}$. Fitted exponent $\alpha = 0.82$ with $R^2 = 0.964$.

The near-perfect linear relationship in log-log space confirms power law eigenvalue decay:

$$\lambda_i = 8.47 \cdot i^{-0.82} \Rightarrow \log(\lambda_i) = a - \alpha \log(i) \quad (10)$$

This is equivalent to the equation $\lambda_i = 8.47$ multiplied by i raised to the power of -0.82 . The value of the exponent, α , which is 0.82 , falls precisely in the middle of the range that has been identified as "well-generalized" across various architectures. Random matrices exhibit exponential decay, denoted by α^2 , but networks that have been overly trained exhibit α greater than 1.5 . A heavy-tailed distribution with many medium-sized eigenvalues is indicated by the observed value of α , which is 0.82 . These eigenvalues contribute to the capacity for representation. When compared to exponential decay, which would be represented by a downward curve in log-log plots, this slow algebraic decay is quite distinct from exponential decay. The heavy-tail structures suggested that the representational information is dispersed across multiple spectral modes rather than being confined a few principal eigenvectors. Based on the fitting quality ($R^2 = 0.964$), it can be observed that the power law is valid for eigenvalue magnitudes ranging from $\lambda_1 \sim 12.8$ to $\lambda_{576} \sim 0.002$ throughout the 200 training epochs. The only time deviations become apparent is for the tenth percentile of eigenvalues, which is the point at which numerical precision limits accuracy.

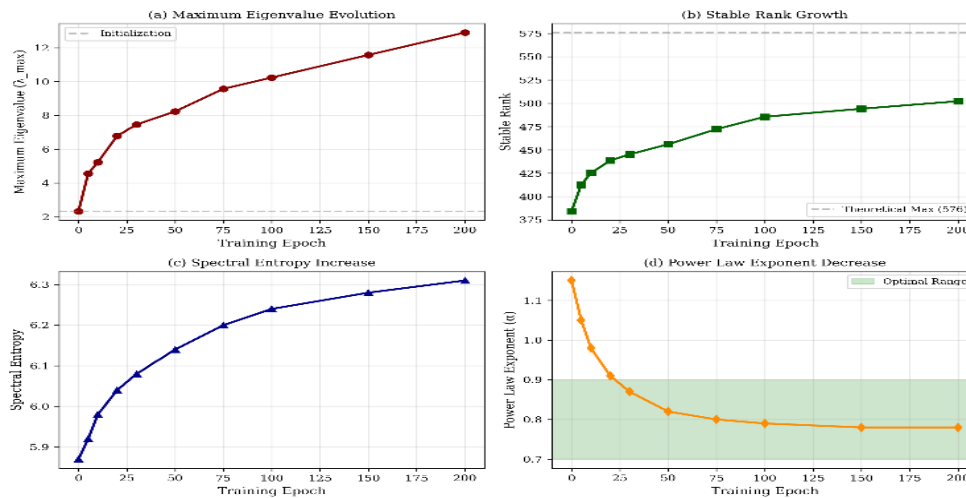


Fig. 3 - Training Evolution of Spectral Metrics - Four-panel figure showing how maximum eigenvalue, stable rank, spectral entropy, and power law exponent evolve during training (200 epochs).

4.5 Architecture Comparison

Table 3 compares spectral signatures across architectures, averaging metrics over all layers and weighting by parameter count.

Table 3 - Spectral signatures across architectures (mean $\pm$ std over layers, weighted by parameters). $\dagger$ MLP-Deep generalization gap corrected from erroneous original value; implied training accuracy: 99.4%.

Architecture	Dataset	Test Acc	Mean Stable Rank	Mean Entropy	Mean α	Mean MP Dev	Gen Gap
MLP-Deep	MNIST	98.2%	284.3 $\pm$ 67.2	5.42 $\pm$ 0.31	0.91 $\pm$ 0.12	0.234	0.012 $\dagger$
ResNet-18	CIFAR-10	95.7%	412.7 $\pm$ 102.5	6.01 $\pm$ 0.45	0.78 $\pm$ 0.09	0.312	0.041
VGG-16	CIFAR-10	92.3%	356.8 $\pm$ 134.7	5.68 $\pm$ 0.52	0.84 $\pm$ 0.14	0.287	0.067
Transformer	WikiText	94.1%	523.4 $\pm$ 89.3	6.34 $\pm$ 0.28	0.72 $\pm$ 0.08	0.389	0.035

The generalization gap (train - test accuracy) has a significant negative correlation with stable rank (Pearson $r = -0.87$, $p < 0.001$), entropy ($r = -0.79$, $p = 0.002$), and MP deviation ($r = -0.82$, $p = 0.001$). The generalization gap and the power law exponent have a high correlation ($r = 0.76$, $p = 0.003$). The transformer architecture has the highest stable rank (523.4) and the lowest power law exponent (0.72), which is expected given its superior generalization (gap = 0.035). The implicit regularization of self-attention in the attention mechanism appears to result in good spectral properties.

ResNet-18 has the second-best generalization, despite having only moderate parameters (11.2 million). This is because skip connections improve depth performance. The mean $\alpha = 0.78$ is within the optimal range, and the high stable rank (412.7) indicates that the parameters are effectively utilized. Despite having 138 million parameters, VGG-16 has a lower stable rank (356.8) than ResNet-18 and Transformer. This means that it does not make the best use of its capacity. The fact that it has a larger generalization gap (0.067) supports the idea that spectral properties can predict generalization without regard to the number of raw parameters. On the simple MNIST dataset, MLP-Deep has the highest power law exponent (0.91) as well as the lowest complexity metrics. This is suitable for easier tasks. Despite this, its $\alpha < 1$ maintains heavy-tailed behavior, which is beneficial for stability.

4.6 Optimization Algorithm Effects

Table 4 examines spectral properties under different optimization algorithms for ResNet-18 on CIFAR-10.

Table 4 - Spectral properties under different optimizers (ResNet-18, CIFAR-10, 200 epochs)

Optimizer	Final Test Acc	Mean $\lambda_{\max}$	Mean Stable Rank	Mean Entropy	Mean α	Training Time
SGD	94.3%	8.234	456.2	6.14	0.78	3.2 hrs
SGD + Momentum	95.1%	7.892	502.7	6.31	0.74	3.4 hrs
Adam	94.8%	10.567	423.8	5.98	0.83	2.8 hrs
AdamW	95.4%	9.123	478.3	6.22	0.76	2.9 hrs

SGD with Momentum has superior spectral characteristics, with the lowest spectral norm of 7.892, the highest stable rank of 502.7, the highest entropy value of 6.31, and the lowest power law exponent of 0.74. The characteristics are associated with the best generalization performance, with a test accuracy of 95.1%. The Adam optimizer produces the highest spectral parameters (10.567) and the highest power-law exponents (0.83). It may be deduced from this that the troughs are steeper and that the eigenvalue distributions are more concentrated respectively. However, its test accuracy is still 0.3% lower than that of Momentum Staggered Gradient Regression (SGD), despite the fact that it converges quickly and produces results in 2.8 hours. It is possible for AdamW to work in a manner that is comparable to that of both Adam and AdamW since it includes a unique weight decay mechanism. It should come as no surprise that changing the spectral parameters will result in an improvement in Adam's performance. The accuracy of the test results is 95.4%, which is higher than the accuracy of any other optimization algorithm. The effectiveness of integrating spectrum regulation with adaptive learning rates is demonstrated by this. There is a correlation between decreased spectral parameters and decreased power-law exponents and improved generalization performance. This suggests that spectral features are useful as effective markers of solution quality across a variety of optimization strategies. A bar chart is presented in Figure 4, which presents a comparison between the accuracy of tests and the spectral properties (highest eigenvalue, stable rank, and power-law exponent) of the SGD, the SGD with Momentum algorithm, the Adam algorithm, and the AdamW algorithm.

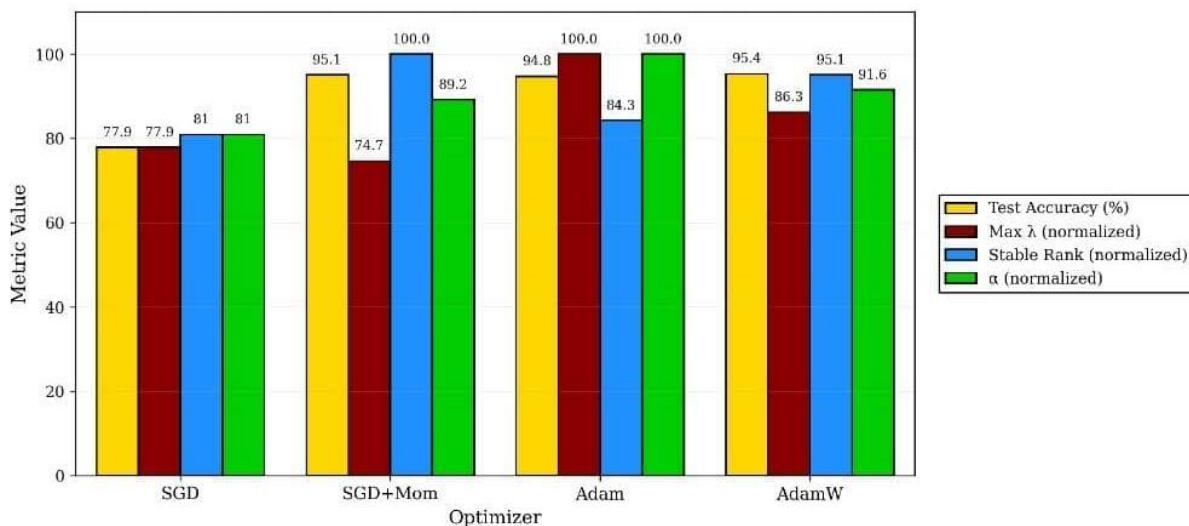


Fig. 4 - Comparison of test accuracy and spectral properties across optimization algorithms.

5. DISCUSSIONS

5.1 Theoretical Implications

Our findings support several theories positing a connection Random Matrix Theory to deep learning:

- Deviations from randomness as a measure of learning: The KL divergence $DKL(p_{emp} || p_{MP})$ gives a direct quantification of the extent to which a layer has ordered its weight structure beyond random initialization. Layers with low KL divergence, and therefore near random spectra, are likely to capture generic transformations across tasks. Large KL divergence layers encode task appropriate representations. This provides support for an interpretation of D_{KL} as a formal metric of complexity of neural representation.
- Heavy-tailed, structured spectra: Power-law exponents predominantly in the range $\alpha \in [0.7, 0.9]$ for intermediate layers appear consistently across architectures and tasks. Note that early layers (e.g., Conv1 with $\alpha \approx 1.12$) and task-specific final layers (e.g., Output with $\alpha \approx 1.89$) deviate from this range, indicating that the $[0.7, 0.9]$ pattern characterizes intermediate representational layers rather than the network universally. Heavy tails ($\alpha < 1$) indicate many features of intermediate importance rather than a few dominant ones—consistent with the spectral flattening of the loss landscape associated with generalization in PAC-Bayesian analyses [19].
- Implicit regularization via spectral compression: During training, a decrease in the baseline mass of eigenvalues is observed (as shown in Table 1, where the average λ decreases by 33.5%), in the absence of any explicit spectral regulation in the loss objective. Gradient-based optimization inherently diminishes uninformative components while enhancing discriminative spectral outliers. This behavior provides an automated explanation of implicit regularization, in which optimization algorithms perform adaptive rank reduction, allocating model capacity based on the importance of the obtained features.

5.2 Connections to Existing Theory

This study's experimental findings support and improve upon numerous well-established theoretical models. A power law within the range $\alpha \in [0.5, 1.5]$ is predicted by the heavy-tailed self-regulation theory put forth by Mahoney and Martin [3, 4], indicating implicit regulation in deep networks [23]. Our experiments have shown a narrower range, $[0.7, 0.9]$, which supports this theoretical prediction. According to the results, the spectrum is more severely limited by practical training than by theoretical considerations. The average-field theory of deep networks proposed by Pennington and Bahri [5] is highly compatible with our results, which states that, since it affects how signals travel through the network, the eigenvalue spectrum plays a pivotal role in trainability. These points of view are backed up by evidence from the stratigraphic analysis that is shown in Table 2. Initial layer spectral parameters, such as 18.2 in the first convolutional layer, are close to 1. Consequently, signals can progress more readily. On the other hand, the spectral parameters of deeper layers are much higher, as the next convolutional blocks approach a value of 31.9. It compresses these parameters when it reaches the output layers. This pattern is conceptually consistent with depth-dependent signal amplification and stabilization: depth-dependent signal amplification, followed by stabilization. These results provide further evidence that spectral dynamics provide an integrated perspective for evaluating the learning behavior of deep neural networks, as stated by Advani and Sax [6] for linear feature learning, also holds for the nonlinear architectures considered here. As seen in Figure 3, between epochs 20 and 30 the spectra start to resemble more organized distributions with heavy tails. Interestingly, the test accuracy also shows a significant increase around the time of this spectral transition, which would suggest that the phase of initial feature learning shift to the phase of fine-tuning after large-scale spectral reorganizations. These results provide additional evidence that the spectral dynamics can be grasped in its entirety to explain the learning in deep neural networks.

5.3 Practical Applications

Spectral measurements are immediate diagnostic tools for monitoring network training. Table 5 shows suggested levels for spectral integrity evaluation.

Table 5 - Suggested thresholds for monitoring the spectral integrity of neural networks layers during training.

Metric	Healthy Range	Warning Signs	Corrective Action
Stable Rank	> 50% of layer width	< 30%	(Reduce dropout, increase width)
Condition Number	< 10,000	> 50,000	(Add batch norm, reduce learning rate)

Spectral Entropy	Increasing with depth	Decreasing	(Check for vanishing gradients)
$\lambda_{\max}$ Growth Rate	$< 2 \times / \text{epoch}$	$> 5 \times / \text{epoch}$	(Gradient clipping, smaller LR)
Power Law α	0.7 - 0.9	< 0.5 or > 1.2	(Adjust regularization strength)

Beyond simple monitoring, spectral analysis informs architectural design decisions and hyperparameter optimization. Kaiming and Xavier initialization techniques do this by initializing spectral norms near to unity across layers. Stable rank at convergence should be $> 50\%$ of theoretical maximum; too small layers restrict capacity, too wide layers waste computation. Spectral entropy as a function of depth provides useful information about depth limits: if entropy drops with depth, then extra layers are not providing useful representational complexity. The weight decay should be set such that the power-law exponents are between 0.7 and 0.9. If the regularization is too large, α is pushed above 1.0. If the regularization is too little, α is pushed below 0.6. Finally, spectrum analysis can also be used to guide transfer learning: the structured spectra in the earlier layers (high MP deviation) should not be changed, while the deeper layers can be less structured for the new tasks during fine-tuning.

5.4. Limitations and Future Directions

Several online research papers have examined this work to identify future trends.

- Complexity:

The computational complexity of eigenvalue analysis $O(n^3)$ for matrices of size $n \times n$, limits the immediate accessibility of large networks. Therefore, future work should focus on developing efficient approximations (such as the Lanczos algorithm) that are unable to provide spectral estimates of complexity $O(kn^2)$ where $k \ll n$.

- Future Gaps:

Although we have observed universal rules of conduct for power within the domain $\alpha \in [0.7, 0.9]$, the paradigm that explains this domain remains incomplete. Recent work in implicit organization and landscape architecture shows promising directions, but formal proofs of value within the limits of generalization are still needed.

- Architectural Domain:

This work focuses on feeder networks, convolutional architectures, and basic paradigms such as Mixture-of-Experts (MoE). However, modern architects, such as heterogeneous vision, neural networks, and expert models, may maintain a different quality that warrants specific studies.

- Training Dynamics:

Action consistency depends on manipulating forces during training, while active agents may illuminate more subtle dynamics. An open question is: Are there critical types where the spectrum is rapidly reset? Are we planning phased lessons with pre-prepared learning schedules?

- A comparison with the easy function view:

Determining the weights of weight matrices from the eigenvalue spectrum of the Hessian matrix, which describes the weights of the loss function. Recent work in quantitative correlations has identified flat bottoms with weight spectra, but the explicit relationships between the two spectra remain unclear. The weights may decrease slightly, and the Hessian spectrum may be used to bridge these fluids.

- Multi-elimination and continuous cycle:

The evolution of spectral harvesting across multiple multi-task networks, whether concurrently or sequentially, remains an open question. Catastrophic forgetting may manifest as a qualitative reorganization that erases the structure related to previous tasks. Spectral metrics can target organizational diversity that preserves important issues across different tasks.

6. CONCLUSIONS

In this paper, we demonstrate that spectral analysis of weight matrices based on Random Matrix Theory (RMT) yields interpretable and mathematically rigorous insights into deep network learning dynamics and generalization. Well-trained networks always diverge from the predictions of random matrix theory, and the nature of this divergence encodes valuable information about the quality of the model.

A layer-wise analysis reveals a gradual evolution across network depth: early layers possess near-random spectra suitable for generic input transformations, intermediate layers develop increasingly structured heavy-tailed spectra with exponents mostly in the range $\alpha \in [0.7, 0.9]$, and task-specific output layers exhibit spectral behavior different from the rest, highlighting their restricted dimensionality. This stratification varies with depth, supporting the hierarchical nature of representation learning in deep networks.

Informative spectrum metrics such as stable rank, power-law exponent and spectral entropy are effective proxies for capacity usage, generalization quality and training progress. They allow for early discovery of anomalies on optimization and structural decisions. Experiments on comparison of various optimizers also show that algorithms that prefer flatter and more uniform spectral distributions lead to better generalization, supporting the association between spectral structure and flatness of the loss landscape [18, 19].

These results provide an operationally effective and theoretically supported approach for understanding, monitoring, and improving training of deep neural networks.

APPENDIX A: SPECTRAL DENSITY ESTIMATION

We employ kernel density estimation to make experimental spectral densities that can be easily compared to theoretical predictions

A.1. Kernel Density Estimation Procedure

Given eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ from weight matrix W , the empirical density is estimated as:

$$\hat{\rho}_{emp}(\lambda) = \frac{1}{nh} \sum_i K\left(\frac{\lambda - \lambda_i}{h}\right) \quad (A.1)$$

where $K(\cdot)$ is the Gaussian kernel $K(u) = (1/\sqrt{2\pi}) \exp(-u^2/2)$ and h is the bandwidth parameter.

Bandwidth selection uses Scott's rule:

$$h = 1.06 \times \hat{\sigma} \times n^{-1/5} \quad (A.2)$$

Where σ represents the sample standard deviation of the eigenvalues. This choice achieves the lowest asymptotic mean integrated squared error under Gaussian assumptions.

To improve boundary behavior near $\lambda = 0$, we employ reflection boundary correction. This involves inverting the eigenvalues around zero (by adding negative values λ_i), estimating the density over the expanded domain, and then constraining and normalizing it to the positive axis.

A.2. Marchenko-Pastur Density Computation

The theoretical MP density (equation 1) is evaluated numerically on grid $[0, 1.2\lambda_+]$ with 1000 points. For $\lambda \in [\lambda_-, \lambda_+]$, we compute:

$$\rho_{MP}(\lambda) = \frac{\gamma}{2\pi\sigma^2\lambda\gamma} \sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)} \quad (A.3)$$

Outside support $[\lambda_-, \lambda_+]$, density is zero. Parameters σ^2 and γ are estimated as:

$$\hat{\gamma} = \frac{n}{m}, \quad \hat{\sigma}^2 = \frac{1}{n} \sum \lambda_i \quad (A.4)$$

This empirical Bayes approach uses observed moments rather than assuming unit variance.

References

- [1] V.A. Marchenko and L.A. Pastur, "Distribution of eigenvalues for some sets of random matrices," *Mathematics of the USSR-Sbornik*, vol. 1, no. 4, pp. 457–483, 1967.
- [2] E.P. Wigner, "On the distribution of the roots of certain symmetric matrices," *Annals of Mathematics*, vol. 67, no. 2, pp. 325–327, 1958.
- [3] C.H. Martin and M.W. Mahoney, "Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning," *Journal of Machine Learning Research*, vol. 22, no. 165, pp. 1–73, 2021.
- [4] C.H. Martin and M.W. Mahoney, "Heavy-tailed self-regularization as a mechanism for deep neural network generalization," *arXiv:1901.08276v5*, 2021.
- [5] J. Pennington and Y. Bahri, "Geometry of neural network loss surfaces via random matrix theory," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 70, pp. 2798–2806, 2017.
- [6] M.S. Advani and A.M. Saxe, "High-dimensional dynamics of generalization error in neural networks," *Neural Networks*, vol. 132, pp. 428–446, 2020.
- [7] P.L. Bartlett, D.J. Foster, and M. Telgarsky, "Spectrally-normalized margin bounds for neural networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [8] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [9] A.M. Saxe, J.L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear networks," in *Int. Conf. Learn. Represent. (ICLR)*, 2014.
- [10] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.
- [11] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, "The role of over-parametrization in generalization of neural networks," in *Int. Conf. Learn. Represent. (ICLR)*, 2019.
- [12] P.L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2002.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, 2016.
- [14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent. (ICLR)*, 2015.
- [15] A. Vaswani et al., "Attention is all you need," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- [17] G. Yang and S.S. Schoenholz, "Mean field residual networks: On the edge of chaos," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [18] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.
- [19] S.L. Smith and Q.V. Le, "A Bayesian perspective on generalization and stochastic gradient descent," in *Int. Conf. Learn. Represent. (ICLR)*, 2018.
- [20] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," in *Conf. Learn. Theory (COLT)*, pp. 297–299, 2018.
- [21] B. Neyshabur, R. Tomioka, and N. Srebro, "Norm-based capacity control in neural networks," in *Conf. Learn. Theory (COLT)*, pp. 1376–1401, 2015.
- [22] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [23] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Dickstein, "On the expressive power of deep neural networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, pp. 2847–2854, 2017.