Random matrix analysis of deep neural network weight matrices

Matthias Thamm,[*] Max Staats,[*] and Bernd Rosenow

Institut für Theoretische Physik, Universität Leipzig, Brüderstrasse 16, 04103 Leipzig, Germany

Neural networks have been used successfully in a variety of fields, which has led to a great deal of interest in developing a theoretical understanding of how they store the information needed to perform a particular task. We study the weight matrices of trained deep neural networks using methods from random matrix theory (RMT) and show that the statistics of most of the singular values follow universal RMT predictions. This suggests that they are random and do not contain system specific information, which we investigate further by comparing the statistics of eigenvector entries to the universal Porter-Thomas distribution. We find that for most eigenvectors the hypothesis of randomness cannot be rejected, and that only eigenvectors belonging to the largest singular values deviate from the RMT prediction, indicating that they may encode learned information. In addition, a comparison with RMT predictions also allows to distinguish networks trained in different learning regimes—from lazy to rich learning. We analyze the spectral distribution of the large singular values using the Hill estimator and find that the distribution cannot in general be characterized by a tail index, i.e., is not of power-law type.

I. INTRODUCTION

The application of deep neural networks (DNNs) to a wide range of problems has been tremendously successful in recent years [1–6]. Beyond the classification of images [3], DNNs have been able to learn games beyond human capabilities [7], and have made significant progress in solving hard problems like protein folding [8]. Given these achievements, it is not surprising that neural networks are being applied to a variety of physics problems as well [5]. Applications range from quantum state tomography [9,10], locating classical [11], and quantum phase transitions [12–15], fluid turbulence [16,17], troposphere temperature prediction [18] to classifying events in the Large Hadron Collider in search of physics beyond the standard model [19,20].

Interestingly, successful neural networks are often highly overparametrized [21–32], such that one might have doubts regarding their capability to generalize beyond the training data set. Furthermore, DNNs can effortlessly memorize large amounts of random training data [33,34], but still generalize well if there is a rule to be learned. According to the traditional concept of a "bias-variance tradeoff" [35], one would indeed expect these networks to overfit and fail in predicting unseen data. However, it turns out that their generalization capability exhibits a so-called double descent behavior [36–38] as a function of the number of network parameters, such that they perform well in the highly overparameterized limit. The apparent contradiction between overparameterization and good generalization performance is further resolved by emerging evidence that ultrawide neural networks are inherently biased toward simple functions [6,39–41]. For the analysis of such highly overparameterized networks we use random ma-

trix theory (RMT) [42–46] as a zero information hypothesis, where deviations from RMT are indications of system specific information. This approach has already proven to be useful for many other problems with inherent randomness such as the analysis of nuclei spectra [43,45–47], the investigation of stock market correlations [48–54], and for the analysis of biological networks [55,56].

Previously RMT has been applied to neural networks for estimating the asymptotic performance of single-layer networks [57,58], and for analyzing the generalization dynamics of linear networks [59]. Outliers and the random part of preactivation covariance matrices were studied in Ref. [60]. Other work focused on the spectra of Jacobians at initialization [61] and on the eigenvalue distribution of the Hessian of the loss matrix [62,63]. The spectral evolution of weight matrices during training was analyzed in Ref. [64], with the main results that for smaller or older networks there is a scale in the singular value spectrum separating signal from noise, while in state of the art DNNs the spectral density exhibits a power-law tail, reminiscent of the spectrum of heavy-tailed random matrices [65,66]. These results were subsequently used to assess the quality of pretrained DNNs [67] by computing spectral norms of weight matrices and by assessing the exponent of a power-law fit to the tail of the singular value spectrum.

Here, we employ a variety of RMT tools to demonstrate that the weight matrices of deep and overparameterized neural networks are predominantly random. Specifically, we compare the singular values of several DNNs with the Wigner nearest-neighbor spacing distribution and find that the bulk of the singular values follows the RMT prediction, even when the networks are fully trained. This finding is also corroborated by the analysis of the number variance of weight matrix singular value spectra. The advantage of analyzing these RMT predictions for spectral correlations is that they are universal properties of random matrices, in contrast to the

---

distribution of singular values which depends sensitively on the specific type of random matrix. The idea that a large fraction of singular values does not encode information is tested by comparing the distribution of eigenvector components to the universal Porter-Thomas distribution. Only for a small fraction of eigenvectors with large singular values do we find significant deviations from the Porter-Thomas distribution, indicating that learned information is encoded in them.

In principle, it is also possible that information is stored in weights that match RMT predictions: For example, for a teacher-student setup with random teacher, one cannot locate information by an RMT analysis even for a perfect student of the same architecture—thus, the absence of deviations from RMT predictions is no guarantee that no information was learned. This example shows that the success of the zero-information hypothesis approach depends on the structure of the data set on which the DNN is trained. We show here, however, that the RMT analysis is particularly fruitful for image classification problems where networks trained on datasets such as MNIST, CIFAR, or ImageNet learn a rule in the images that can be encoded in a low-rank matrix such that the trained weights consist of a random bulk from initialization and a low-rank contribution.

In addition, we train networks in different learning regimes from lazy networks for which weights barely change during training to rich networks where the final weights significantly differ from the initial ones [68–71]. We find that networks trained in the lazy regime follow RMT predictions, different from networks trained in the rich and in the intermediate regime. Thus, the form of the weight spectrum and the comparison of singular vector entries to the Porter-Thomas distribution allows to distinguish between the learning regimes where the best generalization performance is found in between both extremes.

Furthermore, we apply the Hill estimator to study the question of whether large singular values of weight matrices can be described by a power-law tail of the distribution [64,67]. In contrast to previous results based on fitting the probability density function to a power law, both the Hill estimator analysis and p values for the significance of power-law fits reveal that the singular value distribution for most of the considered cases cannot be characterized by a tail exponent, implying that the tail is generally not described by a power law.

II. MAIN RESULTS

In this section, we illustrate our main results using a feedforward DNN with three hidden layers, each containing 512 neurons denoted as MLP512. In the subsequent sections, we provide more details and show that our results are valid more generally for a variety of network architectures. The code needed to reproduce all results is open source and has been made available online [72].

We train MLP512 networks on the CIFAR-10 dataset [30], which consists of images $x^{(k)}$ (3072 pixels each) and corresponding labels $y^{(k)}$, which categorize the images into 10 different classes. Therefore, in total the network has five layers with sizes $\boldsymbol{n} = [3072, 512, 512, 512, 10]$.

Except for the input layer, each layer i has an associated weight matrix W_i, a bias vector $\boldsymbol{b}_i$, and an activation function $f_i(\boldsymbol{x})$ which we choose to be a rectified linear unit (ReLU) $f(x) = \max(x, 0)$ for the hidden layers and softmax $f_{\text{out}}(\boldsymbol{x}) = \exp \boldsymbol{x} / \sum_i \exp x_i$ for the output layer. The activation of the input layer $\boldsymbol{a}_0$ is defined as the input image presented to the network

$$\boldsymbol{a}_0^{(k)} = \boldsymbol{x}^{(k)}, \tag{1}$$

and then the network propagates the activations through each layer such that the activation in layer i is given by

$$\boldsymbol{a}_i^{(k)} = f_i\big(W_i \boldsymbol{a}_{i-1}^{(k)} + \boldsymbol{b}_i\big). \tag{2}$$

Here, the W_i are $n_i \times m_i$ weight matrices, with n_i denoting the width of layer i and m_i denoting the width of layer $i-1$. The output of the network is defined as the activation of the last layer, using the largest entry of $\boldsymbol{a}_{\text{out}}^{(k)}$ as prediction for the class of the input image $\boldsymbol{x}^{(k)}$.

Before training, weights are initialized using the distribution Glorot uniform [73] while the biases are set to zero. Hence, the initial weight matrices are random matrices and obey RMT predictions (we note that Glorot initialization and random Gaussian initialization do not differ in the singular value distribution and other statistical properties). The network's weights and biases are trained by minimizing the cross-entropy cost function

$$l(\boldsymbol{W}, \boldsymbol{b}) = -\frac{1}{N} \sum_{k=1}^{N} \boldsymbol{y}^{(k)} \cdot \ln\big(\boldsymbol{a}_{\text{out}}^{(k)}\big) \tag{3}$$

on the training dataset using gradient descent on mini-batches of size 32. Further details of the training procedure are described in Appendix A.

By analyzing the singular values of weight matrices in trained DNNs using methods of RMT, in the following we argue that a large fraction of the weights remains random even after training, while the learned information is encoded in relatively few large singular values and corresponding vectors. We focus on the second fully connected layer of MLP512, and obtain the singular values ν via the singular value decomposition $W = U \operatorname{diag}(\nu) V^T$ of the corresponding weight matrix W. Here, U and V are orthogonal matrices, and $\operatorname{diag}(\nu)$ is a diagonal matrix containing the singular values which we assume to be rank-ordered in the following.

We compare the trained weight matrices to their initialized states obtained by drawing entries of an $n \times m$ matrix independently and identically from a Glorot uniform distribution [73] with variance σ^2. The singular values follow the Marchenko-Pastur distribution [74–76],

$$P(\nu) = \begin{cases} \frac{n/m}{\pi \tilde{\sigma}^2 \nu} \sqrt{\big(\nu_{\max}^2 - \nu^2\big)\big(\nu^2 - \nu_{\min}^2\big)} & \nu \in [\nu_{\min}, \nu_{\max}], \\ 0 & \text{else}, \end{cases} \tag{4}$$

where $\nu_{\max}^{\min} = \tilde{\sigma}(1 \pm \sqrt{m/n})$ and $\tilde{\sigma} = \sigma \sqrt{n}$. We assume without loss of generality that $m \leqslant n$. In the case of the second hidden layer weights, we have $n = m = 512$. While the distribution Eq. (4) is a parameter free description of the untrained network, the trained network deviates from the Marchenko-Pastur law (see also the discussion in Ref. [64]). In the absence of a microscopic theory for the spectrum of a trained weight matrix, we estimate its random part by fitting the
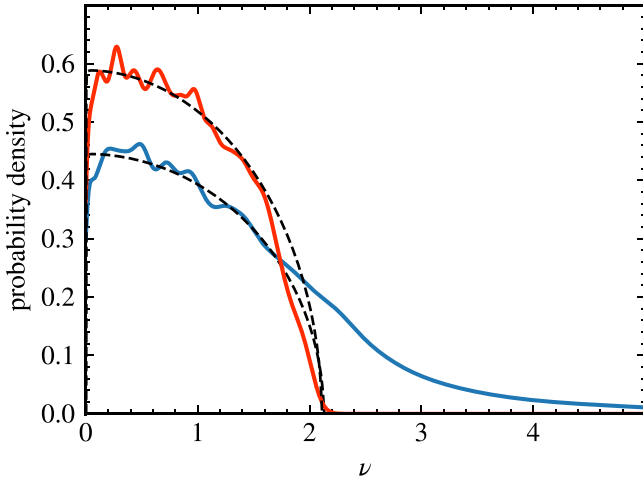
FIG. 1. Distribution of the singular values ν of the weight matrix of the second hidden layer of the three hidden layer network MLP512. The spectral distributions are calculated by broadening with a window size of 15 singular values. The red curve shows the distribution directly after random Glorot initialization, and the blue line depicts the result after fully training the network. The dashed, black lines are fits to the Marchenko-Pastur distribution Eq. (4). After random initialization, the spectrum agrees well with the RMT prediction, and even after training the bulk of the singular values still follows a modified Marchenko-Pastur distribution with similar parameters.

empirical spectra with a modified Marchenko-Pastur law in the following way: we set $\nu_{\min}$ equal to the smallest empirical eigenvalue, and then use $\nu_{\max}$ and σ^2 as fit parameters. This method has an additional free parameter as compared to the strict Marchenko-Pastur distribution, where the additional parameter can be understood as an estimate of the percentage of the spectrum that still follows the Marchenko-Pastur distribution.

Comparing the singular value spectra of the random weights at initialization (red line in Fig. 1) with those of the trained network (blue line in Fig. 1), it becomes apparent that the bulk of the singular values still follows a modified Marchenko-Pastur distribution with similar parameters (dashed, black line in Fig. 1). In addition, there are some larger singular values, which do not occur in the spectrum of the random control.

To further see that the majority of singular values of trained networks are indeed random and do not encode information, we consider the distribution of nearest-neighbor spacings of unfolded singular values. Here, unfolding refers to normalizing the mean density of states of the singular values ν_i to unity, yielding the unfolded spectrum ξ_i [42–46]. In contrast to the singular value distribution, which is nonuniversal and depends on the system at hand, the spacing distribution is a universal property of random matrices. For real random matrices in the universality class of the Gaussian orthogonal ensemble (GOE), the level spacings $s_k = \xi_{k+1} - \xi_k$, i.e., the differences between neighboring singular values, are distributed according to the Wigner surmise [42–47]

$$P_{\mathrm{GOE}}(s) = \frac{\pi s}{2} \exp\left(-\frac{\pi}{4} s^2\right). \qquad (5)$$
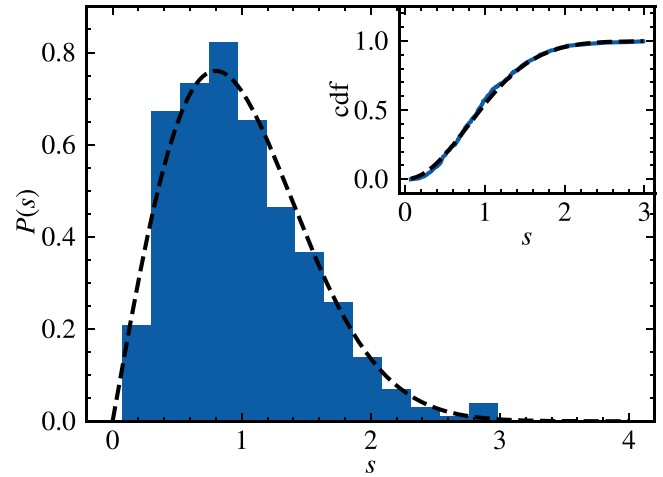

FIG. 2. Spacing distributions of unfolded singular values of the weight matrix of the second hidden layer of MLP512. The main panel depicts the probability density histogram and the inset shows the cumulative distribution. In addition, the RMT prediction (Wigner surmise) Eq. (5) for matrices from the GOE is indicated by a dashed black line. The prediction is confirmed both visually and by a Kolmogorov-Smirnov test, which at the 40% confidence level cannot reject the hypothesis that the Wigner surmise is the correct distribution.

The nearest-neighbor spacings of the weight matrix singular values are in excellent agreement with the RMT prediction Eq. (5) even after training the networks (Fig. 2). This is supported by a Kolmogorov-Smirnov test of the empirical data against Eq. (5) that cannot reject the null hypothesis even at a significance level as high as $\alpha = 0.40$.

Another prediction of RMT that allows to test the random nature of weight matrices is the level number variance, which is sensitive to long range correlations in the spectrum. The number variance describes fluctuations in the number of unfolded singular values $N_{\xi_i}(l)$ in intervals of length l around each singular value ξ_i,

$$\Sigma^2(l) = \langle (N_\xi(l) - l)^2 \rangle_\xi. \qquad (6)$$

For random matrices from the GOE universality class, the level number variance depends on the interval width ℓ according to $\Sigma^2(l) \propto \ln(2\pi l)$ in the regime $l \gtrsim 1$ [43–46] (dashed, black line in Fig. 3). This is in very good agreement with empirical results for the trained example network (Fig. 3). In particular, there are only small differences between randomly initialized weights (red lines in Fig. 3) and the fully trained weight matrices.

In addition, we consider the normalized eigenvectors of the $m \times m$ matrix $\mathsf{W}^\dagger \mathsf{W}$ (the right singular vectors of W), whose components in the case of a random matrix are described by the Porter-Thomas distribution [43,45,46], i.e., a Gaussian distribution with mean of zero and a standard deviation which is fixed by the normalization of the vector's length to unity as $1/\sqrt{m}$. To verify whether the observation that most singular values of trained networks are random also carries over to the associated eigenvectors, we test the empirical distribution of the entries of each eigenvector against such a Gaussian distribution using a Kolmogorov-Smirnov test. If the test returns a
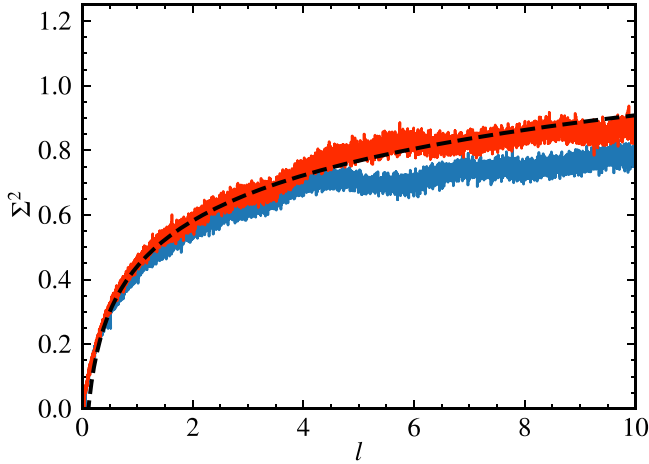
FIG. 3. Level number variance Σ^2 of singular values of the weights for the second hidden layer of MLP512, after initialization (red) and after fully training the network (blue). The dashed black line is the theory prediction for the GOE. We find that even after training, the level number variance grows only logarithmically, as predicted by RMT.

large p value, then the hypothesis of a Gaussian distribution cannot be rejected, which we interpret as an indication that the corresponding vector contains only noise. The rejection of the Gaussian hypothesis for small p values is an indication for stored information. To reveal a trend in the data, we average the resulting p values over neighboring singular values with a window size of 15. Indeed, we find that most eigenvectors are random, especially those belonging to small singular values (Fig. 4). For large singular values, we observe a decrease of the p values, indicating that relevant information is stored in the corresponding eigenvectors, which is also consistent with the results [64].

We further analyze the tail region of the singular value distribution, which was recently described in terms of power-law fits to the probability distribution function [64,67]. To this end, we consider the eigenvalues $\lambda = \nu^2$ of the matrix $W^\dagger W$, and analyze a potential power-law decay of the cumulative distribution function using the Hill estimator [77] obtained by averaging the inverse local slopes of the log-log cumulative distribution. A power-law tail would manifest itself in an extended flat region of a Hill plot, which is absent in Fig. 5. In Sec. VII we extend this analysis to a variety of overparameterized DNNs and do not find evidence for a power-law behavior in most of them, in contrast to the characterization of DNNs in terms of power-law fits to the distribution function of weight matrix singular values [64,67]. These findings are consistent with an analysis of p values for power-law fits.

III. UNIVERSALITY OF LEVEL SPACING DISTRIBUTION

To demonstrate that the results presented in the previous section are typical of trained DNNs, we consider several networks with different architectures and sizes here and in the following sections: (a) a fully connected feedforward network with layers of size [3072, 1024, 512, 512, 10] denoted as MLP1024 and (b) a convolutional network called miniAlexNet consisting of two convolutional layers followed
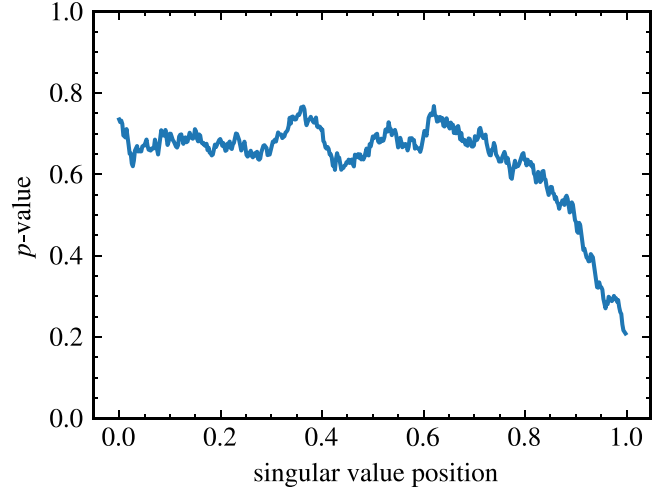

FIG. 4. Randomness of eigenvectors as a function of the singular value position in the spectrum: we quantify the agreement with the RMT Porter-Thomas distribution by computing the p value of a Kolmogorov-Smirnov test on the entries of the eigenvectors of $W^\dagger W$ for the second hidden layer of MLP512. On the x axis, we plot the positions according to the rank ordered singular values, such that 0 corresponds to the smallest and 1 to the largest singular value of the weight matrix. The results are averaged over neighboring singular values with a window size of 15. For large singular values, the hypothesis of random eigenvectors is rejected, indicating that information is stored in these singular values and eigenvectors.

by three dense layers, both trained on the CIFAR-10 [30] dataset. In addition, we analyze the two large networks (c) AlexNet [78] and (d) VGG19 [31], whose models trained on the ImageNet [79] dataset are available via `pytorch` [80] and `tensorflow` [81]. More details on the definition of the networks, training parameters, and performance of the fully trained networks can be found in Appendix A.
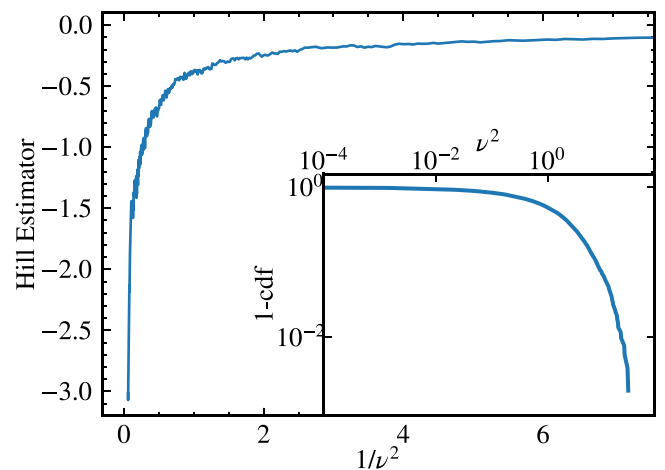

FIG. 5. Estimate of the tail exponent of the singular value spectrum of the second hidden layer of MLP512 obtained by averaging the inverse local slopes obtained via a Hill estimator with window size $a = 20$. The absence of a flat plateau region shows that no power law is present, even though this is not immediately evident in a double logarithmic plot of the cumulative distribution (inset).

TABLE I. Kolmogorov-Smirnov (K-S) test results of the distribution of unfolded singular value spacings of the weight matrices against the Brody distribution with $\beta = 1$. Rejection of the null hypothesis is based on the $\alpha = 0.05$ significance level. The p value indicates how likely it is to obtain a distribution with at least as much cumulative density function deviation as the one tested for drawing random numbers from a Brody distribution with $\beta = 1$. In addition, we show the results of a fit of a Brody distribution with free parameter β to the cumulative density function of the computed level spacings. The error was determined by a bootstrap sampling method. We find excellent agreement with the Wigner surmise for a variety of network architectures.

Network	Reject null hypothesis?			K-S test p value			Brody β from fit		
	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3
MLP512 (seed 1, Sec. II)	No	No	No	0.347	0.401	0.812	0.79 ± 0.09	1.01 ± 0.10	1.04 ± 0.12
MLP512 (seed 2)	No	No	No	0.993	0.421	0.844	0.99 ± 0.11	1.01 ± 0.11	0.99 ± 0.11
MLP512 (seed 3)	No	No	No	0.768	0.784	0.863	0.95 ± 0.11	0.93 ± 0.10	0.92 ± 0.09
MLP1024	No	No	No	0.799	0.812	0.792	0.94 ± 0.07	0.91 ± 0.11	1.03 ± 0.10
miniAlexNet (second conv. layer)		No			0.859			0.85 ± 0.14	
AlexNet (dense layers)—`torch`	No	No	No	0.670	0.229	0.160	0.96 ± 0.04	0.95 ± 0.04	0.83 ± 0.07
VGG16 (dense layers)—`tensorflow`	No	No	No	0.923	0.312	0.309	0.99 ± 0.04	0.92 ± 0.04	0.92 ± 0.07
VGG19 (dense layers)—`torch`	No	No	No	0.376	0.652	0.557	0.97 ± 0.04	0.95 ± 0.04	0.92 ± 0.07

To compare the various RMT predictions with the properties of empirical weight matrices, we compute their singular value decompositions. While this is straightforward for dense layers, in the case of convolutional layers we first reshape the four dimensional weight tensors to a rectangular shape (for details see Appendix B) and then compute their singular values and vectors. To obtain smooth probability densities for the singular value spectra, we perform a Gaussian broadening [44,82] by approximating the probability density as a sum of Gaussian functions centered around each of the m singular values ν_k with widths $\sigma_k = (\nu_{k+a} - \nu_{k-a})/2$, where a is the window size of the broadening [51,83]

$$P(\nu) \approx \frac{1}{m} \sum_{k=1}^{m} \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left[-\frac{(\nu - \nu_k)^2}{2\sigma_k^2}\right]. \qquad (7)$$

To compare the RMT prediction Eq. (5) with the level spacing of the networks weights, we have to unfold the singular value spectrum first. Here, unfolding is a transformation that maps the singular values ν_i to uniformly distributed singular values ξ_i [42–44,46,51]. For this purpose, we first determine the probability density $P(\nu)$ using Eq. (7) and calculate the corresponding cumulative distribution

$$F(\nu) = m \int_{-\infty}^{\nu} P(x)\, dx. \qquad (8)$$

The unfolded singular values are defined as $\xi_i = F(\nu_i)$. We then obtain the spacings of the unfolded and sorted singular values ξ_i via

$$s_k = \xi_{k+1} - \xi_k. \qquad (9)$$

We find excellent agreement with the RMT predictions for all network architectures and layers (Fig. 6). This is also supported by Kolmogorov-Smirnov tests (see Table I) with null hypothesis that the distribution is described by the Wigner surmise Eq. (5). We infer from the tests that for all network types, for fully connected as well as for convolutional layer, the null hypothesis cannot be rejected on the $\alpha = 0.05$ significance level. The p values of the Kolmogorov-Smirnov tests (Table I) even show that in many cases a rejection for much higher α is also not possible.

Furthermore, we consider the more general case of a Brody distribution [42,43,46]

$$P_{\mathrm{Br}}(s) = B(1 + \beta)s^\beta \exp(-Bs^{1+\beta}), \qquad (10)$$

with $B = \{\Gamma([\beta + 2]/[\beta + 1])\}^{1+\beta}$. For $\beta = 1$ this reduces to the Wigner surmise Eq. (5). Fits of the Brody distribution Eq. (10) to the data show very good agreement with $\beta \sim 1$ (see Table I). To obtain β and the error estimate, we use bootstrap sampling [84–86], fit the Brody distribution to each
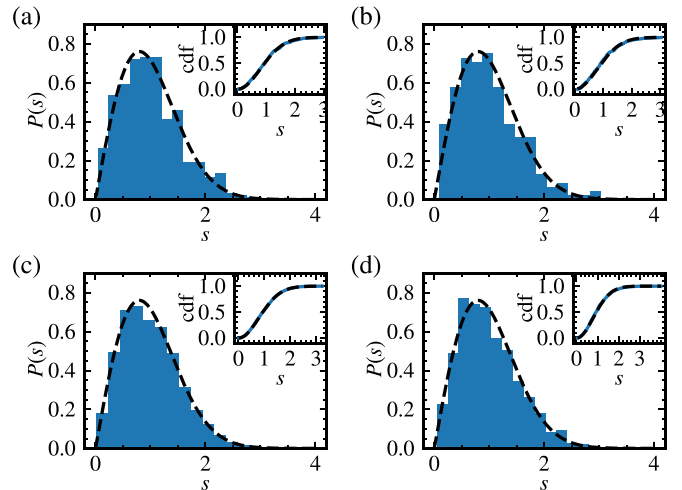

FIG. 6. Nearest-neighbor spacing distributions of unfolded singular values of weight matrices for various neural networks. The main panels depict the probability density histograms and the insets show the cumulative distribution functions. In addition, we depict the Wigner surmise theory prediction Eq. (5) for the GOE with dashed, black lines. (a) Results for the second hidden layer weight matrix of MLP1024. In addition, results for (b) the second convolutional layer in the CNN miniAlexNet, (c) the second fully connected layer in AlexNet, and (d) for the third dense layer in VGG19. In all cases there is excellent visual agreement with the RMT predictions. This is further supported by Kolmogorov-Smirnov tests which cannot reject the null hypothesis at a significance level of (a) 81%, (b) 85%, (c) 31%, and (d) 96%.

sample, and then calculate β as the mean and the error as the standard deviation of all fit results.

The level spacing results suggest that the majority of weights is random even after training. This indicates that the weights, even in the case of the large networks trained on ImageNet, have rather low information density.

IV. LEVEL NUMBER VARIANCE

The nearest-neighbor spacing distribution investigated above probes the level statistics locally. To probe long-range correlations between singular values, we compute the level number variance Eq. (6), which is considerably more sensitive to details of the singular value distribution [43,44]. To determine the level number variance numerically from the unfolded spectrum, for each l we repeatedly draw values $\xi_0 \in [\min(\xi_i) + l/2, \max(\xi_i) - l/2]$ and count the number n of ξ_i in the interval $[\xi_0 - l/2, \xi_0 + l/2]$. These values are used to compute an estimate of the variance according to Eq. (6). For random matrices from the Gaussian orthogonal ensemble (GOE) universality class, for large l the level number variance is given by [43,44,46]

$$\Sigma^2_{\text{GOE}}(l) \approx \frac{2}{\pi^2}\left[\ln(2\pi l) + \gamma + 1 - \frac{\pi^2}{8}\right], \quad (11)$$

where γ is the Euler-Mascheroni constant. It is known for GOE matrices that this formula is also a good approximation in the range of smaller l [44], but one needs to keep in mind that the level number variance computed for empirical spectra depends on the window size a chosen for broadening the spectrum. We do not plot the level number variance for large $\ell \gg a$ as the unfolding procedure breaks down when the window size is much larger than the window over which the distribution is averaged. This limited plot range still allows to make the important distinction between a linearly growing level number variance (uncorrelated singular values) and logarithmic growth (singular values of a matrix with random bulk).

We find good agreement of Eq. (11) (dashed, black line in Fig. 7) with the results for trained networks (Fig. 7) using a window size $a = 15$. In particular, there are only small differences between randomly initialized weights (red lines in Fig. 7) and the fully trained weight matrices.

V. DISTRIBUTION OF ENTRIES OF THE SINGULAR VECTORS

In addition to the singular values, we also analyze the singular vectors of the weight matrices of trained DNNs. For an $n \times m$ weight matrix W, we consider the eigenvectors of $\mathsf{W}^\dagger\mathsf{W}$ (right singular vectors) if $m \leqslant n$ or $\mathsf{W}\mathsf{W}^\dagger$ (left singular vectors) otherwise. For a completely random matrix of rank m, RMT predicts that the entries of normalized eigenvectors follow a Gaussian distribution with zero mean and standard deviation $\sigma = 1/\sqrt{m}$. We test for agreement between the empirical distribution of the m entries of each singular vector with this RMT prediction with the help of the Kolmogorov-Smirnov test, and average the resulting p values over neighboring rank ordered singular values with a window size of 15. A large p value indicates a random singular vector,
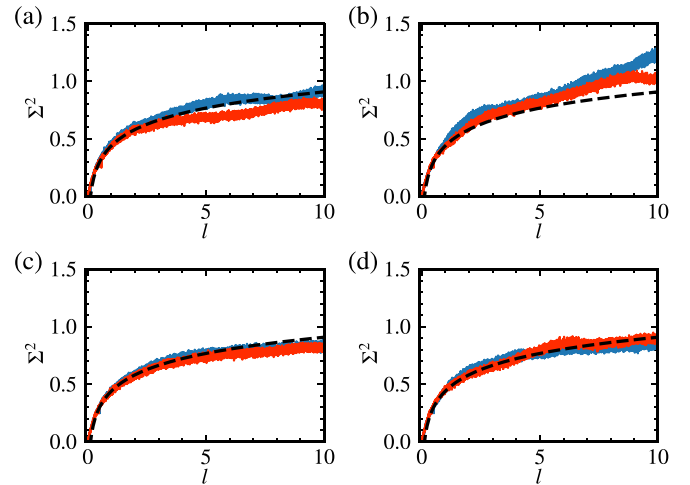

FIG. 7. Level number variance of singular values of the weights for (a) the second hidden layer of MLP1024, (b) the second convolutional layer in the CNN miniAlexNet, (c) the second fully connected layer in AlexNet, and (d) for the third dense layer in VGG19. Red curves show the results for initialized weights and blue lines depict level number variances for fully trained networks. The dashed, black lines depict the theory prediction Eq. (11) for the GOE. In all cases, the level number variance grows logarithmically even after training. Particularly for large networks in panels (c) and (d), where the statistics are most reliable, deviations from the RMT prediction are small.

as observed for small and intermediate singular values for the MLP1024 network and the large pretrained networks in the left panels of Figs. 8(a) and 8(b), respectively. For large singular values, the p values decrease, suggesting that the Gaussian hypothesis is rejected, and that information is stored in these vectors. For the convolutional layer of the CNN in Fig. 8(c), we note that eigenvectors corresponding to both small and large singular values show significant deviations from random behavior. However, the resulting matrix $\mathsf{W}\mathsf{W}^\dagger$ only has rank 300 which makes the analysis less reliable due to statistical uncertainty.

As a second measure for the randomness of singular vectors, we study their degree of localization as measured by the inverse participation ratio (IPR) [43,45,46]:

$$\text{IPR}(\boldsymbol{v}) = \sum_{i=1}^{m} |v_i|^4. \quad (12)$$

To get some intuition for the IPR, we consider two examples: (i) for a normalized uniform m-dimensional vector with equal entries $1/\sqrt{m}$, the IPR is given by $1/m$, the inverse of the number of relevant components. In the case (ii) of a vector with only one nonzero entry, the IPR is equal to unity, again the inverse of the number of relevant components. Since eigenvectors of GOE matrices have many relevant components, the IPR random vectors is on the order of $1/m$, while a larger value indicates deviations from RMT and the presence of learned information. This analysis is in very good agreement with that of the p values (see right panels in Fig. 8) for large singular values.
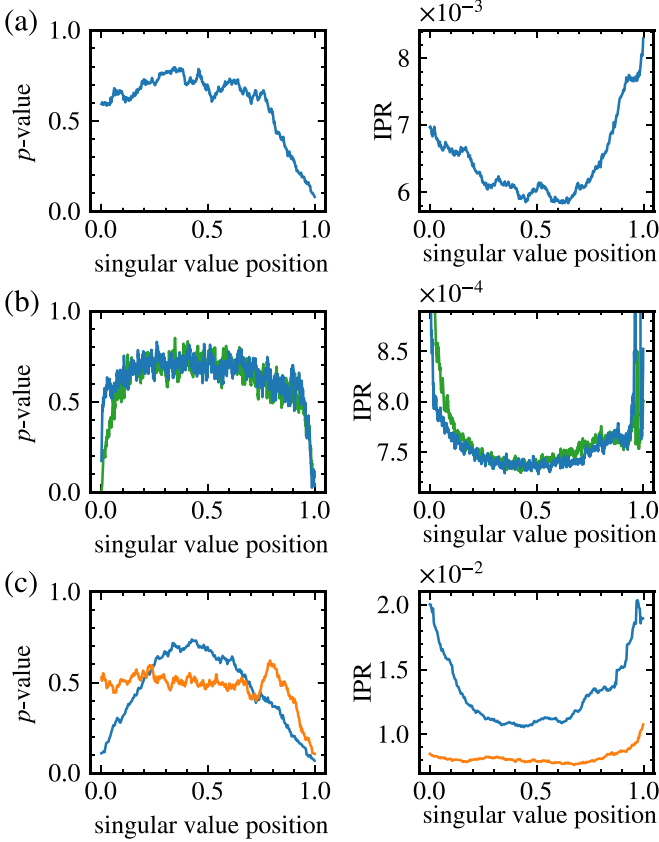
(a)

(b)

(c)

FIG. 8. Analysis of the eigenvectors of $W^\dagger W$. The left panels depict the p values of Kolmogorov-Smirnov tests of the eigenvector entries versus a Gaussian distribution. The right panels show the inverse participation ratios of eigenvectors computed according to Eq. (12). All results are averaged over neighboring eigenvectors with a window size of 15. The x direction describes the position of rank ordered singular values, such that 0 corresponds to the smallest and 1 to the largest singular value of each weight matrix. We show results for (a) the second hidden layer of MLP1024, (b) the first dense layer of the large pretrained DNNs AlexNet (blue) and VGG19 (green), and (c) the second convolutional layer (blue) and first dense layer (orange) of the CNN miniAlexNet. The results for the p values are consistent with those for the inverse participation ratio, indicating that relevant information is stored in eigenvectors corresponding to large singular values.

VI. RMT ANALYSIS OF DIFFERENT LEARNING REGIMES

It was shown [34,87–91] that neural networks can achieve good generalization accuracies even when their weights change only by very small amounts during training. The opposite to this *lazy learning* is denoted as *rich learning*, where the final weights W after training deviate significantly from the initial ones W_0. Based on this, a criterion for estimating the learning regime was proposed by Chizat *et al.* [69]: For a neural network f_W that maps an input x to an output, and an accuracy function $\mathcal{A}(f_W, \{x\}, \{y\})$, where $\{x\}$ is a dataset with labels $\{y\}$, one computes the network's linearization around the initial weights W_0:

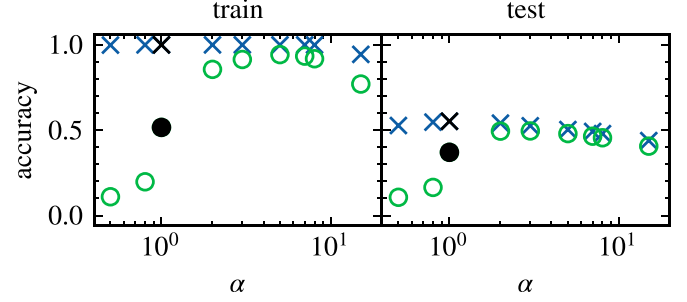$$\tilde{f}_W(x) = f_{W_0}(x) + (W - W_0) \cdot \nabla_W f_W|_{W_0}(x). \quad (13)$$

FIG. 9. Comparison of training and test accuracies for full MLP512 networks (blue crosses) and linearized networks (green circles) around the initial weights of the second layer as a function of the laziness hyperparameter α. The black symbols indicate accuracies for $\alpha = 1$, similar to the training presented in Sec. II. For small $\alpha < 1$ accuracies of linearized and full networks deviate significantly which indicates rich learning, while for large $\alpha > 1$ performance differences are small indicating lazy learning.

In the lazy learning regime, where $W \approx W_0$, linearization is a good approximation such that the accuracies are barely different, i.e.,

$$\mathcal{A}(f_W, \{x\}, \{y\}) \approx \mathcal{A}(\tilde{f}_W, \{x\}, \{y\}). \quad (14)$$

On the contrary, in the rich learning regime, one expects significant deviations such that

$$\mathcal{A}(f_W, \{x\}, \{y\}) \gg \mathcal{A}(\tilde{f}_W, \{x\}, \{y\}). \quad (15)$$

This criterion has the advantage that it can also be studied on a layer-wise basis by linearizing around a single weight matrix only, and as accuracies are in the range [0,1], it gives a scale for laziness comparable between different network architectures. A disadvantage is that it requires to compute the linearization which can be resource-intensive for large networks. For obtaining $\tilde{f}_W$, we use the `autodiff` implementation in the `jax` python package together with the `neural_tangents` package [92].

We train several MLP512 networks, where laziness is controlled by introducing a hyperparameter α that modifies the output activations via [69]

$$a_L = \text{softmax}[\alpha(W_L a_{L-1} + b_L)] \quad (16)$$

and the cost function as

$$l(W, b) = -\frac{1}{N\alpha^2} \sum_{k=1}^{N} y^{(k)} \cdot \ln\left(a_{\text{out}}^{(k)}\right). \quad (17)$$

Here, a large $\alpha > 1$ scales down the gradient updates and therefore encourages lazy learning, while small $\alpha < 1$ steers training toward the rich learning regime [69]. In the following, we focus on the second hidden layer of MLP512 networks considered in Sec. II (see Appendix C for other layers). Figure 9 depicts training accuracy (left panel) and test accuracy (right panel) as a function of the laziness parameter α. As expected, the networks are in the rich regime for $\alpha < 1$, where the full networks (blue crosses) perform significantly better than the linearized networks (green circles), while we observe lazy learning for $\alpha > 1$. The network with $\alpha = 1$ (black symbols), lies about in the middle between the two regimes, where
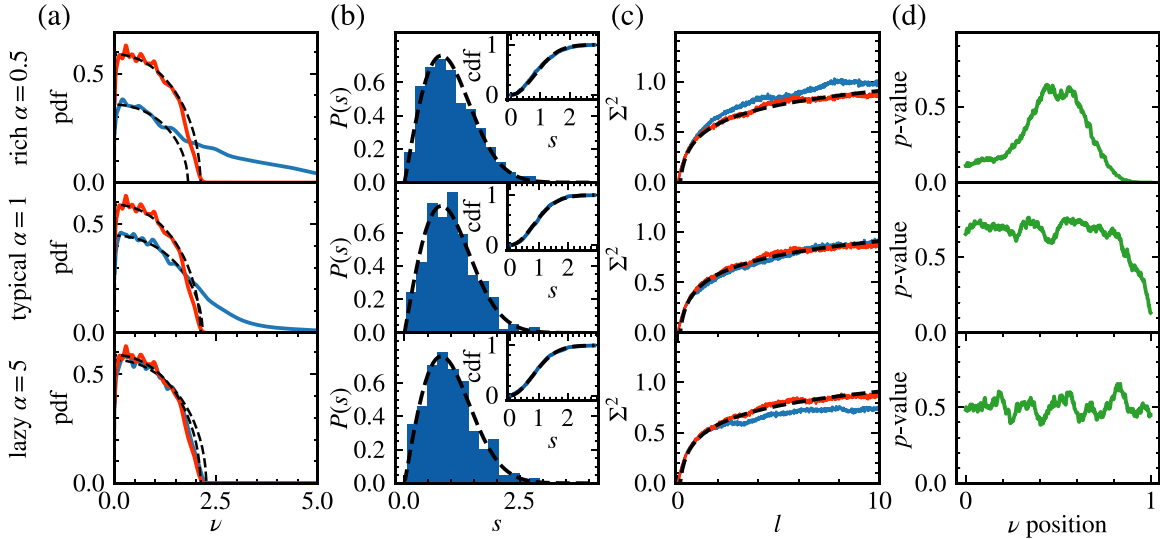
FIG. 10. Random matrix theory analysis of second layer weights of MLP512 networks trained in different learning regimes: rich learning ($\alpha = 0.5$, top panel), typical learning ($\alpha = 1$, middle panel), and lazy learning ($\alpha = 5$, bottom panel). We show (a) the spectra for trained (blue) and randomly initialized networks (red) together with fits of modified Marchenko-Pastur laws (dashed, black), (b) unfolded level spacing distributions (main panel, blue, window size 15) and corresponding cumulative distributions (insets) together with the Wigner surmise (dashed, black), (c) unfolded level number variance (trained: blue, initialized: red), and (d) p values for a comparison of singular vector entries to a Porter-Thomas distribution. While trained networks in all cases follow universal RMT predictions (b and c), indicating a random bulk, lazy networks can be distinguished from typical and rich networks by the spectral distributions in panel (a) and p values in panel (d).

we also find the best test accuracy. We therefore denote $\alpha = 1$ as *typical learning*.

A comparison between the RMT analysis in the three regimes, rich $\alpha = 0.5$ (top panel), typical $\alpha = 1$ (middle panel), and lazy $\alpha = 5$ (bottom panel) in Fig. 10 reveals:

(1) For all networks, the bulks of the spectra are random such that the universal properties, i.e., level spacings [panel (b)] and level number variance [panel (c)] agree with RMT predictions. By comparing the level number variance curves for networks trained with various α (not shown), we confirm that the level number variances in panel (c) display typical deviations from the RMT prediction such that there is no trend of slower grow for networks with larger α.

(2) The rich network has more large singular values compared to the typical one, while the lazy network has almost an unchanged Marchenko-Pastur spectrum [panel (a)]. It is therefore surprising that it still achieves a respectable test accuracy of 50.4%, compared to 52.7% for the rich network and 55.2% for the typical network.

(3) The p values for Kolmogorov-Smirnov tests of the eigenvector entries against a Porter-Thomas distribution [panel (d)] are small only for large singular values in the typical case. In the rich case, we observe small p values also for singular vectors corresponding to the smallest singular values, and for the lazy network all p values fluctuate only slightly around 0.5 as expected for random weights.

These findings indicate that networks trained in the lazy regime do not deviate from RMT predictions after training, in striking contrast to rich and typical networks. Thus, an analysis of singular value spectra and singular vector entry distributions can be used to estimate the learning regime of a network on the level of the individual weights, without the

need for a potentially resource intensive linearization of the networks.

The almost perfect agreement with RMT predictions in the lazy regime raises the question whether the information that allows the network to still generalize quite well is encoded in parts of the spectrum that follow bulk statistics. If this was the case, then it would seem impossible to locate this information by means of an RMT analysis. However, we argue in the following that this is in fact not the case for the networks considered here. However, the RMT analysis in the lazy regime faces two major obstacles that make it difficult to detect the information hidden by the dominant random bulk: (i) an individual layer in the lazy network might carry very little information, as it is for example the case for the second hidden layer of the lazy MLP512 network shown in Fig. 10, where by replacing the final weight with the initial one for this layer, the generalization accuracy of the network only drops from 50.4% to 42.3%. It is therefore not surprising that no extended region in the spectrum containing information is found by the RMT analysis. (ii) In the lazy case, the difference $\delta W = W - W_0$ between initial weight W_0 and the final weight after training W is small, i.e. $||\delta W||/||W_0|| \ll 1$. In this case, the sensitivity of RMT is not sufficient to detect signatures of δW in W: For instance, the crossover from the Gaussian unitary ensemble to Poisson statistics has been studied in Ref. [93], and it turns out that δW would need to be much larger than in our case to have a noticeable effect compared to the bulk statistics from W_0.

As a solution, we suggest analyzing the statistical properties of the difference matrix δW instead of the full weight matrix. Such an analysis indeed indicates that δW consists of a random bulk and a low-rank contribution that encodes relevant information (see Appendix C).

VII. HILL ESTIMATOR FOR TAIL SPECTRA

In previous work [64,67], it has been argued that the spectral density of the singular values of DNN weight matrices can be fitted by a power law. To study the distribution of large singular values further, we employ the Hill estimator for power-law tail exponents [77], which has been widely used in the applied finance, economics, and statistics literature [95–101]. In addition, we also study p values of power-law fits and find agreement with the results based on Hill estimators.

For this purpose, we first rank order the eigenvalues $\lambda_i = \nu_i^2$ of the weight matrices and compute the corresponding cumulative distribution function as

$$F(\lambda_i) = \frac{i}{N},\tag{18}$$

where a small index corresponds to a small eigenvalue. The Hill estimator h is then obtained from the local inverse slopes

$$\zeta_i = \frac{-\ln[\lambda_{i+1}/\lambda_i]}{\ln[F(\lambda_{i+1})/F(\lambda_i)]}\tag{19}$$

by averaging over a surrounding eigenvalues

$$\tilde{\lambda}_i = \frac{1}{a}\sum_{j=-a/2}^{a/2}\lambda_{i+j},\tag{20}$$

$$h(\tilde{\lambda}_i) = \left(\frac{1}{a}\sum_{j=-a/2}^{a/2}\zeta_{i+j}\right)^{-1}.\tag{21}$$

The Hill estimator is sensitive to changes in the slopes of the log-log cumulative distribution. In the presence of a power-law tail, the value of the Hill estimator depends only weakly on the eigenvalue range over which it is determined, and the extrapolation $1/\lambda^2 \to 0$ yields the tail exponent of the distribution. According to this criterion, we do not find evidence of power-law tails for (i) MLP1024, (ii) miniAlexNet, or (iii) AlexNet [Figs. 11(a)–11(c)]. Only in the case of VGG19 [Fig. 11(d)] there is a region of intermediate singular values in the Hill plot with a power-law exponent of approximately one, but in the asymptotic regime of large singular values the exponent drops to approximately two, invalidating the concept that a single power law characterizes the distribution.

To substantiate these findings, we additionally fit power laws $p(x) \propto x^{-\alpha}$ to the spectra, compute their respective p values using the algorithm by Clauset *et al.* [94], and compare the hypothesis of a power law to a truncated power law $p(x) \propto x^{-\alpha}e^{-\lambda x}$ with log-likelihood ratio tests (see also Appendix D). The fitting procedure and the comparison of the power-law hypothesis to other distributions is implemented in the `powerlaw` package while the p values are computed by numerical approximation of the test statistic with 2500 synthetic power-law fits followed by a Kolmogorov-Smirnov test as described in Ref. [94]. Of the four spectra analyzed, the Hill plot does not look consistent with a straight line in cases (a), (c), and (d), while case (b) fluctuates around a straight line. Indeed, for AlexNet (panel c) and MLP512 (panel a) the p values reject the power-law hypothesis. For the third dense layer in VGG19 (panel d), a power law cannot be rejected based on $p = 0.059$, slightly larger than the threshold. From the likelihood ratio test however, we find better agreement for a truncated power law (in agreement with Ref. [64])
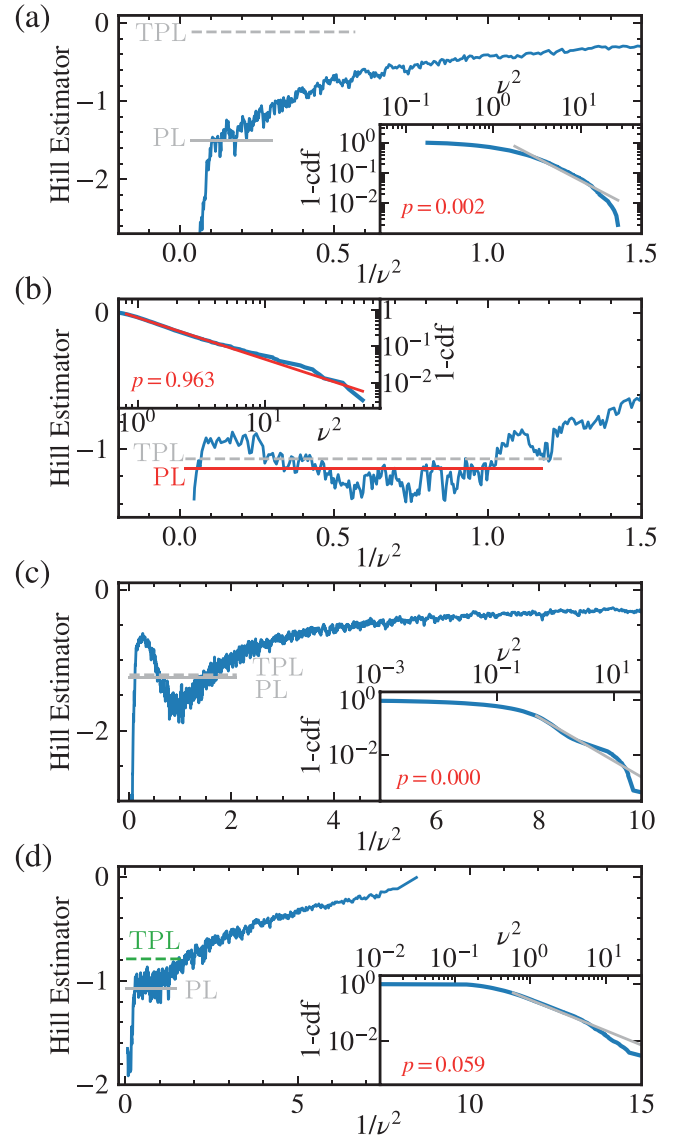

FIG. 11. Analysis of the cumulative distribution of DNN singular values in the tail region. Hill estimators of the weight matrix spectra for (a) the second layer weight matrix of MLP1024, (b) the second convolutional layer of the CNN miniAlexNet, (c) the second fully connected layer of AlexNet, and (d) the third fully connected layer of VGG19. The Hill estimators are obtained using Eq. (20) and Eq. (21) with a window size $a = 20$. The insets depict the corresponding log-log cumulative distributions (blue). In addition, we show results of a power-law fit $p(x) \propto x^{-\alpha}$ (solid gray or red) and a truncated power-law fit $p(x) \propto x^{-\alpha}e^{-\lambda x}$ (dashed gray or green) to the tails, where a power-law tail corresponds to a flat Hill estimator at $-(\alpha - 1)$ starting from $\nu_{\min}$. We show rejected fits in gray and accepted fits colored, based on p values [94] for the power laws and log-likelihood ratio tests versus truncated power laws (see also Table III in the Appendix).

In summary, three of the four examples analyzed here are not described by power-law tails. In Appendix D Table III we present results for additional networks, and in only two of eight cases a power law cannot be rejected. Hence, we conclude that there is no evidence that the singular values of DNN weight matrices are generically described by a

TABLE II. Neural network architectures and performance of trained networks. We use d to indicate a dense layer, c for a convolutional layer, p for max pooling, f for flattening, and r for response normalization layer (with a depth radius of 5, a bias of 1, $\alpha = 1$, and $\beta = 0.5$).

	Network	Dataset	Training acc	Test acc
(i)	MLP512, seed 1 (Sec. II) {d 3072, d 512, d 512, d 512, d 10} [34]	CIFAR-10	100%	54.7%
	MLP512, seed 2 {d 3072, d 512, d 512, d 512, d 10}	CIFAR-10	100%	55.1%
	MLP512, seed 3 {d 3072, d 512, d 512, d 512, d 10}	CIFAR-10	100%	55.2%
(ii)	MLP1024 {d 3072, d 1024, d 512, d 512, d 10}	CIFAR-10	100%	55.4%
(iii)	miniAlexNet {c 300 5×5, p 3×3, r, c 150 5×5, p 3×3, r, f, d 384, d 192, d 10} [34]	CIFAR-10	100%	78.5%
(iv)	AlexNet [78]—torch	ImageNet		56.5%
(v)	VGG16 [31]—tensorflow	ImageNet		67.6%
(vi)	VGG19 [31]—torch	ImageNet		72.4%

power-law tail distributions, and argue that the exponent resulting from a power-law fit to the singular value probability density function can only be viewed as a heuristic tool to characterize different spectra but not a genuine property of the tail of the distribution of singular values. In addition, given the absence of power-law tails in most of the singular value distributions studied here, it is unclear whether weight matrices of fully trained DNNs indeed belong to an ensemble of heavy tailed random matrices as suggested in Ref. [67].

VIII. CONCLUSIONS

The complexity of overparameterized DNNs makes it difficult to understand their learning and generalization behavior. In light of this, we have applied RMT as a zero information hypothesis to separate randomness from information. In particular, since at initialization weights are chosen randomly from a probability distribution, the corresponding weight matrices agree perfectly with predictions of RMT before training. Specifically, the singular value spectra of initialized networks are governed by a Marchenko-Pastur distribution, the level spacing distribution follows the Wigner surmise, and the level number variance only grows logarithmically. By comparing these characteristics between randomly initialized and trained networks, one can understand which parts of the weight matrix singular value spectrum stores information. This approach works well for image classification problems where we find that the underlying rule in the dataset is stored as a low-rank contribution in the trained weights. It remains an open question to what extent this can be applied to networks trained for other types of problems.

We find that even in fully trained DNNs large parts of the eigenvalue spectrum remain random. In particular, we demonstrate that the agreement between the level spacing distribution of the bulk of singular values for fully trained networks and the parameter free Wigner surmise is excellent, and that the even more sensitive level number variance continues to agree with the RMT prediction as well. In agreement with the spectra, an analysis of the singular vectors reveals that they are also predominantly random, except for the ones corresponding to the largest singular values. This shows that the majority of the weight matrix does not contain relevant information, and that learned information may be concentrated in the largest singular values and associated vectors only. Interestingly, we find the strongest deviations from RMT

predictions for networks trained in the so-called rich regime, while for networks trained in the lazy regime the whole spectrum and all eigenvectors follow RMT results. Since the best generalization performance for learning is observed in between lazy and rich regime, the ability to efficiently determining the learning regime on a layer-wise basis using the singular value spectra and singular vectors could allow for algorithms that dynamically steer the training toward the desired regime. An analysis of the tails of the eigenvalue spectra using the Hill estimator shows that the distributions are heavy tailed but there is no evidence for a single power law in the tail region of the distribution in general.

APPENDIX A: DETAILS ON NEURAL NETWORKS

In this article, we consider a variety of different networks to show that our results are valid for a wide range of architectures. Table II lists the network architectures, training datasets, and accuracies achieved on each dataset. We downloaded the large pretrained networks (iv) AlexNet [78] via pytorch [80], (v) VGG16 [31] via tensorflow [81], and (vi) VGG19 [31] via pytorch [80]. Networks (i–iii) are trained using mini-batch stochastic gradient decent for 100 epochs. The weights are initialized using the Glorot uniform distribution [73] and the biases are initialized with zeros. We standardize each image of the CIFAR-10 dataset by subtracting the mean and dividing by the standard deviation. We set the learning rate to 0.001 at the beginning and use an exponential learning rate schedule with decay constant 0.95. For all networks, we choose 0.95 as momentum and the mini-batch size is 32. Network architectures (i–ii) in Table II are trained without L_2 regularization, while we use an L_2 regularization strength of 10^{-4} for training miniAlexNet networks (iii).

APPENDIX B: RESHAPING OF CONVOLUTIONAL LAYER WEIGHTS

In the case of convolutional layers we have to reshape the filters before computing the singular value decomposition [102]. We unfold the four dimensional weights in such a way
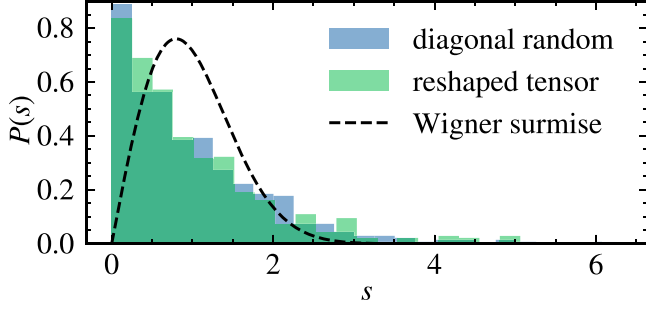
FIG. 12. Level spacing distribution of a diagonal i.i.d. Gaussian random matrix (blue) and for comparison for a tensor $10 \times 10 \times 600 \times 600$ where the 100 constituent matrices are diagonal with i.i.d. Gaussian random entries (green). To compute the singular value decomposition of the tensor, it is reshaped to size 60000×600 as described in Eq. (B1). We find that reshaping conserves the Poisson statistics that is very different from the Wigner surmise (black).

that the number of rows corresponds to the largest dimension of the weights. Each row is then filled with the remaining three dimensions that are listed with the smallest dimensions last to keep points that are close in the 3D tensor also close in 1D. In formal terms, this means that for a convolutional layer W of the form (K, L, M, N), where without loss of generality $K \geqslant L \geqslant M \geqslant N$, the reshaped matrix $\tilde{\mathsf{W}}$ with dimensions $(K, L \cdot M \cdot N)$ is related to the original matrix in the following way:

$$\tilde{\mathsf{W}}_{k,(l \cdot M \cdot N + m \cdot N + n)} = \mathsf{W}_{k,l,m,n}. \tag{B1}$$

All indices start at zero. While this procedure is to some degree arbitrary we convinced ourselves that similar methods give the same qualitative results. We further checked that this reshaping preserves the structure of the original filters: we construct a 4D tensor of shape $10 \times 10 \times 600 \times 600$ by drawing 100 times a diagonal, i.i.d., Gaussian 600×600 matrix, for which the level spacing distribution is a Poisson distribution. After reshaping, we find that the resulting level spacing distribution of the 2D matrix remains Poisson distributed (Fig. 12). This is the theoretically expected distribution of a diagonal matrix [42].
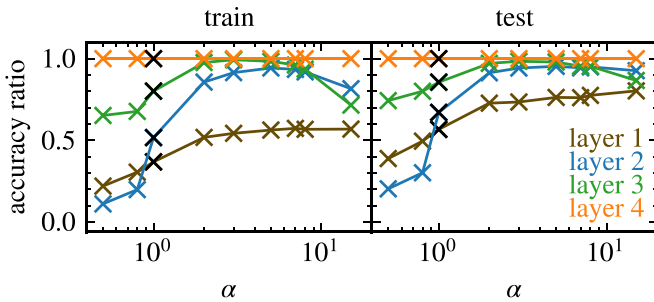

FIG. 13. Comparison of training and test accuracies for full MLP512 networks and linearized networks around the initial weights of several layers (first layer: brown, second layer: blue, third layer: green, output layer: orange) as a function of the laziness hyperparameter α. Crosses show the ratio between linearized and full network accuracy and lines are a guide to the eye. The black crosses indicate accuracy ratios for $\alpha = 1$, similar to the training presented in Sec. II.
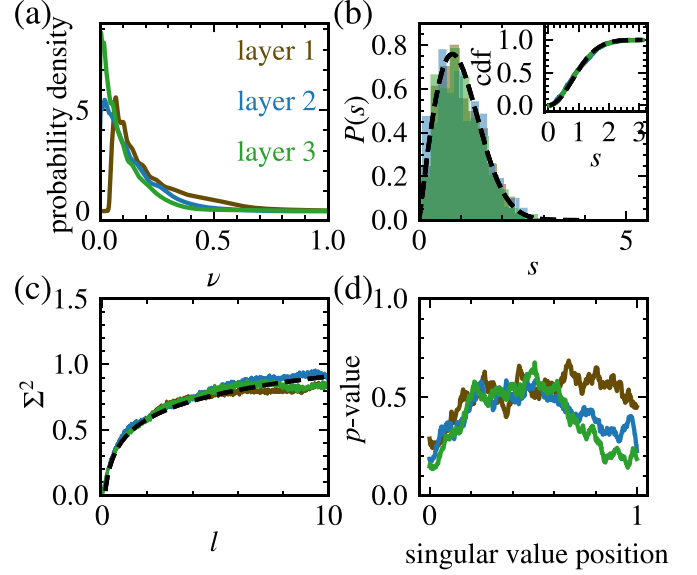

FIG. 14. Random matrix theory analysis for the difference $\delta W^{(l)} = W^{(l)} - W_0^{(l)}$ between the trained weights $W^{(l)}$ and the initial weights $W_0^{(l)}$ for the lazy MLP512 network trained with $\alpha = 5$. We show results for the first hidden layer (brown), second hidden layer (blue), and third hidden layer (green). (a) Singular value spectra obtained via Gaussian broadening. (b) Level spacing statistics of the unfolded singular value spectra. (c) Level number variance of the unfolded spectra. (d) Averaged p values for a comparison between singular vector entries and the Porter-Thomas distribution with window size $a = 15$. The spectrum of $\delta W^{(l)}$ agrees well with the Wigner surmise and a logarithmically growing level number variance. However, the p values in panel (d) indicate that again large parts of the spectrum of $\delta W^{(l)}$ are random and that information is stored in the largest singular values and corresponding vectors of $\delta W^{(l)}$. In addition, the singular value spectrum is not of Marchenko-Pastur type.

APPENDIX C: LEARNING REGIMES OF OTHER MLP512 LAYERS

We additionally compare test and training accuracies similar to main text Fig. 9 for the other layers of the MLP512 networks trained with various laziness parameters α. In Fig. 13, we show the ratio of the training accuracy between linearized and full networks, where a large ratio ≈ 1 indicates lazy learning and small values indicate rich learning. We find that for the second layer, α interpolates nicely between lazy and rich regime, the first layer tends toward rich learning, and the weights for the third layer are closer to lazy learning. The weight of the output layer, which has only rank ten, is always lazy. We note that this imbalance between the layers could be lifted by using different learning rates per layer.

In the lazy learning regime the trained weights $W^{(l)}$ remain close to the initial random matrices $W_0^{(l)}$ such that the random bulk dominates any RMT analysis of $W^{(l)}$ and therefore masks small local deviations from the bulk statistics. We thus consider the deviations from the initial weights $\delta W^{(l)} = W^{(l)} - W_0^{(l)}$, which again have random bulk statistics. Their unfolded spectra follow the Wigner surmise [Fig. 14(a)], and one finds a logarithmic increase of the level number variance

(panel c). In contrast to the full weight matrix, the difference matrix has a qualitatively different distribution function of the singular values (panel a) and the p values (panel d) are consistent with information being stored in parts of the spectrum corresponding to large singular values similar to what we found in the main text for networks in the typical learning regime.

APPENDIX D: DETAILS ON THE TAIL DISTRIBUTIONS OF WEIGHT SPECTRA

In addition to the results of Fig. 11, we fit power laws to the tails of the other dense layers of the `pytorch` model of AlexNet and VGG19, also considered in Ref. [64]. We also provide the p values for the power-law fits [94] and the results of log-likelihood ratio tests between truncated power law $p_{\mathrm{TPL}}(x) \propto x^{-\alpha} e^{-\lambda x}$ and power law $p_{\mathrm{PL}}(x) \propto x^{-\alpha}$. The log-likelihood ratio for n data points in the tail x_i is defined as [94]

$$R = \frac{1}{\sqrt{2n}\sigma} \ln \prod_{i=1}^{n} \frac{p_{\mathrm{TPL}}(x_i)}{p_{\mathrm{PL}}(x_i)}, \tag{D1}$$

where σ is the empirical standard deviation of the difference $\ln p_{\mathrm{TPL}}(x_i) - \ln p_{\mathrm{PL}}(x_i)$. A positive sign of R thus indicates that a truncated power law is a better fit, and a negative sign indicates a better fit for a power law. The p_2 value is then defined as the probability to obtain a ratio with magnitude of at least $|R|$ from a distribution $p(R)$ centered at zero with standard deviation σ, i.e., the probability that the sign is only

TABLE III. Power-law fit results and log-likelihood ratio tests between truncated power law and power law for the dense layers of AlexNet, VGG19, and the weights considered in Fig. 11. Here, d denotes a dense layer and c a convolutional layer. We reject the power law if $p < 0.05$ or in the case where the two-distribution test favors a truncated power law (positive R, $p_2 < 0.05$) [94].

Network	Layer	PL fit p	2-distr. R	2-distr. p_2	PL fit α	PL fit $x_{\min}$	Reject PL
MLP1024	d2	0.0019	2.50	0.000	2.51	3.389	Yes
mAlexNet	c2	0.9628	1.02	0.363	2.14	0.852	No
AlexNet	d1	0.0004	1.55	0.359	2.29	0.418	Yes
	d2	0.0004	1.25	0.211	2.25	0.480	Yes
	d3	0.9990	−0.003	0.999	3.02	2.046	No
VGG19	d1	0.0011	2.01	0.142	2.27	0.275	Yes
	d2	0.0007	1.98	0.055	2.19	0.291	Yes
	d3	0.0590	2.26	0.001	2.07	0.690	Yes

due to fluctuations. Therefore, small $p_2 < 0.05$ indicate a reliable sign of R, while large p_2 indicate an unreliable sign from fluctuations and hence an inconclusive test. For the dense layers of the large pretrained networks (see Table III), we only accept a power law in a single case, the third dense layer of AlexNet. For the third dense layer of VGG19, a power law is not rejected, however, a truncated power law yields a better fit. For the other layers, already the power law is rejected based on $p < 0.05$.

[1] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems*, edited by F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Red Hook, NY, 2012), Vol. 25.

[4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, Mastering the game of go without human knowledge, Nature (London) **550**, 354 (2017).

[5] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[6] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli, Statistical mechanics of deep learning, Annu. Rev. Condens. Matter Phys. **11**, 501 (2020).

[7] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, Mastering the game of go with deep neural networks and tree search, Nature (London) **529**, 484 (2016).

[8] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A.

Potapenko *et al.*, Highly accurate protein structure prediction with alphafold, Nature (London) **596**, 583 (2021).

[9] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nat. Phys. **14**, 447 (2018).

[10] C. Cao, S.-Y. Hou, N. Cao, and B. Zeng, Supervised learning in Hamiltonian reconstruction from local measurements on eigenstates, J. Phys.: Condens. Matter **33**, 064002 (2020).

[11] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).

[12] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**, 435 (2017).

[13] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine Learning Phases of Strongly Correlated Fermions, Phys. Rev. X **7**, 031038 (2017).

[14] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Machine learning quantum phases of matter beyond the fermion sign problem, Sci. Rep. **7**, 8823 (2017).

[15] P. Huembeli, A. Dauphin, and P. Wittek, Identifying quantum phase transitions with adversarial neural networks, Phys. Rev. B **97**, 134109 (2018).

[16] C. Lee, J. Kim, D. Babcock, and R. Goodman, Application of neural networks to turbulence control for drag reduction, Phys. Fluids **9**, 1740 (1997).

[17] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the

incompressible Navier-Stokes equations, J. Comput. Phys. **426**, 109951 (2021).

[18] Z. Chen, J. Gao, W. Wang, and Z. Yan, Physics-informed generative neural network: An application to troposphere temperature prediction, Environ. Res. Lett. **16**, 065003 (2021).

[19] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, J. Ngadiuba, M. Pierini, R. Rivera, N. Tran *et al.*, Fast inference of deep neural networks in FPGAs for particle physics, J. Instrum. **13** P07027 (2018).

[20] D. Guest, K. Cranmer, and D. Whiteson, Deep learning and its application to LHC physics, Annu. Rev. Nucl. Part. Sci. **68**, 161 (2018).

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE **86**, 2278 (1998).

[22] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, Object recognition with gradient-based learning, in *Shape, Contour and Grouping in Computer Vision* (Springer, Berlin, 1999), pp. 319–345.

[23] R. Pascanu, Y. N. Dauphin, S. Ganguli, and Y. Bengio, On the saddle point problem for nonconvex optimization, arXiv:1405.4604.

[24] Y. N Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2014), Vol. 27.

[25] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, Qualitatively characterizing neural network optimization problems, arXiv:1412.6544.

[26] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, Exploring generalization in deep learning, in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2017), Vol. 30.

[27] D. Soudry and E. Hoffer, Exponentially vanishing sub-optimal local minima in multilayer neural networks, arXiv:1702.05777.

[28] Siyuan Ma, Raef Bassily, and Mikhail Belkin, The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning, in *Proceedings of the International Conference on Machine Learning* (2018), p. 3325.

[29] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, Generalization in deep learning, arXiv:1710.05468.

[30] A. Krizhevsky, G. Hinton *et al.*, Learning multiple layers of features from tiny images, Technical report (2009).

[31] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556.

[32] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, Scaling vision transformers, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2022), pp. 12104–12113.

[33] J. Lever, M. Krzywinski, and N. Altman, Points of significance: model selection and overfitting, Nat. Methods **13**, 703 (2016).

[34] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, Understanding deep learning (still) requires rethinking generalization, Commun. ACM **64**, 107 (2021).

[35] S. Geman, E. Bienenstock, and R. Doursat, Neural networks and the bias/variance dilemma, Neural Comput. **4**, 1 (1992).

[36] M. Belkin, D. Hsu, S. Ma, and S. Mandal, Reconciling mod-

ern machine-learning practice and the classical bias-variance trade-off, Proc. Natl. Acad. Sci. USA **116**, 15849 (2019).

[37] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, Deep double descent: Where bigger models and more data hurt, arXiv:1912.02292.

[38] M. Belkin, D. Hsu, and J. Xu, Two models of double descent for weak features, SIAM J. Math. Data Sci. **2**, 1167 (2020).

[39] G. De Palma, B. T. Kiani, and S. Lloyd, Random deep neural networks are biased towards simple functions, in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2019), Vol. 32.

[40] G. Valle-Perez, C. Q. Camargo, and A. A. Louis, Deep learning generalizes because the parameter-function map is biased towards simple functions, in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2019), Vol. 32.

[41] O. Cohen, O. Malka, and Z. Ringel, Learning curves for over-parametrized deep neural networks: A field theory perspective, Phys. Rev. Res. **3**, 023034 (2021).

[42] T. A. Brody, J. Flores, J. B. French, P. A. Mello, A. Pandey, and S. S. M. Wong, Random-matrix physics: Spectrum and strength fluctuations, Rev. Mod. Phys. **53**, 385 (1981).

[43] T. Guhr, A. Müller-Groeling, and H. A. Weidenmüller, Random-matrix theories in quantum physics: Common concepts, Phys. Rep. **299**, 189 (1998).

[44] M. L. Mehta, *Random Matrices*, 3rd ed., Pure and Applied Mathematics Vol. 142 (Elsevier/Academic Press, Amsterdam and San Diego, CA, 2004).

[45] T. Papenbrock and H. A. Weidenmüller, Colloquium: Random matrices and chaos in nuclear spectra, Rev. Mod. Phys. **79**, 997 (2007).

[46] H. A. Weidenmüller and G. E. Mitchell, Random matrices and chaos in nuclear physics: Nuclear structure, Rev. Mod. Phys. **81**, 539 (2009).

[47] E. P. Wigner, On the statistical distribution of the widths and spacings of nuclear resonance levels, in *Mathematical Proceedings of the Cambridge Philosophical Society* (Cambridge University Press, Cambridge, UK, 1951), Vol. 47, pp. 790–798.

[48] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. Nunes Amaral, and H. E. Stanley, Universal and Nonuniversal Properties of Cross Correlations in Financial Time Series, Phys. Rev. Lett. **83**, 1471 (1999).

[49] L. Laloux, P. Cizeau, J.-P. Bouchaud, and M. Potters, Noise Dressing of Financial Correlation Matrices, Phys. Rev. Lett. **83**, 1467 (1999).

[50] L. Laloux, P. Cizeau, M. Potters, and J.-P. Bouchaud, Random matrix theory and financial correlations, Int. J. Theor. Appl. Finance **03**, 391 (2000).

[51] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. Nunes Amaral, T. Guhr, and H. E. Stanley, Random matrix approach to cross correlations in financial data, Phys. Rev. E **65**, 066126 (2002).

[52] B. Rosenow, V. Plerou, P. Gopikrishnan, and H. E. Stanley, Portfolio optimization and the random magnet problem, Europhys. Lett. **59**, 500 (2002).

[53] R. Schäfer, N. F. Nilsson, and T. Guhr, Power mapping with dynamical adjustment for improved portfolio optimization, Quant. Financ. **10**, 107 (2010).

[54] J. Bun, J.-P. Bouchaud, and M. Potters, Cleaning large correlation matrices: Tools from random matrix theory, Phys. Rep. **666**, 1 (2017).

[55] F. Luo, Y. Yang, J. Zhong, H. Gao, L. Khan, D. K. Thompson, and J. Zhou, Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory, BMC Bioinf. **8**, 299 (2007).

[56] Y. Deng, Y.-H. Jiang, Y. Yang, Z. He, F. Luo, and J. Zhou, Molecular ecological network analyses, BMC Bioinf. **13**, 113 (2012).

[57] C. Louart, Z. Liao, and R. Couillet, A random matrix approach to neural networks, Ann. Appl. Probab. **28**, 1190 (2018).

[58] J. Pennington and P. Worah, Nonlinear random matrix theory for deep learning, J. Stat. Mech. (2019) 124005.

[59] A. K. Lampinen and S. Ganguli, An analytic theory of generalization dynamics and transfer learning in deep linear networks, arXiv:1809.10374.

[60] I. Seroussi and Z. Ringel, Separation of scales and a thermodynamic description of feature learning in some cnns, arXiv:2112.15383.

[61] J. Pennington, S. Schoenholz, and S. Ganguli, The emergence of spectral universality in deep networks, in *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, edited by A. Storkey and F. Perez-Cruz, Proceedings of Machine Learning Research Vol. 84 (PMLR, 2018), pp. 1924–1932.

[62] N. P. Baskerville, D. Granziol, and J. P. Keating, Applicability of random matrix theory in deep learning, arXiv:2102.06740.

[63] D. Granziol, Beyond random matrix theory for deep networks, arXiv:2006.07721.

[64] C. H. Martin and M. W. Mahoney, Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning, J. Mach. Learn. Res. **22**, 1 (2021).

[65] P. Cizeau and J. P. Bouchaud, Theory of Lévy matrices, Phys. Rev. E **50**, 1810 (1994).

[66] E. Tarquini, G. Biroli, and M. Tarzia, Level Statistics and Localization Transitions of Lévy Matrices, Phys. Rev. Lett. **116**, 010601 (2016).

[67] C. H. Martin, T. S. Peng, and M. W. Mahoney, Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data, Nat. Commun. **12**, 4122 (2021).

[68] A. Jacot, F. Gabriel, and C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks, arXiv:1806.07572.

[69] L. Chizat, E. Oyallon, and F. Bach, On lazy training in differentiable programming, Adv. Neural Info. Process. Syst. **32** (2019).

[70] G. Yang and E. J. Hu, Feature learning in infinite-width neural networks, arXiv:2011.14522.

[71] S. Fort, G. K. Dziugaite, M. Paul, S. Kharaghani, D. M. Roy, and S. Ganguli, Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel, Adv. Neural Info. Process. Syst. **33**, 5850 (2020).

[72] M. Thamm, M. Staats, and B. Rosenow, All code, scripts and data used in this work are included in a Zenodo archive: https://zenodo.org/record/7075360, Zenodo 10.5281/zenodo.7075360 (2022).

[73] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics* (2010), p. 249.

[74] V. A. Marčenko and L. A. Pastur, Distribution of eigenvalues for some sets of random matrices, Math. USSR-Sbornik **1**, 457 (1967).

[75] A. M. Sengupta and P. P. Mitra, Distributions of singular values for some random matrices, Phys. Rev. E **60**, 3389 (1999).

[76] L. Denby and C. Mallows, *Computing sciences and statistics: Proceedings of the 23rd symposium on the interface*, edited by E. M. Keramidas (Interface Foundation, Fairfax Station, VA, 1991), p. 54.

[77] B. M. Hill, A simple general approach to inference about the tail of a distribution, Ann. Stat. **3** 1163 (1975).

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM **60**, 84 (2017).

[79] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2009), pp. 248–255.

[80] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai *et al.*, Pytorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Red Hook, NY, 2019), Vol. 32, pp. 8024–8035.

[81] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.

[82] M. Brack, J. Damgaard, A. S. Jensen, H. C. Pauli, V. M. Strutinsky, and C. Y. Wong, Funny hills: The shell-correction approach to nuclear shell effects and its applications to the fission process, Rev. Mod. Phys. **44**, 320 (1972).

[83] H. Bruus and J.-C. A. d'Auriac, The spectrum of the two-dimensional Hubbard model at low filling, Europhys. Lett. **35**, 321 (1996).

[84] B. Efron, Bootstrap methods: Another look at the jackknife, Ann. Stat. **7**, 1 (1979).

[85] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*, Monographs on Statistics and Applied Probability Vol. 57 (Chapman & Hall, Boca Raton, FL, 1998).

[86] B. Efron, Second thoughts on the bootstrap, Stat. Sci. **18**, 135 (2003).

[87] S. S. Du, X. Zhai, B. Poczos, and A. Singh, Gradient descent provably optimizes overparameterized neural networks, arXiv:1810.02054.

[88] C. Li and C. Shi, Constrained optimization based low-rank approximation of deep neural networks, in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 732–747.

[89] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, Gradient descent finds global minima of deep neural networks, in *Proceedings of the International Conference on Machine Learning* (PMLR, 2019), pp. 1675–1685.

[90] Z. Allen-Zhu, Y. Li, and Z. Song, A convergence theory for deep learning via overparameterization, in *Proceedings of the International Conference on Machine Learning* (PMLR, 2019), pp. 242–252.

[91] D. Zou, Y. Cao, D. Zhou, and Q. Gu, Gradient descent optimizes overparameterized deep relu networks, Mach. Learn. **109**, 467 (2020).

[92] R. Novak, L. Xiao, J. Hron, J. Lee, A. A. Alemi, J. Sohl-Dickstein, and S. S. Schoenholz, Neural tangents: Fast and easy infinite neural networks in python, in *Proceedings of the International Conference on Learning Representations* (2020).

[93] T. Guhr and A. Müller-Groeling, Spectral correlations in the crossover between GUE and Poisson regularity: On the identification of scales, J. Math. Phys. **38**, 1870 (1997).

[94] A. Clauset, C. R. Shalizi, and M. E. Newman, Power-law distributions in empirical data, SIAM Rev. **51**, 661 (2009).

[95] V. Akgiray and G. G. Booth, The siable-law model of stock returns, J. Bus. Econ. Stat. **6**, 51 (1988).

[96] B. Cheng and S. T. Rachev, Multivariate stable futures prices, Math. Financ. **5**, 133 (1995).

[97] P. Gopikrishnan, V. Plerou, L. A. Nunes Amaral, M. Meyer, and H. E. Stanley, Scaling of the distribution of fluctuations of financial market indices, Phys. Rev. E **60**, 5305 (1999).

[98] T. Lux, The stable paretian hypothesis and the frequency of large returns: An examination of major german stocks, Appl. Financ. Econ. **6**, 463 (1996).

[99] C. Quintos, Z. Fan, and P. C. Phillips, Structural change tests in tail behaviour and the asian crisis, Rev. Econ. Stud. **68**, 633 (2001).

[100] N. H. Chan, S.-J. Deng, L. Peng, and Z. Xia, Interval estimation of value-at-risk based on garch models with heavy-tailed innovations, J. Econ. **137**, 556 (2007).

[101] J. B. Hill, On tail index estimation for dependent, heterogeneous data, Econ. Theory **26**, 1398 (2010).

[102] Y. Yoshida and T. Miyato, Spectral norm regularization for improving the generalizability of deep learning, arXiv:1705.10941.