A. The PL–Norm Relation: Deriving a Special Case of Eqn. (8)

Here, we derive Eqn. (8) in the special case of very small PL exponent, as $\mu \to 0$, for an $N \times M$ random matrix $\mathbf{W}$, with $M = N, Q = 1$, and with elements drawn from Eqn. (3).[10] We seek a relation good in the region $\mu \in [0, 2]$, and we will extend the $\mu \sim 0$ results to this full region. That is, we establish this as an asymptotic relation for the VHT Universality class for very small exponents.

To start, recall that

$$\|\mathbf{W}\|_F^2 = \text{Trace}[\mathbf{W}^T\mathbf{W}] = N \text{ Trace}[\mathbf{X}].$$

Since, $\mu \gtrsim 0$, the eigenvalue spectrum is dominated by a single large eigenvalue, it follows that

$$\|\mathbf{W}\|_F^2 \approx N\lambda^{max},$$

where λ^{max} is the largest eigenvalue of the matrix $\mathbf{X}$ (with the $1/N$ normalization). Taking the log of both sides of this expression and expanding leads to

$$\log \|\mathbf{W}\|_F^2 \approx \log (N\lambda^{max}) = \log N + \log \lambda^{max}.$$

Rearranging, we get that

$$\frac{\log \|\mathbf{W}\|_F^2}{\log \lambda^{max}} \approx \frac{\log N}{\log \lambda^{max}} + 1.$$

Thus, for a parameter α satisfying Eqn. (8), we have that

$$\alpha \approx \frac{\log N}{\log \lambda^{max}} + 1.$$

Recall that the relation between α and μ for the VHT Universality class is given in Eqn. (4a) as $\alpha = \frac{1}{2}\mu + 1$. Thus, to establish our result, we need to show that

$$\frac{\log N}{\log \lambda^{max}} \approx \frac{1}{2}\mu.$$

To do this, we use the relation of Eqn. (5) for the tail statistic, i.e., that $\lambda^{max} \approx N^{4/\mu - 1}$. Taking the log of both sides gives

$$\log \lambda^{max} \approx \log N^{4/\mu - 1} = (4/\mu - 1) \log N,$$

from which it follows that

$$\frac{\log N}{\log \lambda^{max}} \approx \frac{\log N}{(4/\mu - 1) \log N} = \frac{1}{4/\mu - 1}.$$

Finally, we can form the Taylor Series for $\frac{1}{4/\mu - 1}$ around, e.g., $\mu = 1.15 \approx 1$, which gives

$$\frac{1}{4/\mu - 1}\bigg|_{\mu=1.15} \approx \frac{1}{2}\mu - \frac{1}{6} + \cdots \approx \frac{1}{2}\mu.$$

This relation is depicted in Figure 5. This establishes the approximate—and rather surprising—linear relation we want for $\mu \in [0, 2]$ for the VHT Universality class of HT-RMT.

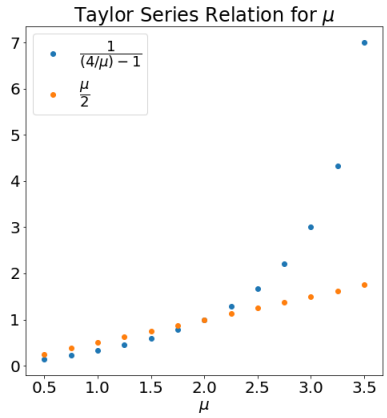


Figure 5. Taylor series expansion for $\frac{1}{4/\mu - 1}$ at $\mu = 1.15 \approx 1$.

B. The PL–Norm Relation: Finite-Size Effects.

Here, we consider finite-size effects in Eqn. (8), both within and across HT Universality classes, i.e., for both VHT and MHT matrices. See Figure 6, which displays $\frac{\log \|\mathbf{W}\|_F^2}{\log \lambda^{max}}$ as a function of the fitted PL exponent α, with varying sizes N (with aspect ratio $Q = 1$). Recall that $\alpha \approx \frac{1}{2}\mu + 1$ for VHT random matrices (Eqn. 4a), while $\alpha = a\mu + b$ for MHT random matrices (Eqn. 4b)), where a, b strongly depend on N and M. Thus, $\mu \in (0, 2)$ for VHT matrices corresponds to $\alpha \in (1, 2)$, while $\alpha \approx (2, 5)$ for MHT matrices.

The numerical results in Figure 6 show that as α increases when $\alpha < 2$, there exists a near-linear relation; and when $\alpha > 2$, for N, M large, the relation saturates, becoming constant, while for smaller N, M, there exists a near-linear relation, but with strong finite-size effects. These numerical results demonstrate that $\log \|\mathbf{W}\|_F^2 \approx \alpha \log \lambda^{max}$ works very well for VHT random matrices, for $\alpha < 2$, and that it works moderately well for MHT matrices and even some WHT matrices. In particular, for MHT matrices, in the finite-size regime, when $N, M \sim \mathcal{O}(100 - 1000)$, which is typical for modern DNNs, the PL-Norm relations holds, *on average*, quite well. This is precisely what we want in

[10] We derive Eqn. (8) at what is sometimes pejoratively known as "at a physics level of rigor." That is fine, as our justification ultimately lies in our empirical results. Recall our goal: to derive a very simple expression relating fitted PL exponents and Frobenius norms that is usable by practical engineers working with state-of-the-art models, i.e., not simply small toy models. There is very little "rigorous" work on HT-RMT, less still on understanding finite-sized effects of HT Universality. Hopefully, our results will lead to more work along these lines.

a practical engineering metric that is designed to describe average test accuracy.

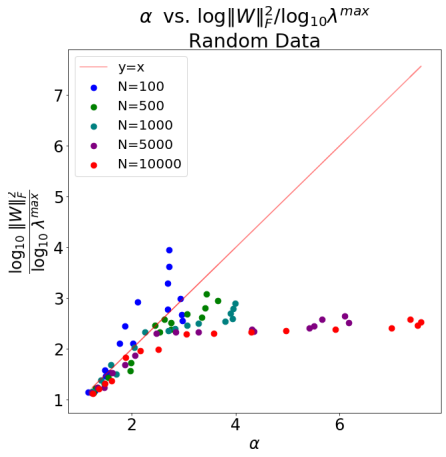


Figure 6. Numerical test of Eqn. (8) for random HT matrices across different HT Universality classes.

C. The PL–Norm Relation: Random Matrices versus Real Data

Here, we show, numerically, that Eqn. (8) holds qualitatively well and more generally than the special case derived previously, including into the MHT Universality class, for both random and real data. To illustrate this, we generate a large number of HT random matrices $\mathbf{W}^{rand}(\mu)$, with varying sizes N (and $Q = 1$), drawn from a Pareto distribution of Eqn. (3), with exponents $\mu \in [0.5, 5]$. We then fit the ESD of each $\mathbf{W}^{rand}(\mu)$ to a PL using the method of Clauset et al. (Clauset et al., 2009; Alstott et al., 2014) to obtain the empirical exponent α. Figure 7(a) shows that there is a near-perfect relation between $\alpha \log \lambda^{max}$ and $\log \|\mathbf{W}\|_F^2$, for this random data. We also performed a similar PL fit for VGG11 weight matrices. (See Section 4 for some details on the VGG11 model.) Figure 7(b) shows the results, demonstrating for the VGG11 data an increasing relation until $\alpha \log \lambda^{max} \approx 2.5$, and a saturation after that point. Figure 7 illustrates (among other things[11]): that multiplying α by $\log \lambda^{max}$ leads to a relation that increases linearly with the (log of the squared) Frobenius norm for HT random matrices; that the two quantities are linearly correlated for real DNN weight matrices; and that both random HT and real, strongly-correlated matrices show similar saturation effects at large PL exponents.

[11]Clearly, there are also differences between the HT random and the real DNN matrices, most notably that $\alpha \log \lambda^{max}$ achieves much larger values for the random matrices. This is discussed in more detail in Appendix D.

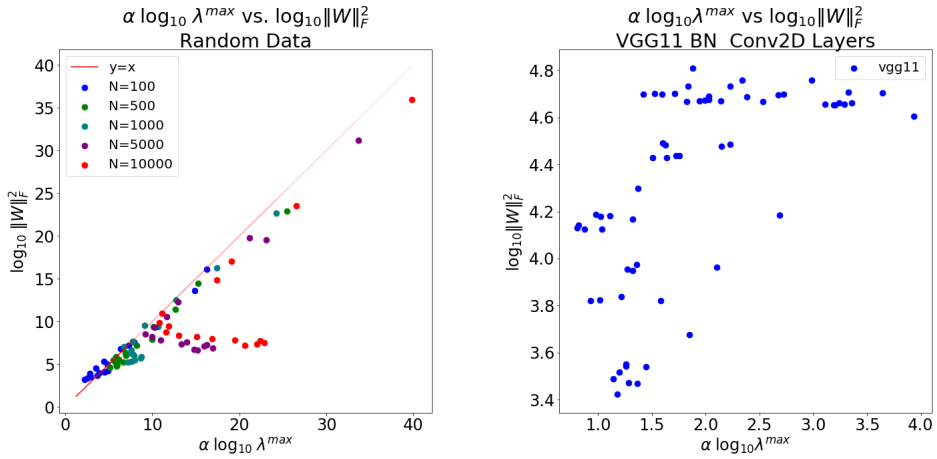
D. Random Pareto versus Non-random DNN Matrices

When we use Universality, as we do in our derivation of the basic PL–Norm Relation, we would like a method that applies both to HT random matrices as well as to non-random, indeed strongly-correlated, pre-trained DNN layer weight matrices that (as evidenced by their ESD properties) are in a HT Universality class. To accomplish this, however, requires some care: while the pre-trained $\mathbf{W}$ matrices do have ESDs that display empirical signatures of HT Universality (Martin & Mahoney, 2018a), they are *not* random Pareto matrices. Many of their properties, including their empirical Frobenius norms, behave very differently than that of a random Pareto matrix. (We saw this in Figure 7, which showed that $\alpha \log \lambda^{max}$ achieves much larger values for HT random matrices than real DNN weight matrices.)

To illustrate this, we generate a large number of HT random matrices $\mathbf{W}^{rand}(\mu)$, with exponents $\mu \in [0.5, 5]$, as described in Section 3. We then fit the ESD of each $\mathbf{W}^{rand}(\mu)$ to a PL using the method of Clauset et al. (Clauset et al., 2009; Alstott et al., 2014) to obtain the empirical PL exponent α. Figure 8(a) displays the relationship between the (log of the squared) Frobenius norm and the μ exponents for these randomly-generated Pareto matrices. (Similar but noisier plots would arise if we plotted this as a function of α, due to imperfections in the PL fit.) We did the same for the weight matrices (extracted from the Conv2D Feature Maps) from the pre-trained VGG11 DNN, again as described in Section 3. Figure 8(b) displays these results, here as a function of α.

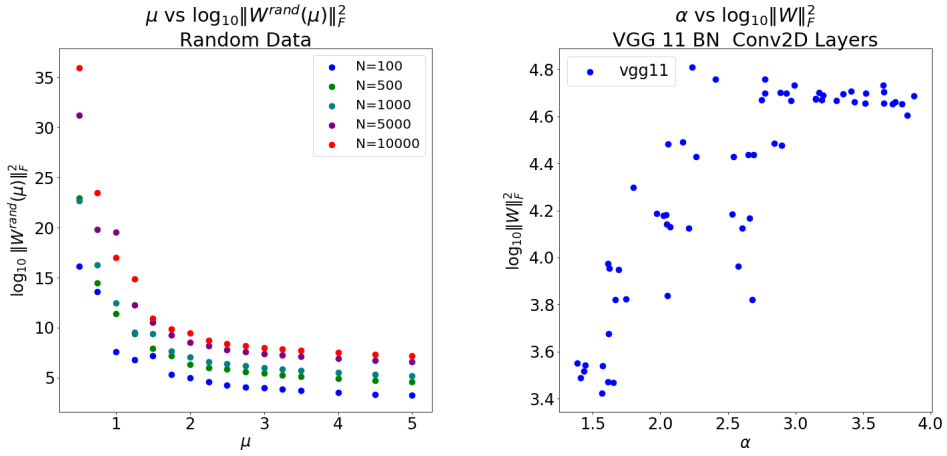
From Figures 8(a) and 8(b), we see that the properties of $\|\mathbf{W}\|_F^2$ differ strikingly for the random Pareto versus real/ron-random DNN weight matrices, and thus care must be taken when applying these Universality principles to strongly correlated systems. For a random Pareto matrix, $\mathbf{W}^{rand}(\mu)$, the Frobenius norm $\|\mathbf{W}^{rand}(\mu)\|_F^2$ *decreases with increasing exponent* (μ); and there is a modest finite-size effect. (In addition, as the tails of the ESD $\rho(\lambda)$ get heavier, the largest eigenvalue λ^{max} of $\mathbf{X}$ scales with the largest element of $\mathbf{W}^{rand}(\mu)$.) For the weight matrices of a pre-trained DNN, however, the Frobenius norm $\|\mathbf{W}\|_F^2$ *increases with increasing exponent* (α), saturating at $\alpha \approx 3$. This happens because, due to the training process, the $\mathbf{W}$ matrices themselves are highly-correlated, and not random matrices with a single large, atypical element.[12] In spite of this, the ESD $\rho(\lambda)$ of these pre-trained correlations matrices $\mathbf{X}$ display Universal HT behavior (Martin & Mahoney, 2018a). In addition, as shown in Figure 7(b), Eqn. (8) is approximately satisfied, in the sense that $\alpha \log \lambda^{max}$ is positively correlated with $\log \|\mathbf{W}\|_F^2$. This is one of the

[12]This is easily seen by simply randomizing the elements of a real DNN weight matrix, and computing the ESD again.



(a) Random Pareto Matrices (b) VGG11 Weight matrices

Figure 7. Relation between $\alpha \log_{10} \lambda^{max}$ and $\log_{10} \|\mathbf{W}\|_F^2$ for random (Pareto) matrices and real (VGG11) DNN weight matrices.



(a) Random Pareto Matrices (b) Pre-trained VGG11 Weight Matrices

Figure 8. Dependence of Frobenius norm on PL exponents for random Pareto versus pre-trained DNN matrices.

remarkable properties of Mechanistic Universality.

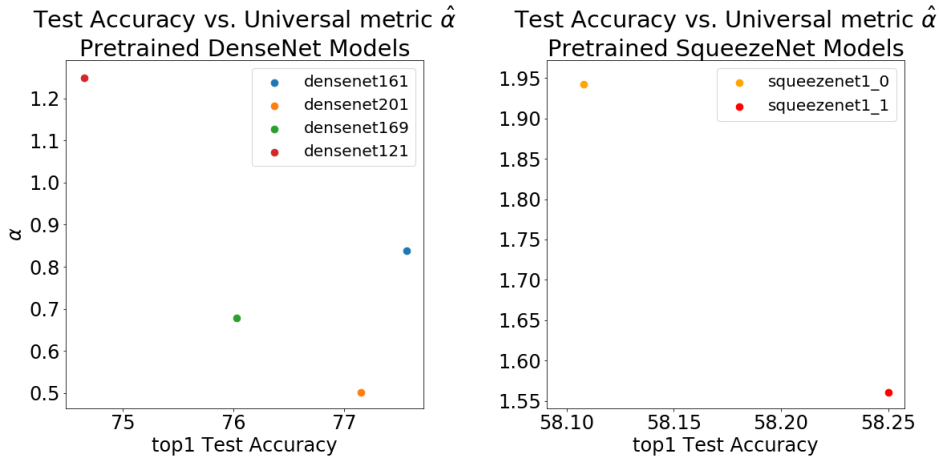
E. Additional Empirical Results

In addition to the VGG and ResNet series of models, we examined a wide range of other DNNs. Here, we summarize some of those results.

More Pre-trained Models. We present results for eleven more series of pre-trained DNN architectures, eight of which show positive results, as with the VGG and ResNet series (in Section 4), and three of which provide counterexample architectures. See Table 3 for a summary of results.

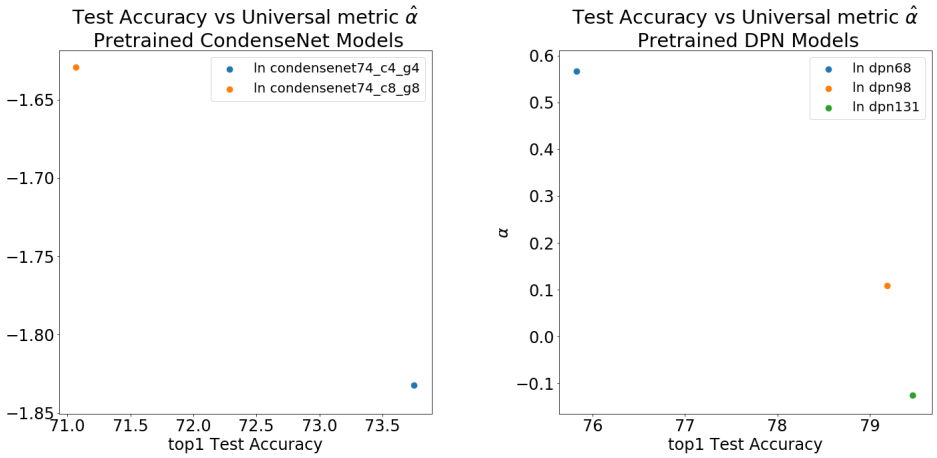
The results that perform as expected are show in Figures 9, 10, 11, and 12. For each set of models, our Universal metric $\hat{\alpha}$ is smaller when, for the most part, the reported (Top 1) test accuracy is larger. This holds approximately true for the three of the four DenseNet models, with densenet169 as an outlier. In fact, this is the only outlier out of 26 DNN models in these 8 architectures. For all of the other pre-trained DNNs, smaller $\hat{\alpha}$ corresponds with smaller test error and larger test accuracy, as predicted by our theory.

Counterexamples. In such a large corpus of DNNs, there are of course exceptions for a predictive theory. See Figure 13 (as well as the corresponding rows of Table 3) for the counterexamples. These are ResNeXt, MeNet, and FDMo-



(a) DenseNet (b) SqueezeNet

Figure 9. Pre-trained Densenet and SqueezeNet PyTorch Models. Top 1 Test Accuracy versus $\hat{\alpha}$.



(a) CondenseNet (b) DPN

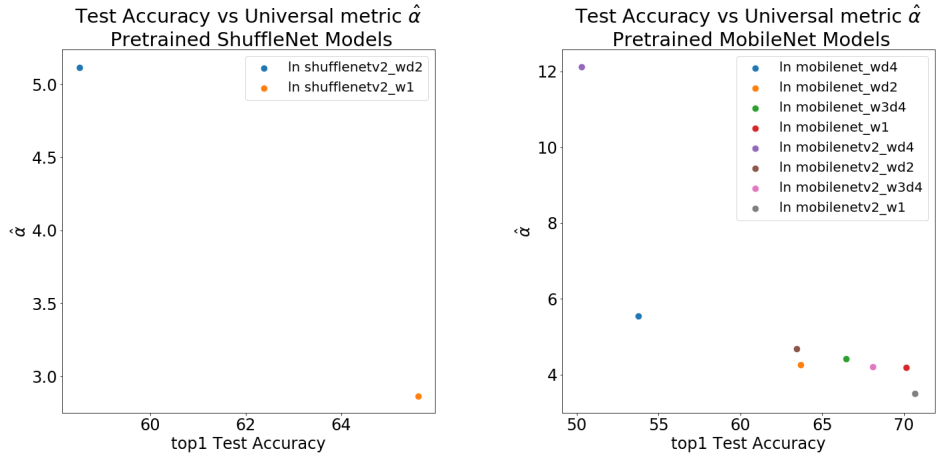
Figure 10. Pre-trained CondenseNet and DPN Models. Top 1 Test Accuracy versus $\hat{\alpha}$.

bileNet. For ResNeXt, there are only two models, and the $\hat{\alpha}$ is larger for the less accurate model. For MeNet, there are seven different models, and there is no discernible pattern in the data. Finally, for FDMobileNet, there are three different pre-trained models, and, again, the $\hat{\alpha}$ is larger for the less accurate models. We have not looked in detail at these results and simply present them for completeness.

F. Additional Discussion

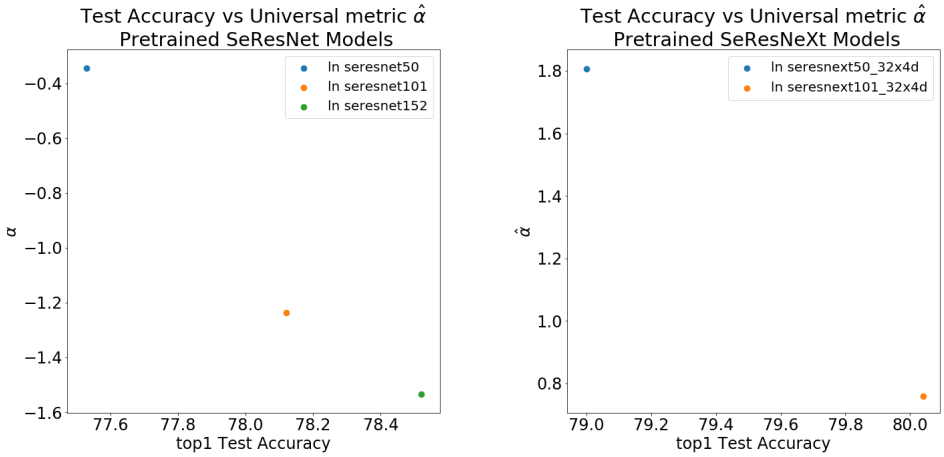
We have presented an *unsupervised* capacity control metric which predicts trends in test accuracies of a trained DNN—without peeking at the test data. This complexity metic, $\hat{\alpha}$ of Eqn. (9), is a weighted average of the PL exponents α for each layer weight matrix, where α is defined in the recent HT-SR Theory (Martin & Mahoney, 2018a), and where the weights are the largest eigenvalue λ^{max} of the correlation matrix $\mathbf{X}$. We examine several commonly-available, pre-trained, production-quality DNNs by plotting $\hat{\alpha}$ versus the reported test accuracies. This covers classes of DNN architectures including the VGG models, ResNet, DenseNet, etc. In nearly every class, and except for a few counterexamples, smaller $\hat{\alpha}$ corresponds to better average test accuracies, thereby providing a strong predictor of model quality. We also show that this new complexity metric $\hat{\alpha}$ is approximately the average log of the squared Frobenius norm of the layer weight matrices, $\langle \log \|\mathbf{W}\|_F^2 \rangle$, when accounting for finite-size effects:

$$\alpha \log \lambda^{max} \approx \log \|\mathbf{W}\|_F^2.$$



(a) ShuffleNet (b) MobileNet

Figure 11. Pre-trained ShuffleNet and MobileNet Models. Top 1 Test Accuracy versus $\hat{\alpha}$.



(a) SeResNet (b) SeResNeXt

Figure 12. Pre-trained SeResNet and SeResNeXt Models. Top 1 Test Accuracy versus $\hat{\alpha}$.

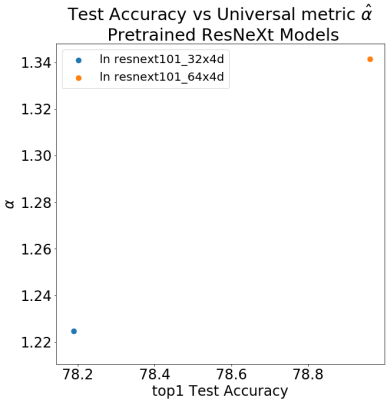
This provides an interesting connection between the Statistical Physics approach to learning (from Martin and Mahoney (Martin & Mahoney, 2017; 2018a), that we extend here) and methods such as that of Liao et al. (Liao et al., 2018), who use norm-based capacity control metrics to bound worst-case generalization error.

It is worth emphasizing that we are taking a very non-standard approach (at least for the DNN and ML communities) to address our main question. We did not train/retrain lots and lots of (typically rather small) models, analyzing training/test curves, trying to glean from them bits of insight that might then extrapolate to more realistic models. Instead, we took advantage of the fact that there already exist many (typically rather large) publicly-available pre-trained models, and we analyzed the properties of these models. That is, we viewed these publicly-available pre-trained models as artifacts of the world that achieve state-of-the-art performance in computer vision, NLP, and related applications; and we attempted to understand why. To do so, we analyzed the empirical (spectral) properties of these models; and we then extracted data-dependent metrics to predict their generalization performance on production-quality models. Given well-known challenges associated with training, we suggest that this methodology be applied more generally.

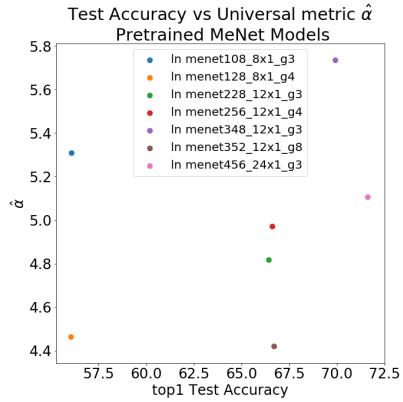
Finally, one interesting aspect of our approach is that we can apply these complexity metrics *across related DNN*

Architecture	Model	Top 1	$\hat{\alpha}$
Working Examples			
DenseNet	densenet121	74.43	1.25
	densenet161	77.14	0.84
	densenet169	75.60	0.68
	densenet201	76.90	0.50
SqueezeNet	squeezenet_v1_0	58.69	2.55
	squeezenet_v1_1	58.18	1.56
CondenseNet	condensenet74_c4_g4	73.75	-1.83
	condensenet74_c8_g8	71.07	-1.63
DPN	dpn68	75.83	0.57
	dpn98	79.19	0.11
	dpn131	79.46	-0.13
ShuffleNet	shufflenetv2_wd2	58.52	5.12
	shufflenetv2_w1	65.61	2.86
MobileNet	mobilenet_wd4	53.74	5.54
	mobilenet_wd2	63.70	4.26
	mobilenet_w3d4	66.46	4.41
	mobilenet_w1	70.14	4.19
	mobilenetv2_wd4	50.28	12.12
	mobilenetv2_wd2	63.46	4.69
	mobilenetv2_w3d4	68.11	4.21
	mobilenetv2_w1	70.69	3.50
SE-ResNet	seresnet50	77.53	-0.35
	seresnet101	78.12	-1.24
	seresnet152	78.52	-1.53
SE-ResNeXt	seresnext50_32x4d	79.00	1.81
	seresnext101_32x4d	80.04	0.76
Counterexamples			
ResNeXt	resnext101_32x4d	78.19	1.22
	resnext101_64x4d	78.96	1.34
MeNet	menet108_8x1_g3	56.08	5.31
	menet128_8x1_g4	56.05	4.46
	menet228_12x1_g3	66.43	4.82
	menet256_12x1_g4	66.59	4.97
	menet348_12x1_g3	69.90	5.74
	menet352_12x1_g8	66.69	4.42
	menet456_24x1_g3	71.60	5.11
FDMobileNet	fdmobilenet_wd4	44.23	6.40
	fdmobilenet_wd2	56.15	7.01
	fdmobilenet_w1	65.30	7.10

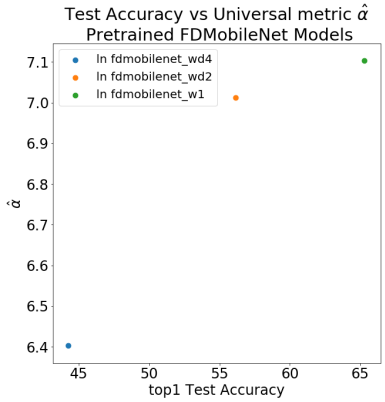
Table 3. Results for more pre-trained DNN models. Models provided in the OSMR Sandbox, implemented in pyTorch. Top 1 refers to the Top 1 Accuracy, which 100.0 minus the Top 1 reported error.



(a) ResNeXt



(b) MeNet



(c) FDMobileNet

Figure 13. Pre-trained ResNeXt, MeNet, and FDMobileNet Models provide counterexamples to our main trends. Top 1 Test Accuracy versus $\hat{\alpha}$.

architectures. This is in contrast to the standard practice in ML. The equivalent notion would be to compare margins across SVMs, applied to the same data, but with different kernels. One loose interpretation is that a set of related of DNN models (i.e., VGG11, VGG13, etc.) is analogous to a single, very complicated kernel, and that the hierarchy of architectures is analogous to the hierarchy of hypothesis spaces in more traditional VC theory. Making this idea precise is clearly of interest.

We expect our result will have applications in the fine-tuning of pre-trained DNNs used for transfer learning, as in NLP and related applications. Moreover, because we do not need to peek at the test data, our approach may prevent information from leaking from the test set into the model, thereby helping to prevent overtraining and making fined-

tuned DNNs more robust. Finally, our work also leads to a much harder theoretical question: is it possible to characterize properties of realistic DNNs to determine whether a DNN is overtrained—without peeking at the test data?