# Appendix: On the Overlooked Structure of Stochastic Gradients

Zeke Xie[1], Qian-Yuan Tang[2], Mingming Sun[1] and Ping Li[1]

[1]Baidu Research
[2]Department of Physics, Hong Kong Baptist University
Correspondence: *xiezeke@baidu.com,tangqy@hkbu.edu.hk*

## A Experimental Settings

**Computational environment.** The experiments are conducted on a computing cluster with NVIDIA® V100 GPUs and Intel® Xeon® CPUs. We can produce/reproduce the main experiments using PaddlePaddle (Ma et al., 2019) and PyTorch (Paszke et al., 2019).

**Model:** LeNet (LeCun et al., 1998), Fully Connected Networks (FCN), and ResNet18 (He et al., 2016). **Dataset**: MNIST (LeCun, 1998), CIFAR-10/100 (Krizhevsky and Hinton, 2009), and Avila (De Stefano et al., 2018). Avila is a non-image dataset.

### A.1 Gradient History Matrices

In this paper, we compute the Gradient History Matrices and the covariance for multiple models on multiple datasets. Then, we use the elements in Gradient History Matrices and the eigenvalues of the covariance/second-moment to evaluate the goodness of fitting Gaussian distributions or power-law distributions via $\chi^2$ tests and KS tests.

The Gradient History Matrix is an $n \times T$ matrix. For the experiment of LeNet and FCN, we compute the gradients for $T = 5000$ iterations at a fixed randomly initialized position $\theta^{(0)}$ or a pretrained position $\theta^{\star}$. Due to limit of memory capacity, for the experiment of ResNet18, we compute the gradients for $T = 200$ iterations at $\theta^{(0)}$ or $\theta^{\star}$.

A Gradient History Matrix can be used to compute the covariance or the second moment of stochastic gradients for a neural network. Note that a covariance matrix is an $n \times n$ matrix, which is extremely large for modern neural networks. Thus, we mainly analyze the gradient structures of LeNet and FCN at an affordable computational cost.

### A.2 Models and Datasets

**Models:** LeNet (LeCun et al., 1998), Fully Connected Networks (FCN), and ResNet18 (He et al., 2016). We mainly used two-layer FCN which has 70 neurons for each hidden layer, ReLu activations, and BatchNorm layers, unless we specify otherwise.

**Datasets:** MNIST (LeCun, 1998) and CIFAR-10/100 (Krizhevsky and Hinton, 2009), and non-image Avila (De Stefano et al., 2018).

**Optimizers:** SGD, SGD with Momentum, and Adam (Kingma and Ba, 2015).

### A.3 Image classification on MNIST

We perform the common per-pixel zero-mean unit-variance normalization as data preprocessing for MNIST.

**Pretraining Hyperparameter Settings:** We train neural networks for 50 epochs on MNIST for obtaining pretrained models. For the learning rate schedule, the learning rate is divided by 10 at the epoch of $40\%$ and $80\%$. We use $\eta = 0.1$ for SGD/Momentum and $\eta = 0.001$ for Adam. The batch size is set to 128. The strength of weight decay defaults to $\lambda = 0.0005$ for pretrained models. We set the momentum hyperparameter $\beta_1 = 0.9$ for SGD Momentum. As for other optimizer hyperparameters, we apply the default settings directly.

**Hyperparameter Settings for $G$:** We use $\eta = 0.1$ for SGD/Momentum and $\eta = 0.001$ for Adam. The batch size is set to 1 and no weight decay is used, unless we specify them otherwise.

### A.4 Image classification on CIFAR-10 and CIFAR-100

**Data Preprocessing For CIFAR-10 and CIFAR-100:** We perform the common per-pixel zero-mean unit-variance normalization, horizontal random flip, and $32 \times 32$ random crops after padding with 4 pixels on each side.

**Pretraining Hyperparameter Settings:** In the experiments on CIFAR-10 and CIFAR-100: $\eta = 1$ for Vanilla SGD; $\eta = 0.1$ for SGD (with Momentum); $\eta = 0.001$ for Adam. For the learning rate schedule, the learning rate is divided by 10 at the epoch of $\{80, 160\}$ for CIFAR-10 and $\{100, 150\}$ for CIFAR-100, respectively. The batch size is set to 128 for both CIFAR-10 and CIFAR-100. The batch size is set to 128 for MNIST, unless we specify it otherwise. The strength of weight decay defaults to $\lambda = 0.0005$ as the baseline for all optimizers unless we specify it otherwise. We set the momentum hyperparameter $\beta_1 = 0.9$ for SGD and adaptive gradient methods which involve in Momentum. As for other optimizer hyperparameters, we apply the default settings directly.

**Hyperparameter Settings for $G$:** We use $\eta = 1$ for SGD, $\eta = 0.1$ for SGD with Momentum, and $\eta = 0.001$ for Adam. The batch size is set to 1 and no weight decay is used, unless we specify them otherwise.

### A.5 Learning with noisy labels

We trained LeNet via SGD (with Momentum) on corrupted MNIST with various (asymmetric) label noise. We followed the setting of Han et al. (2018) for generating noisy labels for MNIST. The symmetric label noise is generated by flipping every label to other labels with uniform flip rates $\{40\%, 80\%\}$. In this paper, we used symmetric label noise.

For obtaining datasets with random labels which have little knowledge behind the pairs of instances and labels, we also randomly shuffle the labels of MNIST to produce Random MNIST.

## B Goodness-of-Fit Tests

### B.1 Kolmogorov-Smirnov Test

In this section, we introduce how to conduct the Kolmogorov-Smirnov Goodness-of-Fit Test.

We used Maximum Likelihood Estimation (MLE) (Myung, 2003; Clauset et al., 2009) for estimating the parameter $\beta$ of the fitted power-law distributions and the Kolmogorov-Smirnov Test (KS Test) (Massey Jr, 1951; Goldstein et al., 2004) for statistically testing the goodness of fitting power-law distributions. The KS test statistic is the KS distance $d_{\mathrm{ks}}$ between the hypothesized (fitted) distribution and the empirical data, which measures the goodness of fit. It is defined as

$$d_{\mathrm{ks}} = \sup_{\lambda} |F^{\star}(\lambda) - \hat{F}(\lambda)|, \tag{1}$$

where $F^{\star}(\lambda)$ is the hypothesized cumulative distribution function and $\hat{F}(\lambda)$ is the empirical cumulative distribution function based on the sampled data (Goldstein et al., 2004). The estimated power exponent via MLE (Clauset et al., 2009) can be written as

$$\hat{\beta} = 1 + K \left[ \sum_{i=1}^{K} \ln \left( \frac{\lambda_i}{\lambda_{\min}} \right) \right]^{-1}, \tag{2}$$

where $K$ is the number of tested samples and we set $\lambda_{\min} = \lambda_k$. In this paper, we choose the top $K = 1000$ data points for the power-law hypothesis tests, unless we specify it otherwise. We note that the Powerlaw library (Alstott et al., 2014) provides a convenient tool to compute the KS distance, $d_{\mathrm{ks}}$, and estimate the power exponent.

According to the practice of Kolmogorov-Smirnov Test (Massey Jr, 1951), we state ***the null hypothesis*** that the tested spectrum is not power-law. We state the alternative hypothesis, called ***the power-law hypothesis***, that the tested spectrum is power-law. If $d_{\mathrm{ks}}$ is higher than the critical value $d_{\mathrm{c}}$ at the $\alpha = 0.05$ significance level, we would accept the null hypothesis. In contrast, if $d_{\mathrm{ks}}$ is lower than the critical value $d_{\mathrm{c}}$ at the $\alpha = 0.05$ significance level, we would reject the null hypothesis and accept the power-law hypothesis.

For each KS test in this paper, we select top $k = 1000$ data points from dimension-wise gradients and iteration-wise gradients and top $k = 1000$ covariance eigenvalues as the tested sets to measure the goodness of power laws. We choose the largest data points for two reasons. First, focusing on relatively large values is very reasonable and common in various fields' power-law studies (Stringer et al., 2019; Reuveni et al., 2008; Tang and Kaneko, 2020), as real-world distributions typically follow power laws only after/large than some cutoff values (Clauset et al., 2009) for ensuring the convergence of the probability distribution. Second, researchers are usually more interested in significantly large eigenvalues due to the low-rank matrix approximation.

### B.2 $\chi^2$ Test

In this section, we introduce how we conduct $\chi^2$ Test to evaluate the Gaussianity.

We directly used the $\chi^2$ Normal Test implemented by the classical Python-based scientific computing package, Scipy (Virtanen et al., 2020), to evaluate the Gaussianity of empirical data. Note that we need to normalize the empirical data via whitening (zero-mean and unit-variance) before the tests.

The Gaussianity test statistic, $p$-value, is returned by the squared sum of the statistics of Skewness Test and Kurtosis Test (Cain et al., 2017). Skewness is a measure of symmetry. A distribution or dataset is symmetric if it looks the same to the left and right of the center. Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. Empirical data with high kurtosis tend to have heavy tails. Empirical data with low kurtosis tend to have light tails. Thus, $\chi^2$ Test can reflect both Gaussianity and heavy tails. In this paper, we randomly choose $K = 100$ data points for the Gaussian hypothesis tests, unless we specify it otherwise.

We may write the $p$-value return by $\chi^2$ Test as

$$p = z_S^2 + z_K^2, \tag{3}$$

where $z_S$ is the Skewness Test statistic and $z_K$ is the Kurtosis Test statistic. There are a number of ways to compute $z_S$ and $z_K$ in practice. It is convenient to use the default two-sided setting in Virtanen et al. (2020). Please refer to Virtanen et al. (2020) and the source code of $stats.skewtest$ and $stats.kurtosistest$ for the detailed implementation.

For each $\chi^2$ test in this paper, we randomly select $k = 100$ data points from both dimension-wise gradients and iteration-wise gradients as the tested set to measure the Gaussianity. The returned test statistic, $p$-value, is a classical indicator of the relative goodness of Gaussianity for two types of gradients.

## C Statistical Test Results

We present the statistical test results of the eigengaps of the gradient covariances in Table 1 and the visualized results in Figure 1.

Table 1: KS statistics of the covariance eigengaps of LeNet and FCN.

| Dataset | Model | Training | $d_{\mathrm{ks}}$ | $d_{\mathrm{c}}$ | Power-Law | $\hat{s}$ |
|---|---|---|---|---|---|---|
| MNIST | LeNet | Pretrain | 0.0205 | 0.043 | Yes | 5.111 |
| MNIST | LeNet | Random | 0.0221 | 0.043 | Yes | 2.232 |
| MNIST | FCN | Pretrain | 0.0219 | 0.043 | Yes | 1.668 |
| MNIST | FCN | Random | 0.0231 | 0.043 | Yes | 1.668 |

We present the statistical test results of dimension-wise gradients and iteration-wise gradients of LeNet and ResNet18 on various datasets in Tables 2 and 3.

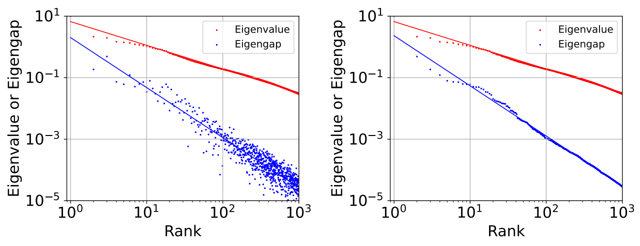

Figure 1: The eigengaps of stochastic gradient covariances are also approximately power-laws. Dataset: MNIST. Model: FCN. Left figure displays the eigengaps by original rank indices sorted by eigenvalues. Right figure displays the eigengaps by rank indices re-sorted by eigengaps.

We conducted the KS Tests for all of our studied covariance spectra. We display the KS test statistics and the estimated power exponents $\hat{s}$ in the tables. For better visualization, we color accepting the power-law hypothesis in blue and color accepting the null hypothesis (and the cause) in red. The KS Test statistics of the covariance spectra are shown in Tables 4, 5, 6, 7, 8, and 9.

Table 2: The KS and $\chi^2$ statistics and the hypothesis acceptance rates of iteration-wise gradients with respect to the batch size. Model: LeNet. Dataset: MNIST

| Type | Training | Setting | $\bar{d}_{\text{ks}}$ | $d_{\text{c}}$ | Power-Law Rate | $\bar{p}$-value | Gaussian Rate |
|---|---|---|---|---|---|---|---|
| Iteration | Random | $B=1$ | 0.428 | 0.0430 | 0.067% | 0.047 | 12.6% |
| Iteration | Random | $B=3$ | 0.385 | 0.0430 | 0.17% | 0.089 | 21.5% |
| Iteration | Random | $B=10$ | 0.267 | 0.0430 | 0.25% | 0.173 | 28.5% |
| Iteration | Random | $B=30$ | 0.249 | 0.0430 | 0.16% | 0.240 | 50.9% |
| Iteration | Random | $B=100$ | 0.191 | 0.0430 | 0.079% | 0.321 | 65.1% |
| Iteration | Random | $B=300$ | 0.119 | 0.0430 | 0.033% | 0.382 | 74.5% |
| Iteration | Random | $B=1000$ | 0.120 | 0.0430 | 0.041% | 0.388 | 75.5% |
| Dimension | Random | $B=1$ | 0.0306 | 0.0430 | 90.6% | $4.51 \times 10^{-5}$ | 0% |
| Dimension | Random | $B=3$ | 0.0358 | 0.0430 | 74.5% | $9.07 \times 10^{-5}$ | 0.02% |
| Dimension | Random | $B=10$ | 0.0392 | 0.0430 | 65.3% | $1.78 \times 10^{-4}$ | 0% |
| Dimension | Random | $B=30$ | 0.0379 | 0.0430 | 68.9% | $2.29 \times 10^{-4}$ | 0.14% |
| Dimension | Random | $B=100$ | 0.0355 | 0.0430 | 76.6% | $4.11 \times 10^{-4}$ | 0.18% |
| Dimension | Random | $B=300$ | 0.0269 | 0.0430 | 97.5% | $1.21 \times 10^{-3}$ | 0.48% |
| Dimension | Random | $B=1000$ | 0.0309 | 0.0430 | 90.6% | $1.43 \times 10^{-4}$ | 0% |

Table 3: The KS and $\chi^2$ statistics and the hypothesis acceptance rates of the gradients over dimensions and iterations, respectively. Model: ResNet18. Batch Size: 100. In the second column "random" means randomly initialized models, while "pretrain" means pretrained models.

| Dataset | Training | SG Type | $\bar{d}_{\text{ks}}$ | $d_{\text{c}}$ | Power-Law Rate | $\bar{p}$-value | Gaussian Rate |
|---|---|---|---|---|---|---|---|
| CIFAR-10 | Random | Dimension | 0.0924 | 0.0962 | 54.3% | $1.73 \times 10^{-2}$ | 6.4% |
| CIFAR-10 | Random | Iteration | 0.141 | 0.0962 | 1.32% | 0.495 | 93.4% |
| CIFAR-10 | Pretrain | Dimension | 0.0717 | 0.0962 | 82.6% | $1.1 \times 10^{-2}$ | 3.2% |
| CIFAR-10 | Pretrain | Iteration | 0.140 | 0.0962 | 1.38% | 0.497 | 93.5% |
| CIFAR-100 | Random | Dimension | 0.0631 | 0.0962 | 92.4% | $8.55 \times 10^{-3}$ | 3% |
| CIFAR-100 | Random | Iteration | 0.141 | 0.0962 | 1.36% | 0.496 | 93.2% |
| CIFAR-100 | Pretrain | Dimension | 0.0637 | 0.0962 | 88.5% | $8.11 \times 10^{-3}$ | 3.4% |
| CIFAR-100 | Pretrain | Iteration | 0.140 | 0.0962 | 1.37% | 0.496 | 93.1% |

Table 4: The KS statistics of the second-moment spectra of dimension-wise gradients for LeNet on MNIST.

| Dataset | Model | Training | Batch | Sample size | Setting | $d_{\mathrm{ks}}$ | $d_{\mathrm{c}}$ | Power-Law | $\hat{s}$ |
|---|---|---|---|---|---|---|---|---|---|
| MNIST | LeNet | Pretrain | 1 | 1000 | - | 0.0206 | 0.0430 | Yes | 1.302 |
| MNIST | LeNet | Pretrain | 10 | 1000 | - | 0.0244 | 0.0430 | Yes | 1.313 |
| MNIST | LeNet | Pretrain | 100 | 1000 | - | 0.0171 | 0.0430 | Yes | 1.390 |
| MNIST | LeNet | Pretrain | 1000 | 1000 | - | 0.0173 | 0.0430 | Yes | 1.314 |
| MNIST | LeNet | Pretrain | 10000 | 1000 | - | 0.0204 | 0.0430 | Yes | 1.290 |
| MNIST | LeNet | Pretrain | 60000 | 1000 | - | 0.106 | 0.0430 | No | 0.206 |
| MNIST | LeNet | Random | 1 | 1000 | - | 0.0220 | 0.0430 | Yes | 1.428 |
| MNIST | LeNet | Random | 10 | 1000 | - | 0.0223 | 0.0430 | Yes | 1.334 |
| MNIST | LeNet | Random | 100 | 1000 | - | 0.0228 | 0.0430 | Yes | 1.313 |
| MNIST | LeNet | Random | 1000 | 1000 | - | 0.0198 | 0.0430 | Yes | 1.423 |
| MNIST | LeNet | Random | 10000 | 1000 | - | 0.0213 | 0.0430 | Yes | 1.284 |
| MNIST | LeNet | Random | 60000 | 1000 | - | 0.203 | 0.0430 | No | 0.271 |

Table 5: The KS statistics of the covariance spectra of dimension-wise gradients for LeNet on MNIST.

| Dataset | Model | Training | Batch | Sample size | Setting | $d_{\mathrm{ks}}$ | $d_{\mathrm{c}}$ | Power-Law | $\hat{s}$ |
|---|---|---|---|---|---|---|---|---|---|
| MNIST | LeNet | Random | 1 | 1000 | - | 0.0226 | 0.0430 | Yes | 1.425 |
| MNIST | LeNet | Random | 10 | 1000 | - | 0.0227 | 0.0430 | Yes | 1.331 |
| MNIST | LeNet | Random | 100 | 1000 | - | 0.0230 | 0.0430 | Yes | 1.311 |
| MNIST | LeNet | Random | 1000 | 1000 | - | 0.0200 | 0.0430 | Yes | 1.423 |
| MNIST | LeNet | Random | 10000 | 1000 | - | 0.0287 | 0.0430 | Yes | 1.320 |
| MNIST | LeNet | Pretrain | 1 | 1000 | - | 0.0206 | 0.0430 | Yes | 1.299 |
| MNIST | LeNet | Pretrain | 10 | 1000 | - | 0.0247 | 0.0430 | Yes | 1.310 |
| MNIST | LeNet | Pretrain | 100 | 1000 | - | 0.0171 | 0.0430 | Yes | 1.386 |
| MNIST | LeNet | Pretrain | 1000 | 1000 | - | 0.0174 | 0.0430 | Yes | 1.312 |
| MNIST | LeNet | Pretrain | 10000 | 1000 | - | 0.0223 | 0.0430 | Yes | 1.331 |
| MNIST | LeNet | Pretrain | 1 | 1000 | Label Noise 40% | 0.0289 | 0.0430 | Yes | 1.453 |
| MNIST | LeNet | Pretrain | 1 | 1000 | Label Noise 80% | 0.0138 | 0.0430 | Yes | 11.442 |
| MNIST | LeNet | Pretrain | 1 | 1000 | Random Label | 0.0129 | 0.0430 | Yes | 1.374 |
| MNIST | LeNet | Pretrain | 1 | 1000 | GradClip=1 | 0.0226 | 0.0430 | Yes | 1.323 |
| MNIST | LeNet | Pretrain | 1 | 1000 | GradClip=0.1 | 0.0261 | 0.0430 | Yes | 1.343 |

## References

Alstott, J., Bullmore, E., and Plenz, D. (2014). powerlaw: a python package for analysis of heavy-tailed distributions. *PloS one*, 9(1):e85777.

Cain, M. K., Zhang, Z., and Yuan, K.-H. (2017). Univariate and multivariate skewness and kurtosis for measuring nonnormality: Prevalence, influence and estimation. *Behavior research methods*, 49(5):1716–1735.

Clauset, A., Shalizi, C. R., and Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review*, 51(4):661–703.

De Stefano, C., Maniaci, M., Fontanella, F., and di Freca, A. S. (2018). Reliable writer identification in medieval manuscripts through page layout features: The “avila” bible case. *Engineering Applications of Artificial Intelligence*, 72:99–110.

Goldstein, M. L., Morris, S. A., and Yen, G. G. (2004). Problems with fitting to the power-law distribution. *The European Physical Journal B-Condensed Matter and Complex Systems*, 41(2):255–258.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537.

Table 6: The KS statistics of the second-moment spectra of dimension-wise gradients for FCN on MNIST.

| Dataset | Model | Training | Batch | Sample size | Setting | $d_{\text{ks}}$ | $d_{\text{c}}$ | Power-Law | $\hat{s}$ |
|---|---|---|---|---|---|---|---|---|---|
| MNIST | 2Layer-FCN | Pretrain | 10 | 1000 | - | 0.0415 | 0.0430 | Yes | 0.866 |
| MNIST | 2Layer-FCN | Random | 10 | 1000 | - | 0.0418 | 0.0430 | Yes | 0.864 |
| MNIST | 2Layer-FCN | Random | 10 | 1000 | Noise | 0.0427 | 0.0430 | Yes | 0.862 |
| MNIST | 2Layer-FCN | Pretrain | 10 | 1000 | Width=70 | 0.0415 | 0.0430 | Yes | 0.866 |
| MNIST | 2Layer-FCN | Pretrain | 10 | 1000 | Width=30 | 0.0425 | 0.0430 | Yes | 0.869 |
| MNIST | 2Layer-FCN | Pretrain | 10 | 1000 | Width=10 | 0.0486 | 0.0430 | No | |
| MNIST | 2Layer-FCN | Random | 10 | 1000 | Width=70 | 0.0418 | 0.0430 | Yes | 0.864 |
| MNIST | 2Layer-FCN | Random | 10 | 1000 | Width=30 | 0.0488 | 0.0430 | No | |
| MNIST | 2Layer-FCN | Random | 10 | 1000 | Width=10 | 0.0491 | 0.0430 | No | |
| MNIST | 1Layer-FCN | Random | 10 | 1000 | - | 0.0384 | 0.0430 | Yes | 1.357 |
| MNIST | 1Layer-FCN | Pretrain | 10 | 1000 | - | 0.0384 | 0.0430 | Yes | 1.355 |

Table 7: The KS statistics of the second-moment spectra of dimension-wise gradients for LNN on MNIST.

| Dataset | Model | Training | Batch | Sample size | Setting | $d_{\text{ks}}$ | $d_{\text{c}}$ | Power-Law | $\hat{s}$ |
|---|---|---|---|---|---|---|---|---|---|
| MNIST | 4Layer-LNN | Pretrain | 10 | 1000 | - | 0.0445 | 0.0430 | No | 1.629 |
| MNIST | 4Layer-LNN | Pretrain | 10 | 1000 | BatchNorm | 0.0268 | 0.0430 | Yes | 0.955 |
| MNIST | 4Layer-LNN | Pretrain | 10 | 1000 | ReLU | 0.0154 | 0.0430 | Yes | 1.074 |

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015*.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images.

LeCun, Y. (1998). The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Ma, Y., Yu, D., Wu, T., and Wang, H. (2019). Paddlepaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Domputing*, 1(1):105–115.

Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

Reuveni, S., Granek, R., and Klafter, J. (2008). Proteins: coexistence of stability and flexibility. *Physical review letters*, 100(20):208101.

Stringer, C., Pachitariu, M., Steinmetz, N., Carandini, M., and Harris, K. D. (2019). High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365.

Tang, Q.-Y. and Kaneko, K. (2020). Long-range correlation in protein dynamics: Confirmation by structural data and normal mode analysis. *PLoS computational biology*, 16(2):e1007670.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

Table 8: The KS statistics of the covariance spectra of dimension-wise gradients for LeNet on CIFAR-10.

| Dataset | Model | Training | Batch | Sample size | Setting | $d_{\text{ks}}$ | $d_{\text{c}}$ | Power-Law | $\hat{s}$ |
|---|---|---|---|---|---|---|---|---|---|
| CIFAR-10 | LeNet | Pretrain | 1 | 1000 | - | 0.0201 | 0.0430 | Yes | 1.257 |
| CIFAR-10 | LeNet | Random | 1 | 1000 | - | 0.0214 | 0.0430 | Yes | 1.300 |
| CIFAR-10 | LeNet | Pretrain | 1 | 1000 | GradClip=0.1 | 0.0244 | 0.0430 | Yes | 1.348 |
| CIFAR-10 | LeNet | Random | 100 | 1000 | SGD | 0.00818 | 0.0430 | Yes | 1.305 |
| CIFAR-10 | LeNet | Random | 100 | 1000 | Weight Decay | 0.0107 | 0.0430 | Yes | 1.300 |
| CIFAR-10 | LeNet | Random | 100 | 1000 | Momentum | 0.00806 | 0.0430 | Yes | 1.262 |
| CIFAR-10 | LeNet | Random | 100 | 1000 | Adam | 0.0634 | 0.0430 | No | |

Table 9: The KS statistics of the covariance spectra of LeNet on CIFAR-100.

| Dataset | Model | Training | Batch | Sample size | Setting | $d_{\text{ks}}$ | $d_{\text{c}}$ | Power-Law | $\hat{s}$ |
|---|---|---|---|---|---|---|---|---|---|
| CIFAR-100 | LeNet | Pretrain | 1 | 1000 | - | 0.0287 | 0.0430 | Yes | 1.276 |
| CIFAR-100 | LeNet | Random | 1 | 1000 | - | 0.0307 | 0.0430 | Yes | 1.229 |
| CIFAR-100 | LeNet | Pretrain | 1 | 1000 | - | 0.0197 | 0.0430 | Yes | 1.076 |